BEA WebLogic Workshop™ Help

Version 8.1 SP2
November 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Table of Contents

XMLBeans API Reference.....1

com.bea.xml Package.....2

com.bea.xml FilterXmlObject Class.....7

com.bea.xml GDate Class.....42

com.bea.xml GDateBuilder Class.....53

com.bea.xml GDateSpecification Interface.....71

com.bea.xml GDuration Class.....80

com.bea.xml GDurationBuilder Class.....87

com.bea.xml GDurationSpecification Interface.....95

com.bea.xml QNameCache Class.....98

com.bea.xml QNameSet Class.....100

com.bea.xml QNameSetBuilder Class.....109

com.bea.xml QNameSetSpecification Interface.....117

com.bea.xml SchemaAttributeGroup Interface.....121

com.bea.xml SchemaAttributeModel Interface.....123

com.bea.xml SchemaComponent Interface.....126

com.bea.xml SchemaField Interface.....130

com.bea.xml SchemaGlobalAttribute Interface.....133

com.bea.xml SchemaGlobalElement Interface.....135

com.bea.xml SchemaIdentityConstraint Interface.....138

com.bea.xml SchemaLocalAttribute Interface.....141

com.bea.xml SchemaLocalElement Interface.....143

com.bea.xml SchemaModelGroup Interface.....145

Table of Contents

com.bea.xml SchemaParticle Interface.....147

com.bea.xml SchemaProperty Interface.....155

com.bea.xml SchemaStringEnumEntry Interface.....166

com.bea.xml SchemaType Interface.....168

com.bea.xml SchemaTypeElementSequencer Interface.....210

com.bea.xml SchemaTypeLoader Interface.....211

com.bea.xml SchemaTypeLoaderException Class.....221

com.bea.xml SchemaTypeSystem Interface.....226

com.bea.xml SimpleValue Interface.....232

com.bea.xml StringEnumAbstractBase Class.....252

com.bea.xml StringEnumAbstractBase.Table Class.....255

com.bea.xml XmlAnySimpleType Interface.....257

com.bea.xml XmlAnySimpleType.Factory Class.....260

com.bea.xml XmlAnyURI Interface.....267

com.bea.xml XmlAnyURI.Factory Class.....269

com.bea.xml XmlBase64Binary Interface.....276

com.bea.xml XmlBase64Binary.Factory Class.....279

com.bea.xml XmlBeans Class.....286

com.bea.xml XmlBoolean Interface.....293

com.bea.xml XmlBoolean.Factory Class.....296

com.bea.xml XmlByte Interface.....303

com.bea.xml XmlByte.Factory Class.....307

com.bea.xml XmlCalendar Class.....314

Table of Contents

com.bea.xml XmlCursor Interface.....320

com.bea.xml XmlCursor.ChangeStamp Interface.....365

com.bea.xml XmlCursor.TokenType Class.....366

com.bea.xml XmlCursor.XmlBookmark Class.....375

com.bea.xml XmlCursor.XmlMark Interface.....378

com.bea.xml XmlDate Interface.....379

com.bea.xml XmlDate.Factory Class.....384

com.bea.xml XmlDateTime Interface.....391

com.bea.xml XmlDateTime.Factory Class.....396

com.bea.xml XmlDecimal Interface.....403

com.bea.xml XmlDecimal.Factory Class.....406

com.bea.xml XmlDocumentProperties Class.....413

com.bea.xml XmlDouble Interface.....421

com.bea.xml XmlDouble.Factory Class.....424

com.bea.xml XmlDuration Interface.....431

com.bea.xml XmlDuration.Factory Class.....434

com.bea.xml XmlENTITIES Interface.....441

com.bea.xml XmlENTITIES.Factory Class.....445

com.bea.xml XmlENTITY Interface.....452

com.bea.xml XmlENTITY.Factory Class.....455

com.bea.xml XmlError Class.....462

com.bea.xml XmlException Class.....473

com.bea.xml XmlFactoryHook.ThreadContext Class.....477

Table of Contents

com.bea.xml XmlFactoryHook Interface.....479

com.bea.xml XmlFloat Interface.....483

com.bea.xml XmlFloat.Factory Class.....486

com.bea.xml XmlGDay Interface.....493

com.bea.xml XmlGDay.Factory Class.....498

com.bea.xml XmlGMonth Interface.....505

com.bea.xml XmlGMonth.Factory Class.....510

com.bea.xml XmlGMonthDay Interface.....517

com.bea.xml XmlGMonthDay.Factory Class.....521

com.bea.xml XmlGYear Interface.....528

com.bea.xml XmlGYear.Factory Class.....533

com.bea.xml XmlGYearMonth Interface.....540

com.bea.xml XmlGYearMonth.Factory Class.....544

com.bea.xml XmlHexBinary Interface.....551

com.bea.xml XmlHexBinary.Factory Class.....554

com.bea.xml XmlID Interface.....561

com.bea.xml XmlID.Factory Class.....564

com.bea.xml XmlIDREF Interface.....571

com.bea.xml XmlIDREF.Factory Class.....574

com.bea.xml XmlIDREFS Interface.....581

com.bea.xml XmlIDREFS.Factory Class.....585

com.bea.xml XmlInt Interface.....592

com.bea.xml XmlInt.Factory Class.....596

Table of Contents

com.bea.xml XmlInteger Interface.....603

com.bea.xml XmlInteger.Factory Class.....607

com.bea.xml XmlLanguage Interface.....614

com.bea.xml XmlLanguage.Factory Class.....617

com.bea.xml XmlLineNumber Class.....624

com.bea.xml XmlLong Interface.....627

com.bea.xml XmlLong.Factory Class.....631

com.bea.xml XmlName Interface.....638

com.bea.xml XmlName.Factory Class.....641

com.bea.xml XmlNCName Interface.....648

com.bea.xml XmlNCName.Factory Class.....651

com.bea.xml XmlNegativeInteger Interface.....658

com.bea.xml XmlNegativeInteger.Factory Class.....661

com.bea.xml XmlNMTOKEN Interface.....668

com.bea.xml XmlNMTOKEN.Factory Class.....671

com.bea.xml XmlNMTOKENS Interface.....678

com.bea.xml XmlNMTOKENS.Factory Class.....682

com.bea.xml XmlNonNegativeInteger Interface.....689

com.bea.xml XmlNonNegativeInteger.Factory Class.....692

com.bea.xml XmlNonPositiveInteger Interface.....699

com.bea.xml XmlNonPositiveInteger.Factory Class.....702

com.bea.xml XmlNormalizedString Interface.....709

com.bea.xml XmlNormalizedString.Factory Class.....711

Table of Contents

com.bea.xml XmlNOTATION Interface.....718

com.bea.xml XmlNOTATION.Factory Class.....720

com.bea.xml XmlObject Interface.....727

com.bea.xml XmlObject.Factory Class.....738

com.bea.xml XmlOptions Class.....748

com.bea.xml XmlPositiveInteger Interface.....765

com.bea.xml XmlPositiveInteger.Factory Class.....768

com.bea.xml XmlQName Interface.....775

com.bea.xml XmlQName.Factory Class.....778

com.bea.xml XmlRuntimeException Class.....785

com.bea.xml XmlSaxHandler Interface.....789

com.bea.xml XmlShort Interface.....791

com.bea.xml XmlShort.Factory Class.....795

com.bea.xml XmlSimpleList Class.....802

com.bea.xml XMLStreamValidationException Class.....810

com.bea.xml XmlString Interface.....812

com.bea.xml XmlString.Factory Class.....814

com.bea.xml XmlTime Interface.....821

com.bea.xml XmlTime.Factory Class.....825

com.bea.xml XmlToken Interface.....832

com.bea.xml XmlToken.Factory Class.....835

com.bea.xml XmlTokenSource Interface.....842

com.bea.xml XmlUnsignedByte Interface.....851

Table of Contents

com.bea.xml XmlUnsignedByte.Factory Class.....855

com.bea.xml XmlUnsignedInt Interface.....862

com.bea.xml XmlUnsignedInt.Factory Class.....866

com.bea.xml XmlUnsignedLong Interface.....873

com.bea.xml XmlUnsignedLong.Factory Class.....876

com.bea.xml XmlUnsignedShort Interface.....883

com.bea.xml XmlUnsignedShort.Factory Class.....887

XMLBeans API Reference

The XMLBeans API provides classes and interfaces for accessing and creating XML, both with and without schema. All of the API is included in one package.

Topics Included in This Section

com.bea.xml Package Summary

Provides a list and descriptions of this package's contents, which includes classes and interfaces designed to provide access to XML.

Related Topics

Getting Started with XMLBeans

com.bea.xml Package

Provides classes for XML Beans schema, infoset, xpath, and xquery handling.

The XML Beans API is designed to work with the XML Beans *schema compiler*. The schema compiler produces XML Bean classes that extend and interact with the basic XML Bean classes and interfaces in this package. It is also possible to use XML Beans without the schema compiler to manipulate and validate XML in a late−bound way (with no types, or with runtime−loaded types only).

This package contains:

1. All the built−in XML Bean classes such as `XmlObject`, `XmlAnySimpleType`, and `XmlInt` (one for each of the 46 built−in XML Schema types).
2. Support for full XML infoset, XPath, and XQuery (via `XmlCursor`, `XmlObject.selectPath(String)` and related methods).
3. Support for computation using XML Schema data types (via datatype classes such as `GDuration`).
4. Support for working with XML Schema type metadata itself (via `SchemaType`, `SchemaTypeSystem`, and related classes).

The place to begin when working with the XML Beans API is the following fundamental classes:

1. `XmlObject` is the base class for all XML Beans.
2. `XmlObject.Factory` is used to load and create XML Bean instances. Each XML Bean class has its own nested factory class.
3. `XmlCursor` is an efficient treewalking API for the full XML infoset.
4. `SchemaType` is the type metadata interface for XML Schema types
5. `XmlBeans` provides a collection of utility functions.

Interface Summary

GDateSpecification	Represents an XML Schema−compatible Gregorian date.
GDurationSpecification	Represents an XML Schema−compatible duration.
QNameSetSpecification	Represents a lattice of finite and infinite sets of QNames.
SchemaAttributeGroup	Represents an attribute group.
SchemaAttributeModel	Represents the attribute structure allowed on a complex type.
SchemaComponent	Represents a global Schema Component.
SchemaField	Represents an element or an attribute declaration.
SchemaGlobalAttribute	Represents a global attribute definition
SchemaGlobalElement	Represents a global element definition.
SchemaIdentityConstraint	Represents an identity constraint definition.
SchemaLocalAttribute	Represents a local or global attribute definition.
SchemaLocalElement	Represents a local or global element definition.
SchemaModelGroup	Represents a model group.
SchemaParticle	Represents a Schema particle definition.
SchemaProperty	Represents a summary of similar SchemaFields in a complex type.
SchemaStringEnumEntry	Describes a code−generated string enumeration

SchemaType	Represents a schema type.
SchemaTypeElementSequencer	This class is used to programatically validate the contents of an XML element.Call to both `SchemaTypeElementSequencer.next(QName)` and `SchemaTypeElementSequencer.peek(QName)` will return true if the element with the provided name is allowed at the current position in the element content, the difference being that `SchemaTypeElementSequencer.next(QName)` will advance the current position, while `SchemaTypeElementSequencer.peek(QName)` won't.
SchemaTypeLoader	Represents a searchable set of XML Schema component definitions.
SchemaTypeSystem	A finite set of XML Schema component definitions.
SimpleValue	All XmlObject implementations can be coerced to SimpleValue.
XmlAnySimpleType	Corresponds to the XML Schema xs:anySimpleType type.
XmlAnyURI	Corresponds to the XML Schema xs:anyURI type.
XmlBase64Binary	Corresponds to the XML Schema xs:base64Binary type.
XmlBoolean	Corresponds to the XML Schema xs:boolean type.
XmlByte	Corresponds to the XML Schema xs:byte type.
XmlCursor	Represents a position between two logical tokens in an XML document.
XmlCursor.ChangeStamp	Represents the state of a dcoument at a particular point in time.
XmlCursor.XmlMark	An abstract `XmlCursor` factory.
XmlDate	Corresponds to the XML Schema xs:date type.
XmlDateTime	Corresponds to the XML Schema xs:dateTime type.
XmlDecimal	Corresponds to the XML Schema xs:decimal type.
XmlDouble	Corresponds to the XML Schema xs:double type.
XmlDuration	Corresponds to the XML Schema xs:duration type.
XmlENTITIES	Corresponds to the XML Schema xs:ENTITIES type, a list type.
XmlENTITY	Corresponds to the XML Schema xs:ENTITY type.
XmlFactoryHook	A hook for the XML Bean Factory mechanism.
XmlFloat	Corresponds to the XML Schema xs:float type.
XmlGDay	Corresponds to the XML Schema xs:gDay type.
XmlGMonth	Corresponds to the XML Schema xs:gMonth type.
XmlGMonthDay	Corresponds to the XML Schema xs:gMonthDay type.
XmlGYear	Corresponds to the XML Schema xs:gYear type.
XmlGYearMonth	Corresponds to the XML Schema xs:gYearMonth type.
XmlHexBinary	Corresponds to the XML Schema xs:hexBinary type.
XmlID	Corresponds to the XML Schema xs:ID type.
XmlIDREF	Corresponds to the XML Schema xs:IDREF type.
XmlIDREFS	Corresponds to the XML Schema xs:IDREFS type, a list type.
XmlInt	Corresponds to the XML Schema xs:int type.
XmlInteger	Corresponds to the XML Schema xs:integer type.
XmlLanguage	Corresponds to the XML Schema xs:language type.
XmlLong	Corresponds to the XML Schema xs:long type.

XmlName	Corresponds to the XML Schema xs:Name type.
XmlNCName	Corresponds to the XML Schema xs:Name type.
XmlNegativeInteger	Corresponds to the XML Schema xs:negativeInteger type.
XmlNMTOKEN	Corresponds to the XML Schema xs:NMTOKEN type.
XmlNMTOKENS	Corresponds to the XML Schema xs:NMTOKENS type, a list type.
XmlNonNegativeInteger	Corresponds to the XML Schema xs:nonNegativeInteger type.
XmlNonPositiveInteger	Corresponds to the XML Schema xs:nonPositiveInteger type.
XmlNormalizedString	Corresponds to the XML Schema xs:normalizedString type.
XmlNOTATION	Corresponds to the XML Schema xs:NOTATION type.
XmlObject	Corresponds to the XML Schema xs:anyType, the base type for all XML Beans.
XmlObjectList	Interface to represent lists of XmlObject instances returned from a query.
XmlPositiveInteger	Corresponds to the XML Schema xs:positiveInteger type.
XmlQName	Corresponds to the XML Schema xs:QName type.
XmlSaxHandler	A holder for a SAX `ContentHandler` and `LexicalHandler` that are capable of loading an `XmlObject` instance.
XmlShort	Corresponds to the XML Schema xs:short type.
XmlString	Corresponds to the XML Schema xs:string type.
XmlTime	Corresponds to the XML Schema xs:time type.
XmlToken	Corresponds to the XML Schema xs:token type.
XmlTokenSource	Represents a holder of XML that can return an `XmlCursor` or copy itself to various media such as `Writer` or `File`.
XmlUnsignedByte	Corresponds to the XML Schema xs:unsignedByte type.
XmlUnsignedInt	Corresponds to the XML Schema xs:unsignedInt type.
XmlUnsignedLong	Corresponds to the XML Schema xs:unsignedLong type.
XmlUnsignedShort	Corresponds to the XML Schema xs:unsignedShort type.

Class Summary

FilterXmlObject	A FilterXmlObject delegates to some other XmlObject, which it can use as its basic source of data, possibly transforming the data along the way or providing additional functionality.
GDate	Represents an XML Schema−compatible Gregorian date.
GDateBuilder	Used to build `GDate`.
GDuration	Represents an XML Schema−compatible duration.
GDurationBuilder	Used to build `GDuration`.
QNameCache	A cache that can be used to pool QName instances.
QNameSet	This interface represents a lattice of finite and infinite sets of QNames.
QNameSetBuilder	Used to build `QNameSet`.
StringEnumAbstractBase	The base class for code−generated string enumeration value classes.
StringEnumAbstractBase.Table	Used to manage singleton instances of enumerations.
XmlAnySimpleType.Factory	A class with methods for creating instances of `XmlAnySimpleType`.
XmlAnyURI.Factory	A class with methods for creating instances of `XmlAnyURI`.

XmlBase64Binary.Factory	A class with methods for creating instances of `XmlBase64Binary`.
XmlBeans	Provides an assortment of utilities for managing XML Bean types, type systems, QNames, paths, and queries.
XmlBoolean.Factory	A class with methods for creating instances of `XmlBoolean`.
XmlByte.Factory	A class with methods for creating instances of `XmlByte`.
XmlCalendar	An XML Schema compatible subclass of `GregorianCalendar`.
XmlCursor.TokenType	An enumeration that identifies the type of an XML token.
XmlCursor.XmlBookmark	Subclasses of XmlBookmark can be used to annotate an XML document.
XmlDate.Factory	A class with methods for creating instances of `XmlDate`.
XmlDateTime.Factory	A class with methods for creating instances of `XmlDateTime`.
XmlDecimal.Factory	A class with methods for creating instances of `XmlDecimal`.
XmlDocumentProperties	This class is used to attach arbitrary information to an XML document.
XmlDouble.Factory	A class with methods for creating instances of `XmlDouble`.
XmlDuration.Factory	A class with methods for creating instances of `XmlDuration`.
XmlENTITIES.Factory	A class with methods for creating instances of `XmlENTITIES`.
XmlENTITY.Factory	A class with methods for creating instances of `XmlENTITY`.
XmlError	Represents a message at a specific XML location.
XmlFactoryHook.ThreadContext	Used to manage the XmlFactoryHook for the current thread.
XmlFloat.Factory	A class with methods for creating instances of `XmlFloat`.
XmlGDay.Factory	A class with methods for creating instances of `XmlGDay`.
XmlGMonth.Factory	A class with methods for creating instances of `XmlGMonth`.
XmlGMonthDay.Factory	A class with methods for creating instances of `XmlGMonthDay`.
XmlGYear.Factory	A class with methods for creating instances of `XmlGYear`.
XmlGYearMonth.Factory	A class with methods for creating instances of `XmlGYearMonth`.
XmlHexBinary.Factory	A class with methods for creating instances of `XmlHexBinary`.
XmlID.Factory	A class with methods for creating instances of `XmlID`.
XmlIDREF.Factory	A class with methods for creating instances of `XmlIDREF`.
XmlIDREFS.Factory	A class with methods for creating instances of `XmlIDREFS`.
XmlInt.Factory	A class with methods for creating instances of `XmlInt`.
XmlInteger.Factory	A class with methods for creating instances of `XmlInteger`.
XmlLanguage.Factory	A class with methods for creating instances of `XmlLanguage`.
XmlLineNumber	A subclass of XmlBookmark that holds line number information.
XmlLong.Factory	A class with methods for creating instances of `XmlLong`.
XmlName.Factory	A class with methods for creating instances of `XmlName`.
XmlNCName.Factory	A class with methods for creating instances of `XmlNCName`.
XmlNegativeInteger.Factory	A class with methods for creating instances of `XmlNegativeInteger`.
XmlNMTOKEN.Factory	A class with methods for creating instances of `XmlNMTOKEN`.
XmlNMTOKENS.Factory	A class with methods for creating instances of `XmlNMTOKENS`.
XmlNonNegativeInteger.Factory	A class with methods for creating instances of `XmlNonNegativeInteger`.
XmlNonPositiveInteger.Factory	A class with methods for creating instances of `XmlNonPositiveInteger`.

XmlNormalizedString.Factory	A class with methods for creating instances of `XmlNormalizedString`.
XmlNOTATION.Factory	A class with methods for creating instances of `XmlNOTATION`.
XmlObject.Factory	Static factory class for creating new instances.
XmlObjectList.Factory	Static factory class for creating new instances of XmlObjectList
XmlOptions	Used to supply options for loading, saving, and compiling, and validating.
XmlPositiveInteger.Factory	A class with methods for creating instances of `XmlPositiveInteger`.
XmlQName.Factory	A class with methods for creating instances of `XmlQName`.
XmlShort.Factory	A class with methods for creating instances of `XmlShort`.
XmlSimpleList	The immutable `List` returned for XML simple list values.
XmlString.Factory	A class with methods for creating instances of `XmlString`.
XmlTime.Factory	A class with methods for creating instances of `XmlTime`.
XmlToken.Factory	A class with methods for creating instances of `XmlToken`.
XmlUnsignedByte.Factory	A class with methods for creating instances of `XmlUnsignedByte`.
XmlUnsignedInt.Factory	A class with methods for creating instances of `XmlUnsignedInt`.
XmlUnsignedLong.Factory	A class with methods for creating instances of `XmlUnsignedLong`.
XmlUnsignedShort.Factory	A class with methods for creating instances of `XmlUnsignedShort`.
XmlUtils	Class providing some Utils for Xml Objects.

Exception Summary

SchemaTypeLoaderException	An exception that is thrown if there is corruption or a version mismatch in a compiled schema type system.
XmlException	A checked exception that can be thrown while processing, parsing, or compiling XML.
XmlRuntimeException	An unchecked XML exception.
XMLStreamValidationException	An exception thrown from a validating XMLInputStream.
XmlUtilsException	Exception for Transformation runtime exceptions.

com.bea.xml

FilterXmlObject Class

public abstract class ***FilterXmlObject***

extends `Object`
implements `SimpleValue`, `XmlObject`

A FilterXmlObject delegates to some other XmlObject, which it can use as its basic source of data, possibly transforming the data along the way or providing additional functionality. The class FilterXmlObject itself simply overrides all methods of XmlObject with versions that pass all requests to the underlying XmlObject. Subclasses of FilterXmlObject may further override some of these methods and may also provide additional methods and fields.

Note: it is important that FilterXmlOBject has no storage (i.e., no non−transient fields), because subclasses may be serializable and adding storage would break the serialization format.

Hierarchy

```
Object
 FilterXmlObject
```

All Implemented Interfaces

`SimpleValue`, `XmlObject`, `XmlTokenSource`

Constructor Summary

FilterXmlObject()

Method Summary

`public BigDecimal` ***`bigDecimalValue()`***
Returns the value as a `BigDecimal`

`public BigInteger` ***`bigIntegerValue()`***
Returns the value as a `BigInteger`

`public boolean` ***`booleanValue()`***
Returns the value as a boolean

`public byte[]`

byteArrayValue()
Returns the value as a byte array

`public byte`
byteValue()
Returns the value as a byte

`public Calendar`
calendarValue()
Returns the value as a `Calendar`

`public XmlObject`
changeType(`SchemaType` newType)
Changes the schema type associated with this data and returns a new XmlObject instance whose schemaType is the new type.

`public int`
compareTo(`Object` obj)
Impelements the Comparable interface by comparing two simple xml values based on their standard XML schema ordering.

`public int`
compareValue(`XmlObject` obj)
This comparison method is similar to compareTo, but rather than throwing a ClassCastException when two values are incomparable, it returns the number 2.

`public XmlObject`
copy()
Returns a deep copy of this XmlObject.

`public Date`
dateValue()
Returns the value as a `Date`

`public XmlDocumentProperties`
documentProperties()
Returns the XmlDocumentProperties object for the document this token source is associated with.

`public double`
doubleValue()
Returns the value as a double

`public StringEnumAbstractBase`
enumValue()
Returns the value as a `StringEnumAbstractBase`

`public XmlObject[]`
execQuery(`String` query)
Executes a query.

`public XmlObject[]`
execQuery(`String` query, `XmlOptions` options)
Executes a query with options.

`public float`
floatValue()
Returns the value as a float

`public GDate`

gDateValue()
Returns the value as a `GDate`

`public GDuration`
gDurationValue()
Returns the value as a `GDuration`

`public BigDecimal`
getBigDecimalValue()
Returns the value as a `BigDecimal`.

`public BigInteger`
getBigIntegerValue()
Returns the value as a `BigInteger`.

`public boolean`
getBooleanValue()
Returns the value as a boolean.

`public byte[]`
getByteArrayValue()
Returns the value as a byte array.

`public byte`
getByteValue()
Returns the value as a byte.

`public Calendar`
getCalendarValue()
Returns the value as a `Calendar`.

`public Date`
getDateValue()
Returns the value as a `Date`.

`public double`
getDoubleValue()
Returns the value as a double.

`public StringEnumAbstractBase`
getEnumValue()
Returns the value as a `StringEnumAbstractBase`.

`public float`
getFloatValue()
Returns the value as a float.

`public GDate`
getGDateValue()
Returns the value as a `GDate`.

`public GDuration`
getGDurationValue()
Returns the value as a `GDuration`.

`public int`
getIntValue()
Returns the value as an int.

`public List`
getListValue()
Returns the value as a `List` of friendly Java objects (String,

Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration).

`public long`
getLongValue()
Returns the value as a long.

`public Object`
getObjectValue()
Returns a union value as a its natural friendly Java object (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration).

`public QName`
getQNameValue()
Returns the value as a `QName`.

`public short`
getShortValue()
Returns the value as a short.

`public String`
getStringValue()
Returns the value as a `String`.

`public SchemaType`
instanceType()
The same as getSchemaType unless this is a union instance or nil value.

`public int`
intValue()
Returns the value as an int

`public boolean`
isImmutable()
True if the value is an immutable value.

`public boolean`
isNil()
True if the value is nil.

`public List`
listValue()
Returns the value as a `List` of friendly Java objects (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration)

`public long`
longValue()
Returns the value as a long

`public Object`
monitor()
Returns the synchronization object for the document.

`public XmlCursor`
newCursor()
Returns a new XML cursor.

`public Node`

newDomNode()
Returns a W3C DOM Node containing the XML represented by this source.

`public Node`
newDomNode(`XmlOptions` options)
Just like newDomNode() but with options.

`public InputStream`
newInputStream()
Returns a new stream containing standard XML text, encoded according to the given encoding.

`public InputStream`
newInputStream(`XmlOptions` options)
Just like newInputStream(String encoding) but with options.

`public Reader`
newReader()
Returns a new character reader containing XML text.

`public Reader`
newReader(`XmlOptions` options)
Just like newReader() but with options.

`public XMLInputStream`
newXMLInputStream()
Returns a new XmlInputStream.

`public XMLInputStream`
newXMLInputStream(`XmlOptions` options)
Just like newXMLInputStream() but with any of a number of options.

`public void`
objectSet(`Object` obj)
Sets the value as an arbitrary `Object`.

`public Object`
objectValue()
Returns a union value as a its natural friendly Java object (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration)

`public QName`
qNameValue()
Returns the value as a `QName`

`public void`
save(`ContentHandler` ch, `LexicalHandler` lh)
Writes the XML represented by this source to the given SAX content and lexical handlers.

`public void`
save(`File` file)
Writes the XML represented by this source to the given File.

`public void`
save(`OutputStream` os)
Writes the XML represented by this source to the given output stream.

`public void` ***save***(`Writer` *w*)
Writes the XML represented by this source to the given writer.

`public void` ***save***(`ContentHandler` *ch*, `LexicalHandler` *lh*, `XmlOptions` *options*)
Writes the XML represented by this source to the given SAX content and lexical handlers.

`public void` ***save***(`File` *file*, `XmlOptions` *options*)
Writes the XML represented by this source to the given File.

`public void` ***save***(`OutputStream` *os*, `XmlOptions` *options*)
Writes the XML represented by this source to the given output stream.

`public void` ***save***(`Writer` *w*, `XmlOptions` *options*)
Writes the XML represented by this source to the given writer.

`public SchemaType` ***schemaType***()
The schema type for this instance.

`public XmlObject[]` ***selectPath***(`String` *path*)
Selects a path.

`public XmlObject[]` ***selectPath***(`String` *path*, `XmlOptions` *options*)
Selects a path, applying options.

`public XmlObject` ***set***(`XmlObject` *srcObj*)
Set the value/type of this XmlObject to be a copy of the source XmlObject.

`public void` ***set***(`String` *obj*)
Sets the value as a `String`

`public void` ***set***(*boolean v*)
Sets the value as a boolean

`public void` ***set***(*byte v*)
Sets the value as a byte.

`public void` ***set***(*short v*)
Sets the value as a short.

`public void` ***set***(*int v*)
Sets the value as an int.

`public void` ***set***(*long v*)

Sets the value as a long.

public void
set(*BigInteger obj*)
Sets the value as a BigInteger.

public void
set(*BigDecimal obj*)
Sets the value as a BigDecimal

public void
set(*float v*)
Sets the value as a float.

public void
set(*double v*)
Sets the value as a double.

public void
set(*byte[] obj*)
Sets the value as a byte array.

public void
set(*StringEnumAbstractBase obj*)
Sets the value as a StringEnumAbstractBase.

public void
set(*Calendar obj*)
Sets the value as a Calendar.

public void
set(*Date obj*)
Sets the value as a Date.

public void
set(*GDateSpecification obj*)
Sets the value as a GDate.

public void
set(*GDurationSpecification obj*)
Sets the value as a GDuration.

public void
set(*QName obj*)
Sets the value as a QName.

public void
set(*List obj*)
Sets the value as a List.

public void
setBigDecimalValue(*BigDecimal obj*)
Sets the value as a BigDecimal.

public void
setBigIntegerValue(*BigInteger obj*)
Sets the value as a BigInteger.

public void
setBooleanValue(*boolean v*)
Sets the value as a boolean.

public void

`setByteArrayValue`(*byte[] obj*)
Sets the value as a byte array.

`public void`
`setByteValue`(*byte v*)
Sets the value as a byte.

`public void`
`setCalendarValue`(`Calendar` *obj*)
Sets the value as a `Calendar`.

`public void`
`setDateValue`(`Date` *obj*)
Sets the value as a `Date`.

`public void`
`setDoubleValue`(*double v*)
Sets the value as a double.

`public void`
`setEnumValue`(`StringEnumAbstractBase` *obj*)
Sets the value as a `StringEnumAbstractBase`.

`public void`
`setFloatValue`(*float v*)
Sets the value as a float.

`public void`
`setGDateValue`(`GDate` *obj*)
Sets the value as a `GDate`.

`public void`
`setGDurationValue`(`GDuration` *obj*)
Sets the value as a `GDuration`.

`public void`
`setIntValue`(*int v*)
Sets the value as an int.

`public void`
`setListValue`(`List` *obj*)
Sets the value as a `List`.

`public void`
`setLongValue`(*long v*)
Sets the value as a long.

`public void`
`setNil`()
Sets the value to nil.

`public void`
`setObjectValue`(`Object` *obj*)
Sets the value as an arbitrary `Object`.

`public void`
`setQNameValue`(`QName` *obj*)
Sets the value as a `QName`.

`public void`
`setShortValue`(*short v*)
Sets the value as a short.

`public void` ***setStringValue***(`String` *obj*)
Sets the value as a `String`.

`public short` ***shortValue***()
Returns the value as a short

`public String` ***stringValue***()
Returns the value as a `String`

`public abstract XmlObject` ***underlyingXmlObject***()
This abstract method is called to obtain the underlying XmlObject.

`public boolean` ***validate***()
Returns true if the contents of this object are valid accoring to schemaType().

`public boolean` ***validate***(`XmlOptions` *options*)
Just like validate(), but with options.

`public boolean` ***valueEquals***(`XmlObject` *obj*)
True if the xml values are equal.

`public int` ***valueHashCode***()

`public List` ***xgetListValue***()
Returns the value as a `List` of XmlAnySimpleType objects.

`public List` ***xlistValue***()
Returns the value as a `List` of XmlAnySimpleType objects

`public String` ***xmlText***()
Returns standard XML text.

`public String` ***xmlText***(`XmlOptions` *options*)
Just like xmlText() but with options.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods from interface `com.bea.xml.SimpleValue`

`bigDecimalValue, bigIntegerValue, booleanValue, byteArrayValue, byteValue, calendarValue, dateValue, doubleValue, enumValue, floatValue, gDateValue, gDurationValue, getBigDecimalValue, getBigIntegerValue, getBooleanValue, getByteArrayValue, getByteValue, getCalendarValue, getDateValue, getDoubleValue, getEnumValue, getFloatValue,`

getGDateValue, getGDurationValue, getIntValue, getListValue, getLongValue, getObjectValue, getQNameValue, getShortValue, getStringValue, instanceType, intValue, listValue, longValue, objectSet, objectValue, qNameValue, set, set, set, set, set, set, set, set, set, set, set, set, set, set, set, set, setBigDecimalValue, setBigIntegerValue, setBooleanValue, setByteArrayValue, setByteValue, setCalendarValue, setDateValue, setDoubleValue, setEnumValue, setFloatValue, setGDateValue, setGDurationValue, setIntValue, setListValue, setLongValue, setObjectValue, setQNameValue, setShortValue, setStringValue, shortValue, stringValue, xgetListValue, xlistValue

Methods from interface `com.bea.xml.XmlObject`

changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode

Methods from interface `com.bea.xml.XmlTokenSource`

documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText

Constructor Detail

FilterXmlObject

`public FilterXmlObject()`

Method Detail

bigDecimalValue() Method

`public BigDecimal bigDecimalValue()`

Description copied from `SimpleValue.bigDecimalValue()`
Returns the value as a `BigDecimal`. *

---

bigIntegerValue() Method

`public BigInteger bigIntegerValue()`

Description copied from `SimpleValue.bigIntegerValue()`
Returns the value as a `BigInteger`. *

booleanValue() Method

`public boolean booleanValue()`

Description copied from `SimpleValue.booleanValue()`
Returns the value as a boolean. *

byteArrayValue() Method

`public byte[] byteArrayValue()`

Description copied from `SimpleValue.byteArrayValue()`
Returns the value as a byte array. *

byteValue() Method

`public byte byteValue()`

Description copied from `SimpleValue.byteValue()`
Returns the value as a byte. *

calendarValue() Method

`public Calendar calendarValue()`

Description copied from `SimpleValue.calendarValue()`
Returns the value as a `Calendar`. *

changeType(SchemaType) Method

`public XmlObject changeType(SchemaType` *`newType`*`)`

Description copied from `XmlObject.changeType(SchemaType)`
Changes the schema type associated with this data and returns a new XmlObject instance whose schemaType is the new type.

Returns null if the type change is not allowed. Certain type changes may be prohibited on the interior of an xml tree due to schema type system constraints (that is, due to a parent container within which the newly specified type is not permissible), but there are no constraints at the roottype changes are never prohibited at the root of an xml tree.

If the type change is allowed, then the new XmlObject should be used rather than the old one. The old XmlObject instance and any other XmlObject instances in the subtree are permanently invalidated and should not be used. (They will return InvalidStateException if you try to use them.) If a type change is done on the

interior of an Xml tree, then xsi:type attributes are updated as needed.

---

compareTo(Object) Method

`public int compareTo(Object obj)`

Description copied from `XmlObject.compareTo(Object)`
Impelements the Comparable interface by comparing two simple xml values based on their standard XML schema ordering. Throws a ClassCastException if no standard ordering applies, or if the two values are incomparable within a partial order.

---

compareValue(XmlObject) Method

`public int compareValue(XmlObject obj)`

Description copied from `XmlObject.compareValue(XmlObject)`
This comparison method is similar to compareTo, but rather than throwing a ClassCastException when two values are incomparable, it returns the number 2. The result codes are −1 if this object is less than obj, 1 if this object is greater than obj, zero if the objects are equal, and 2 if the objects are incomparable.

---

copy() Method

`public XmlObject copy()`

Description copied from `XmlObject.copy()`
Returns a deep copy of this XmlObject. The returned object has the same type as the current object, and has all the content of the XML document underneath the current object. Note that any parts of the XML document above or outside this XmlObject are not copied.

---

dateValue() Method

`public Date dateValue()`

Description copied from `SimpleValue.dateValue()`
Returns the value as a `Date`. *

---

documentProperties() Method

`public XmlDocumentProperties documentProperties()`

Description copied from `XmlTokenSource.documentProperties()`
Returns the XmlDocumentProperties object for the document this token source is associated with.

---

doubleValue() Method

```
public double doubleValue()
```

Description copied from `SimpleValue.doubleValue()`
Returns the value as a double. *

---

enumValue() Method

```
public StringEnumAbstractBase enumValue()
```

Description copied from `SimpleValue.enumValue()`
Returns the value as a `StringEnumAbstractBase`. *

---

execQuery(String) Method

```
public XmlObject[] execQuery(String query)
```

Description copied from `XmlObject.execQuery(String)`
Executes a query. Query can be a string or precompiled query String.

An XQuery is very similar to an XPath, except that it also permits construction of new XML. As a result, the XmlObjects that are returned from execQuery are in newly created documents, separate from the XmlObject on which the query is executed.

Syntax and usage is otherwise similar to selectPath.

---

execQuery(String, XmlOptions) Method

```
public XmlObject[] execQuery(String query,
                             XmlOptions options)
```

Description copied from `XmlObject.execQuery(String, XmlOptions)`
Executes a query with options. Use the *options* parameter to specify the following:

To specify this	Use this method
The document type for the root element.	`XmlOptions.setDocumentType(SchemaType)`
To replace the document element with the specified QName when constructing the resulting document.	`XmlOptions.setLoadReplaceDocumentElement(QName)`
To strip all insignificant whitespace when constructing a document.	`XmlOptions.setLoadStripWhitespace()`
To strip all comments when constructing a document.	`XmlOptions.setLoadStripComments()`
To strip all processing instructions	`XmlOptions.setLoadStripProcinsts()`

when constructing a document.	
A map of namespace URI substitutions to use when constructing a document.	`XmlOptions.setLoadSubstituteNamespaces(Map)`
Additional namespace mappings to be added when constructing a document.	`XmlOptions.setLoadAdditionalNamespaces(Map)`
To trim the underlying XML text buffer immediately after constructing a document, resulting in a smaller memory footprint.	`XmlOptions.setLoadTrimTextBuffer()`
Whether value facets should be checked as they are set.	`XmlOptions.setValidateOnSet()`

Parameters

query
The XQuery expression.

options
Options as described.

---

floatValue() Method

`public float floatValue()`

Description copied from `SimpleValue.floatValue()`
Returns the value as a float. *

---

gDateValue() Method

`public GDate gDateValue()`

Description copied from `SimpleValue.gDateValue()`
Returns the value as a `GDate`. *

---

gDurationValue() Method

`public GDuration gDurationValue()`

Description copied from `SimpleValue.gDurationValue()`
Returns the value as a `GDuration`. *

---

getBigDecimalValue() Method

`public BigDecimal getBigDecimalValue()`

Description copied from `SimpleValue.getBigDecimalValue()`
Returns the value as a `BigDecimal`.

---

getBigIntegerValue() Method

`public BigInteger getBigIntegerValue()`

Description copied from `SimpleValue.getBigIntegerValue()`
Returns the value as a `BigInteger`.

---

getBooleanValue() Method

`public boolean getBooleanValue()`

Description copied from `SimpleValue.getBooleanValue()`
Returns the value as a boolean.

---

getByteArrayValue() Method

`public byte[] getByteArrayValue()`

Description copied from `SimpleValue.getByteArrayValue()`
Returns the value as a byte array.

---

getByteValue() Method

`public byte getByteValue()`

Description copied from `SimpleValue.getByteValue()`
Returns the value as a byte.

---

getCalendarValue() Method

`public Calendar getCalendarValue()`

Description copied from `SimpleValue.getCalendarValue()`
Returns the value as a `Calendar`.

---

getDateValue() Method

`public Date getDateValue()`

Description copied from `SimpleValue.getDateValue()`
Returns the value as a `Date`.

---

getDoubleValue() Method

`public double getDoubleValue()`

Description copied from `SimpleValue.getDoubleValue()`
Returns the value as a double.

---

getEnumValue() Method

`public StringEnumAbstractBase getEnumValue()`

Description copied from `SimpleValue.getEnumValue()`
Returns the value as a `StringEnumAbstractBase`.

---

getFloatValue() Method

`public float getFloatValue()`

Description copied from `SimpleValue.getFloatValue()`
Returns the value as a float.

---

getGDateValue() Method

`public GDate getGDateValue()`

Description copied from `SimpleValue.getGDateValue()`
Returns the value as a `GDate`.

---

getGDurationValue() Method

`public GDuration getGDurationValue()`

Description copied from `SimpleValue.getGDurationValue()`
Returns the value as a `GDuration`.

---

getIntValue() Method

`public int getIntValue()`

Description copied from `SimpleValue.getIntValue()`
Returns the value as an int.

---

getListValue() Method

`public List getListValue()`

Description copied from `SimpleValue.getListValue()`
Returns the value as a `List` of friendly Java objects (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration).

---

getLongValue() Method

`public long getLongValue()`

Description copied from `SimpleValue.getLongValue()`
Returns the value as a long.

---

getObjectValue() Method

`public Object getObjectValue()`

Description copied from `SimpleValue.getObjectValue()`
Returns a union value as a its natural friendly Java object (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration).

---

getQNameValue() Method

`public QName getQNameValue()`

Description copied from `SimpleValue.getQNameValue()`
Returns the value as a `QName`.

---

getShortValue() Method

`public short getShortValue()`

Description copied from `SimpleValue.getShortValue()`
Returns the value as a short.

---

getStringValue() Method

`public String getStringValue()`

Description copied from `SimpleValue.getStringValue()`
Returns the value as a `String`.

---

instanceType() Method

`public SchemaType instanceType()`

Description copied from `SimpleValue.instanceType()`
The same as getSchemaType unless this is a union instance or nil value.

For unions, this returns the non−union consituent type of this instance. This type may change if setters are called that cause the instance to change to another constituent type of the union.

For nil values, this returns null.

---

intValue() Method

`public int intValue()`

Description copied from `SimpleValue.intValue()`
Returns the value as an int. *

---

isImmutable() Method

`public boolean isImmutable()`

Description copied from `XmlObject.isImmutable()`
True if the value is an immutable value. Immutable values do not have a position in a tree; rather, they are stand−alone simple type values. If the object is immutable, the equals() methods tests for value equality, and the object can be used as the key for a hash.

---

isNil() Method

`public boolean isNil()`

Description copied from `XmlObject.isNil()`
True if the value is nil. Note that in order to be nil, the value must be in an element, and the element containing the value must be marked as nillable in the schema.

---

listValue() Method

`public List listValue()`

Description copied from `SimpleValue.listValue()`
Returns the value as a `List` of friendly Java objects (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration). *

---

longValue() Method

`public long longValue()`

Description copied from `SimpleValue.longValue()`
Returns the value as a long. *

---

monitor() Method

`public Object monitor()`

Description copied from `XmlTokenSource.monitor()`
Returns the synchronization object for the document. If concurrent multithreaded access to a document is required, the access should should be protected by synchronizing on this monitor() object. There is one monitor per XML document tree.

---

newCursor() Method

`public XmlCursor newCursor()`

Description copied from `XmlTokenSource.newCursor()`
Returns a new XML cursor. A cursor provides random access to all the tokens in the XML data, plus the ability to extract strongly−typed XmlObjects for the data. If the data is not read−only, the XML cursor also allows modifications to the data. Using a cursor for the first time typically forces the XML document into memory.

---

newDomNode() Method

`public Node newDomNode()`

Description copied from `XmlTokenSource.newDomNode()`
Returns a W3C DOM Node containing the XML represented by this source. This is a copy of the XML, it is not a live with the underlying store of this token source. If this is the document node, then a Document is returned, else a DocumentFragment is returned.

---

newDomNode(XmlOptions) Method

public Node newDomNode(XmlOptions *options*)

Description copied from `XmlTokenSource.newDomNode(XmlOptions)`
Just like newDomNode() but with options. Options map may be null.

---

newInputStream() Method

public InputStream newInputStream()

Description copied from `XmlTokenSource.newInputStream()`
Returns a new stream containing standard XML text, encoded according to the given encoding. The byte stream contains contents starting at the current begin−tag or begin−document and ending at the matching end−tag or end−document. The specified encoding is used and also emitted in a PI at the beginning of the stream. This is a fail−fast stream, so if the underlying data is changed while the stream is being read, the stream throws a ConcurrentModificationException. Throws an IllegalStateException if the XmlTokenSource is not positioned at begin−tag or begin−document (e.g., if it is at an attribute).

---

newInputStream(XmlOptions) Method

public InputStream newInputStream(XmlOptions *options*)

Description copied from `XmlTokenSource.newInputStream(XmlOptions)`
Just like newInputStream(String encoding) but with options. Options map may be null.

---

newReader() Method

public Reader newReader()

Description copied from `XmlTokenSource.newReader()`
Returns a new character reader containing XML text. The contents of the reader represents the document contents starting at the current begin−tag or begin−document and ending at the matching end−tag or end−document. No encoding annotation will be made in the text itself. This is a fail−fast reader, so if the underlying data is changed while the reader is being read, the reader throws a ConcurrentModificationException. Throws an IllegalStateException if the XmlTokenSource is not positioned at begin−tag or begin−document (e.g., if it is at an attribute).

---

newReader(XmlOptions) Method

public Reader newReader(XmlOptions *options*)

Description copied from `XmlTokenSource.newReader(XmlOptions)`
Just like newReader() but with options. Options map may be null.

---

newXMLInputStream() Method

DEPRECATED Superceded by JSR 173

`public XMLInputStream newXMLInputStream()`

Description copied from `com.bea.xml.XmlTokenSource.newXMLInputStream()`
Returns a new XmlInputStream. The stream starts at the current begin−tag or begin−document position and ends at the matching end−tag or end−document. This is a fail−fast stream, so if the underlying data is changed while the stream is being read, the stream throws a ConcurrentModificationException. Throws an IllegalStateException if the XmlTokenSource is not positioned at begin−tag or begin−document (e.g., if it is at an attribute).

---

newXMLInputStream(XmlOptions) Method

DEPRECATED Superceded by JSR 173

`public XMLInputStream newXMLInputStream(XmlOptions` *`options`***`)`**

Description copied from
`com.bea.xml.XmlTokenSource.newXMLInputStream(com.bea.xml.XmlOptions)`

Just like newXMLInputStream() but with any of a number of options. Use the *options* parameter to specify the following:

To specify this	Use this method
The character encoding to use when converting the character data in the XML to bytess.	`XmlOptions.setCharacterEncoding(String)`
Prefix−to−namespace mappings that should be assumed when saving this XML. This is useful when the resulting XML will be part of a larger XML document, ensuring that this inner document will take advantage of namespaces defined in the outer document.	`XmlOptions.setSaveImplicitNamespaces(Map)`
Suggested namespace prefixes to use when saving. Used only when a namespace attribute needs to be synthesized.	`XmlOptions.setSaveSuggestedPrefixes(Map)`
That namespace attributes should occur first in elements when the XML is saved. By default, they occur last.	`XmlOptions.setSaveNamespacesFirst()`
The XML should be pretty printed when saved. Note that this should only be used for debugging.	`XmlOptions.setSavePrettyPrint()`

The number of spaces to use when indenting for pretty printing. The default is 2.	`XmlOptions.setSavePrettyPrintIndent(int)`
The additional number of spaces indented from the left for pretty printed XML.	`XmlOptions.setSavePrettyPrintOffset(int)`
To minimize the number of namespace attributes generated for the saved XML. Note that this can reduce performance significantly.	`XmlOptions.setSaveAggresiveNamespaces()`
To reduce the size of the saved document by allowing the use of the default namespace. Note that this can potentially change the semantic meaning of the XML if unprefixed QNames are present as the value of an attribute or element.	`XmlOptions.setUseDefaultNamespace()`
To filter out processing instructions with the specified target name.	`XmlOptions.setSaveFilterProcinst(String)`
Change the QName of the synthesized root element when saving. This replaces "xml−fragment" with "fragment" in the namespace http://www.openuri.org/fragment	`XmlOptions.setSaveUseOpenFrag()`
Saving should begin on the element's contents.	`XmlOptions.setSaveInner()`
Saving should begin on the element, rather than its contents.	`XmlOptions.setSaveOuter()`
To rename the document element, or to specify the document element for this XML.	`XmlOptions.setSaveSyntheticDocumentElement(QName)`

Parameters

options
Any of the described options.

Returns

A new validating XMLInputStream.

---

objectSet(Object) Method

`public void objectSet(Object` *`obj`***`)`**

Description copied from `SimpleValue.objectSet(Object)`
Sets the value as an arbitrary `Object`.

objectValue() Method

```
public Object objectValue()
```

Description copied from `SimpleValue.objectValue()`
Returns a union value as a its natural friendly Java object (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration). *

qNameValue() Method

```
public QName qNameValue()
```

Description copied from `SimpleValue.qNameValue()`
Returns the value as a `QName`. *

save(ContentHandler, LexicalHandler) Method

```
public void save(ContentHandler ch,
                 LexicalHandler lh)
     throws SAXException
```

Description copied from `XmlTokenSource.save(ContentHandler, LexicalHandler)`
Writes the XML represented by this source to the given SAX content and lexical handlers.

Exceptions

`SAXException`

save(File) Method

```
public void save(File file)
     throws IOException
```

Description copied from `XmlTokenSource.save(File)`
Writes the XML represented by this source to the given File.

Exceptions

`IOException`

save(OutputStream) Method

```
public void save(OutputStream os)
     throws IOException
```

Description copied from `XmlTokenSource.save(OutputStream)`
Writes the XML represented by this source to the given output stream.

Exceptions

`IOException`

---

save(Writer) Method

```
public void save(Writer w)
     throws IOException
```

Description copied from `XmlTokenSource.save(Writer)`
Writes the XML represented by this source to the given writer.

Exceptions

`IOException`

---

save(ContentHandler, LexicalHandler, XmlOptions) Method

```
public void save(ContentHandler ch,
                 LexicalHandler lh,
                 XmlOptions options)
     throws SAXException
```

Description copied from `XmlTokenSource.save(ContentHandler, LexicalHandler, XmlOptions)`
Writes the XML represented by this source to the given SAX content and lexical handlers.

Exceptions

`SAXException`

---

save(File, XmlOptions) Method

```
public void save(File file,
                 XmlOptions options)
     throws IOException
```

Description copied from `XmlTokenSource.save(File, XmlOptions)`
Writes the XML represented by this source to the given File.

Exceptions

`IOException`

---

save(OutputStream, XmlOptions) Method

```
public void save(OutputStream os,
                 XmlOptions options)
     throws IOException
```

Description copied from `XmlTokenSource.save(OutputStream, XmlOptions)`
Writes the XML represented by this source to the given output stream.

Exceptions

IOException

---

save(Writer, XmlOptions) Method

```
public void save(Writer w,
                 XmlOptions options)
     throws IOException
```

Description copied from `XmlTokenSource.save(Writer, XmlOptions)`
Writes the XML represented by this source to the given writer.

Exceptions

IOException

---

schemaType() Method

```
public SchemaType schemaType()
```

Description copied from `XmlObject.schemaType()`
The schema type for this instance. This is a permanent, unchanging property of the instance.

---

selectPath(String) Method

```
public XmlObject[] selectPath(String path)
```

Description copied from `XmlObject.selectPath(String)`
Selects a path. Path can be a string or precompiled path String.

The path must be a relative path, where "." represents the element or attribute containg this XmlObject, and it must select only other elements or attributes. If a non−element or non−attribute is selected, an unchecked exception is thrown.

The array that is returned contains all the selected XmlObjects, within the same document, listed in document order. The actual array type of the result is inferred from the closest common base type of selected results.

Here is an example of usage. Suppose we have a global element definition for "owner" whose type is "person":

```
<schema targetNamespace="http://openuri.org/sample">
   <element name="owner" type="person"/>
   <complexType name="person">
      [...]
   </complexType>
</schema>
```

and suppose "owner" tags can be scattered throughout the document. Then we can write the following code to find them all:

```
import org.openuri.sample.Person;
import com.bea.xml.*;
[...]
  XmlObject xobj = XmlObject.Factory.parse(myFile);
  Person[] results;
  results = (Person[])xobj.selectPath(
     "declare namespace s='http://www.openuri.org/sample' " +
     ".//s:owner");
```

Notice the way in which namespace declarations are done in XPath 2.0. Since XPath can only navigate within an XML document − it cannot construct new XML − the resulting XmlObjects all reside in the same XML document as this XmlObject itself.

---

selectPath(String, XmlOptions) Method

public XmlObject[] selectPath(String *path*,
XmlOptions *options***)**

Description copied from `XmlObject.selectPath(String, XmlOptions)`
Selects a path, applying options.

---

set(XmlObject) Method

public XmlObject set(XmlObject *srcObj***)**

Description copied from `XmlObject.set(XmlObject)`
Set the value/type of this XmlObject to be a copy of the source XmlObject. Because the type of the source may be different than this target, this XmlObject may become defunct. In this case the new XmlObject is returned. If no type change happens, the same this will be returned.

---

set(String) Method

public void set(String *obj***)**

Description copied from `SimpleValue.set(String)`

Sets the value as a `String`. *

---

set(boolean) Method

`public void set(boolean v)`

Description copied from `SimpleValue.set(boolean)`
Sets the value as a boolean. *

---

set(byte) Method

`public void set(byte v)`

Description copied from `SimpleValue.set(byte)`
Sets the value as a byte.

---

set(short) Method

`public void set(short v)`

Description copied from `SimpleValue.set(short)`
Sets the value as a short.

---

set(int) Method

`public void set(int v)`

Description copied from `SimpleValue.set(int)`
Sets the value as an int.

---

set(long) Method

`public void set(long v)`

Description copied from `SimpleValue.set(long)`
Sets the value as a long.

---

set(BigInteger) Method

`public void set(BigInteger obj)`

Description copied from `SimpleValue.set(BigInteger)`
Sets the value as a `BigInteger`.

---

set(BigDecimal) Method

`public void set(BigDecimal obj)`

Description copied from `SimpleValue.set(BigDecimal)`
Sets the value as a `BigDecimal`

---

set(float) Method

`public void set(float v)`

Description copied from `SimpleValue.set(float)`
Sets the value as a float.

---

set(double) Method

`public void set(double v)`

Description copied from `SimpleValue.set(double)`
Sets the value as a double.

---

set(byte[]) Method

`public void set(byte[] obj)`

Description copied from `SimpleValue.set(byte[])`
Sets the value as a byte array.

---

set(StringEnumAbstractBase) Method

`public void set(StringEnumAbstractBase obj)`

Description copied from `SimpleValue.set(StringEnumAbstractBase)`
Sets the value as a `StringEnumAbstractBase`.

---

set(Calendar) Method

`public void set(Calendar obj)`

Description copied from `SimpleValue.set(Calendar)`
Sets the value as a `Calendar`.

---

set(Date) Method

`public void set(Date obj)`

Description copied from `SimpleValue.set(Date)`
Sets the value as a `Date`.

---

set(GDateSpecification) Method

`public void set(GDateSpecification obj)`

Description copied from `SimpleValue.set(GDateSpecification)`
Sets the value as a `GDate`.

---

set(GDurationSpecification) Method

`public void set(GDurationSpecification obj)`

Description copied from `SimpleValue.set(GDurationSpecification)`
Sets the value as a `GDuration`.

---

set(QName) Method

`public void set(QName obj)`

Description copied from `SimpleValue.set(QName)`
Sets the value as a `QName`.

---

set(List) Method

`public void set(List obj)`

Description copied from `SimpleValue.set(List)`
Sets the value as a `List`.

---

setBigDecimalValue(BigDecimal) Method

`public void setBigDecimalValue(BigDecimal obj)`

Description copied from `SimpleValue.setBigDecimalValue(BigDecimal)`
Sets the value as a `BigDecimal`.

---

setBigIntegerValue(BigInteger) Method

`public void setBigIntegerValue(BigInteger obj)`

Description copied from `SimpleValue.setBigIntegerValue(BigInteger)`
Sets the value as a `BigInteger`.

---

setBooleanValue(boolean) Method

`public void setBooleanValue(boolean v)`

Description copied from `SimpleValue.setBooleanValue(boolean)`
Sets the value as a boolean.

---

setByteArrayValue(byte[]) Method

`public void setByteArrayValue(byte[] obj)`

Description copied from `SimpleValue.setByteArrayValue(byte[])`
Sets the value as a byte array.

---

setByteValue(byte) Method

`public void setByteValue(byte v)`

Description copied from `SimpleValue.setByteValue(byte)`
Sets the value as a byte.

---

setCalendarValue(Calendar) Method

`public void setCalendarValue(Calendar obj)`

Description copied from `SimpleValue.setCalendarValue(Calendar)`
Sets the value as a `Calendar`.

---

setDateValue(Date) Method

`public void setDateValue(Date obj)`

Description copied from `SimpleValue.setDateValue(Date)`
Sets the value as a `Date`.

---

setDoubleValue(double) Method

`public void setDoubleValue(double v)`

Description copied from `SimpleValue.setDoubleValue(double)`
Sets the value as a double.

---

setEnumValue(StringEnumAbstractBase) Method

`public void setEnumValue(StringEnumAbstractBase obj)`

Description copied from `SimpleValue.setEnumValue(StringEnumAbstractBase)`
Sets the value as a `StringEnumAbstractBase`.

---

setFloatValue(float) Method

`public void setFloatValue(float v)`

Description copied from `SimpleValue.setFloatValue(float)`
Sets the value as a float.

---

setGDateValue(GDate) Method

`public void setGDateValue(GDate obj)`

Description copied from `SimpleValue.setGDateValue(GDate)`
Sets the value as a `GDate`.

---

setGDurationValue(GDuration) Method

`public void setGDurationValue(GDuration obj)`

Description copied from `SimpleValue.setGDurationValue(GDuration)`
Sets the value as a `GDuration`.

---

setIntValue(int) Method

`public void setIntValue(int v)`

Description copied from `SimpleValue.setIntValue(int)`
Sets the value as an int.

---

setListValue(List) Method

`public void setListValue(List obj)`

Description copied from `SimpleValue.setListValue(List)`
Sets the value as a `List`.

---

setLongValue(long) Method

`public void setLongValue(long v)`

Description copied from `SimpleValue.setLongValue(long)`
Sets the value as a long.

---

setNil() Method

`public void setNil()`

Description copied from `XmlObject.setNil()`
Sets the value to nil. The element containing the value must be marked as nillable in the schema.

---

setObjectValue(Object) Method

`public void setObjectValue(Object obj)`

Description copied from `SimpleValue.setObjectValue(Object)`
Sets the value as an arbitrary `Object`.

---

setQNameValue(QName) Method

`public void setQNameValue(QName obj)`

Description copied from `SimpleValue.setQNameValue(QName)`
Sets the value as a `QName`.

---

setShortValue(short) Method

`public void setShortValue(short v)`

Description copied from `SimpleValue.setShortValue(short)`
Sets the value as a short.

---

setStringValue(String) Method

`public void setStringValue(String obj)`

Description copied from `SimpleValue.setStringValue(String)`
Sets the value as a `String`.

---

shortValue() Method

`public short shortValue()`

Description copied from `SimpleValue.shortValue()`
Returns the value as a short. *

---

stringValue() Method

`public String stringValue()`

Description copied from `SimpleValue.stringValue()`
Returns the value as a `String`. *

---

underlyingXmlObject() Method

`public abstract XmlObject underlyingXmlObject()`

This abstract method is called to obtain the underlying XmlObject. Override this method to supply or compute the wrapped object.

Every other method of this class delegates to the object returned from this method. It is assumed that the object implements all the methods of both XmlObject and SimpleValue.

---

validate() Method

`public boolean validate()`

Description copied from `XmlObject.validate()`
Returns true if the contents of this object are valid accoring to schemaType().

Does a deep validation of the entire subtree under the object, but does not validate the parents or siblings of the object if the object is in the interior of an xml tree.

---

validate(XmlOptions) Method

`public boolean validate(XmlOptions options)`

Description copied from `XmlObject.validate(XmlOptions)`

Just like validate(), but with options.

If you wish to collect error messages and locations while validating, use the `XmlOptions.setErrorListener(Collection)` method. With that method, you can specify an object in which to store messages related to validation. The following is a simple example.

```
// Create an XmlOptions instance and set the error listener.
XmlOptions validateOptions = new XmlOptions();
ArrayList errorList = new ArrayList();
validateOptions.setErrorListener(errorList);

// Validate the XML.
boolean isValid = newEmp.validate(validateOptions);

// If the XML isn't valid, loop through the listener's contents,
// printing contained messages.
if (!isValid)
{
     for (int i = 0; i <errorList.size(); i++)
     {
         XmlError error = (XmlError)errorList.get(i);

         System.out.println("\n");
         System.out.println("Message: " + error.getMessage() + "\n");
         System.out.println("Location of invalid XML: " +
             error.getCursorLocation().xmlText() + "\n");
     }
}
```

Parameters

options
An object that implements the `Collection` interface.

---

valueEquals(XmlObject) Method

`public boolean valueEquals(XmlObject` *`obj`*`)`

Description copied from `XmlObject.valueEquals(XmlObject)`
True if the xml values are equal. Two different objects (which are distinguished by equals(obj) == false) may of course have equal values (valueEquals(obj) == true).

Usually this method can be treated as an ordinary equvalence relation, but actually it is not is not transitive. Here is a precise specification:

There are two categories of XML object: objects with a known instance type, and objects whose only known type is one of the ur−types (either AnyType or AnySimpleType). The first category is compared in terms of logical value spaces, and the second category is compared lexically.

Within each of these two categories, valueEquals is a well−behaved equivalence relation. However, when

comparing an object of known type with an object with ur−type, the comparison is done by attempting to convert the lexical form of the ur−typed object into the other type, and then comparing the results. Ur−typed objects are therefore treated as lexical wildcards and may be equal to objects in different value spaces, even though the objects in different value spaces are not equal to each other.

For example, the anySimpleType value "1" will compare as an equalValue to the string "1", the float value "1.0", the double value "1.0", the decimal "1", and the GYear "1", even though all these objects will compare unequal to each other since they lie in different value spaces.

---

valueHashCode() Method

`public int valueHashCode()`

---

xgetListValue() Method

`public List xgetListValue()`

Description copied from `SimpleValue.xgetListValue()`
Returns the value as a `List` of XmlAnySimpleType objects.

---

xlistValue() Method

`public List xlistValue()`

Description copied from `SimpleValue.xlistValue()`
Returns the value as a `List` of XmlAnySimpleType objects. *

---

xmlText() Method

`public String xmlText()`

Description copied from `XmlTokenSource.xmlText()`
Returns standard XML text. The text returned represents the document contents starting at the current begin−tag or begin−document and ending at the matching end−tag or end−document. This is same content as newReader, but it is returned as a single string. Throws an IllegalStateException if the XmlTokenSource is not positioned at begin−tag or begin−document (e.g., if it is at an attribute).

---

xmlText(XmlOptions) Method

`public String xmlText(XmlOptions` *`options`***`)`**

Description copied from `XmlTokenSource.xmlText(XmlOptions)`
Just like xmlText() but with options. Options map may be null.

com.bea.xml

GDate Class

public final class ***GDate***

extends `Object`
implements `GDateSpecification`, `Serializable`

Represents an XML Schema−compatible Gregorian date.

There are many date types in XML Schema, and this type represents the natural union of all those types. A GDate can hold any subset of date fields (Year, Month, Day, Time, Timezone, or some combination). Wherever the specification provides guidance, the guidelines in the XML Schema 1.0 specification (plus published errata) are followed.

Instances may separately have values or no values for the year, month, day−of−month, and time−of−day. Not all operations are meaningful on all combinations.

Hierarchy

```
Object
 GDate
```

All Implemented Interfaces

`GDateSpecification`, `Serializable`

Constructor Summary

GDate*(`GDateSpecification` gdate)*

Constructs a GDate based on another GDateSpecification.

GDate*(int year, int month, int day, int hour, int minute, int second, `BigDecimal` fraction, int tzSign, int tzHour, int tzMinute)*

Constructs an absolute GDate with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, and in the timezone specified.

GDate*(int year, int month, int day, int hour, int minute, int second, `BigDecimal` fraction)*

Constructs a GDate with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, in an unspecified timezone.

GDate(`CharSequence` *string)*

Constructs a GDate based on a lexical representation.

GDate(`Calendar` *calendar)*

Constructs a GDate based on a java.util.Calendar.

GDate(`Date` *date)*

Constructs a GDate based on a java.util.Date.

Method Summary

public GDate	***add***(`GDurationSpecification` *duration)* Adds a duration to this GDate, and returns a new GDate.
public String	***canonicalString***() The canonical string representation.
public int	***compareToGDate***(`GDateSpecification` *datespec)* Comparison to another GDate.
public boolean	***equals***(`Object` *obj)* GDate is an immutable class, and equality is computed based on its canonical value.
public int	***getBuiltinTypeCode***() Returns the builtin type code for the shape of the information contained in this instance, or 0 if the instance doesn't contain information corresponding to a Schema type.
public XmlCalendar	***getCalendar***() Retrieves the value of the current time as an `XmlCalendar`.
public Date	***getDate***() Retrieves the value of the current time as a java.util.Date instance.
public final int	***getDay***() Gets the day−of−month.
public int	***getFlags***() Returns a combination of flags indicating the information contained by this GDate.
public final BigDecimal	***getFraction***() Gets the fraction−of−second.

public final int ***getHour()***
Gets the hour−of−day.

public int ***getJulianDate()***
Returns the Julian date corresponding to this Gregorian date.

public int ***getMillisecond()***
Gets the rounded millisecond value.

public final int ***getMinute()***
Gets the minute−of−hour.

public final int ***getMonth()***
Gets the month−of−year.

public final int ***getSecond()***
Gets the second−of−minute.

public final int ***getTimeZoneHour()***
Gets the time zone hour.

public final int ***getTimeZoneMinute()***
Gets the time zone minutes.

public final int ***getTimeZoneSign()***
Gets the time zone sign.

public final int ***getYear()***
Gets the year.

public final boolean ***hasDate()***
True if this date/time specification specifies a full date (year, month, day)

public final boolean ***hasDay()***
True if this date/time specification specifies a day−of−month.

public int ***hashCode()***
Returns a hash code for this GDate.

public final boolean ***hasMonth()***
True if this date/time specification specifies a month−of−year.

public final boolean ***hasTime()***
True if this date/time specification specifies a time−of−day.

public final boolean ***hasTimeZone()***
True if this date/time specification specifies a timezone.

public final boolean	***hasYear***() True if this date/time specification specifies a year.
public final boolean	***isImmutable***() Returns true: all GDate instances are immutable.
public boolean	***isValid***() True if this GDate corresponds to a valid gregorian date value in XML schema.
public GDate	***subtract***(`GDurationSpecification` *duration*) Adds a duration to this GDate, and returns a new GDate.
public String	***toString***() The natural string representation.

Methods from class `java.lang.Object`

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Methods from interface `com.bea.xml.GDateSpecification`

`canonicalString, compareToGDate, getBuiltinTypeCode, getCalendar, getDate, getDay, getFlags, getFraction, getHour, getJulianDate, getMillisecond, getMinute, getMonth, getSecond, getTimeZoneHour, getTimeZoneMinute, getTimeZoneSign, getYear, hasDate, hasDay, hasMonth, hasTime, hasTimeZone, hasYear, isImmutable, isValid, toString`

Constructor Detail

GDate

```
public GDate(GDateSpecification gdate)
```

Constructs a GDate based on another GDateSpecification.

GDate

```
public GDate(int year,
             int month,
             int day,
             int hour,
             int minute,
             int second,
             BigDecimal fraction,
             int tzSign,
             int tzHour,
             int tzMinute)
```

Constructs an absolute GDate with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, and in the timezone specified.

If you wish to have a time or date that isn't in a specified timezone, then use the constructor that does not include the timezone arguments.

GDate

```
public GDate(int year,
             int month,
             int day,
             int hour,
             int minute,
             int second,
             BigDecimal fraction)
```

Constructs a GDate with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, in an unspecified timezone.

Note that by not specifying the timezone the GDate becomes partially unordered with respect to times that do have a specified timezone.

GDate

```
public GDate(CharSequence string)
```

Constructs a GDate based on a lexical representation.

GDate

```
public GDate(Calendar calendar)
```

Constructs a GDate based on a java.util.Calendar.

If the calendar does not have some fields set, the same absence of information is reflected in the GDate. Note that java.util.GregorianCalendar fills in all fields as soon as any are fetched, so constructing a GDate with the same calendar object twice may result in a different GDate because of a changed calendar. Note that com.bea.xml.XmlCalendar is stable if you re−get a set field, so it does not have the same problem.

GDate

```
public GDate(Date date)
```

Constructs a GDate based on a java.util.Date.

The current offset of the default timezone is used as the timezone.

For example, if eastern daylight time is in effect at the given date, the timezone on the east coast of the united states translates to GMT−05:00 (EST) + 1:00 (DT offset) == GMT−04:00.

Method Detail

add(GDurationSpecification) Method

`public GDate add(GDurationSpecification duration)`

Adds a duration to this GDate, and returns a new GDate.

---

canonicalString() Method

`public String canonicalString()`

The canonical string representation. Specific moments or times−of−day in a specified timezone are normalized to UTC time to produce a canonical string form for them. Other recurring time specifications keep their timezone information.

---

compareToGDate(GDateSpecification) Method

`public int compareToGDate(GDateSpecification datespec)`

Comparison to another GDate.

- Returns −1 if this < date. (less−than)
- Returns 0 if this == date. (equal)
- Returns 1 if this > date. (greater−than)
- Returns 2 if this date. (incomparable)

Two instances are incomparable if they have different amounts of information.

Parameters

datespec
the date to compare against.

---

equals(Object) Method

`public boolean equals(Object obj)`

GDate is an immutable class, and equality is computed based on its canonical value.

Overrides
`Object.equals(Object)`

getBuiltinTypeCode() Method

`public int getBuiltinTypeCode()`

Returns the builtin type code for the shape of the information contained in this instance, or 0 if the instance doesn't contain information corresponding to a Schema type.

Value will be equal to `SchemaType.BTC_NOT_BUILTIN`, `SchemaType.BTC_G_YEAR`, `SchemaType.BTC_G_YEAR_MONTH`, `SchemaType.BTC_G_MONTH`, `SchemaType.BTC_G_MONTH_DAY`, `SchemaType.BTC_G_DAY`, `SchemaType.BTC_DATE`, `SchemaType.BTC_DATE_TIME`, or `SchemaType.BTC_TIME`.

getCalendar() Method

`public XmlCalendar getCalendar()`

Retrieves the value of the current time as an `XmlCalendar`.

`XmlCalendar` is a subclass of `GregorianCalendar` which is slightly customized to match XML schema date rules.

The returned `XmlCalendar` has only those time and date fields set that are reflected in the GDate object. Because of the way the `Calendar` contract works, any information in the isSet() vanishes as soon as you view any unset field using get() methods. This means that if it is important to understand which date fields are set, you must call isSet() first before get().

getDate() Method

`public Date getDate()`

Retrieves the value of the current time as a java.util.Date instance.

getDay() Method

`public final int getDay()`

Gets the day−of−month. The first day of each month is 1.

getFlags() Method

`public int getFlags()`

Returns a combination of flags indicating the information contained by this GDate. The five flags are HAS_TIMEZONE, HAS_YEAR, HAS_MONTH, HAS_DAY, and HAS_TIME.

getFraction() Method

```
public final BigDecimal getFraction()
```

Gets the fraction−of−second. Range from 0 (inclusive) to 1 (exclusive).

getHour() Method

```
public final int getHour()
```

Gets the hour−of−day. Midnight is 0, and 11PM is 23.

getJulianDate() Method

```
public int getJulianDate()
```

Returns the Julian date corresponding to this Gregorian date. The Julian date (JD) is a continuous count of days from 1 January 4713 BC (= −4712 January 1).

getMillisecond() Method

```
public int getMillisecond()
```

Gets the rounded millisecond value. Range from 0 to 999

getMinute() Method

```
public final int getMinute()
```

Gets the minute−of−hour. Range from 0 to 59.

getMonth() Method

```
public final int getMonth()
```

Gets the month−of−year. January is 1.

getSecond() Method

```
public final int getSecond()
```

Gets the second−of−minute. Range from 0 to 59.

getTimeZoneHour() Method

```
public final int getTimeZoneHour()
```

Gets the time zone hour. This is always positive: for the sign, look at getTimeZoneSign().

getTimeZoneMinute() Method

```
public final int getTimeZoneMinute()
```

Gets the time zone minutes. This is always positive: for the sign, look at getTimeZoneSign().

getTimeZoneSign() Method

```
public final int getTimeZoneSign()
```

Gets the time zone sign. For time zones east of GMT, this is positive; for time zones west, this is negative.

getYear() Method

```
public final int getYear()
```

Gets the year. Should be a four−digit year specification.

hasDate() Method

```
public final boolean hasDate()
```

True if this date/time specification specifies a full date (year, month, day)

hasDay() Method

```
public final boolean hasDay()
```

True if this date/time specification specifies a day−of−month.

hashCode() Method

```
public int hashCode()
```

Returns a hash code for this GDate.

Overrides

`Object.hashCode()`

---

hasMonth() Method

`public final boolean hasMonth()`

True if this date/time specification specifies a month−of−year.

---

hasTime() Method

`public final boolean hasTime()`

True if this date/time specification specifies a time−of−day.

---

hasTimeZone() Method

`public final boolean hasTimeZone()`

True if this date/time specification specifies a timezone.

---

hasYear() Method

`public final boolean hasYear()`

True if this date/time specification specifies a year.

---

isImmutable() Method

`public final boolean isImmutable()`

Returns true: all GDate instances are immutable.

---

isValid() Method

`public boolean isValid()`

True if this GDate corresponds to a valid gregorian date value in XML schema.

---

subtract(GDurationSpecification) Method

`public GDate subtract(GDurationSpecification` *`duration`*`)`

Adds a duration to this GDate, and returns a new GDate.

---

toString() Method

`public String toString()`

The natural string representation. This represents the information that is available, including timezone. For types that correspond to defined schema types (schemaBuiltinTypeCode() > 0), this provides the natural lexical representation.

When both time and timezone are specified, this string is not the canonical representation unless the timezone is UTC (Z) (since the same moment in time can be expressed in different timezones). To get a canonical string, use the canonicalString() method.

Overrides

`Object.toString()`

GDateBuilder Class

public final class ***GDateBuilder***

extends `Object`
implements `GDateSpecification`, `Serializable`

Used to build `GDate`.

Like GDate, a GDateBuilder represents an Gregorian Date, Time, and Timezone, or subset of information (Year, Month, Day, Time, Timezone, or some combination). Wherever it provides guidance, the XML Schema 1.0 specification (plus published errata) is followed.

Instances may separately set or clear the year, month, day−of−month, and time−of−day. Not all operations are meaningful on all combinations. In particular, timezone normalization is only possible if there is a time, or a time together with a full date.

Hierarchy

```
Object
 GDateBuilder
```

All Implemented Interfaces

`GDateSpecification`, `Serializable`

Constructor Summary

GDateBuilder*()*

Constructs a GDateBuilder specifying no date or time

GDateBuilder*(`GDateSpecification` gdate)*

Construts a GDateBuilder by copying another GDateSpecificaiton.

GDateBuilder*(int year, int month, int day, int hour, int minute, int second, `BigDecimal` fraction, int tzSign, int tzHour, int tzMinute)*

Constructs an absolute GDateBuilder with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, and in the timezone specified.

GDateBuilder*(int year, int month, int day, int hour, int minute, int second, `BigDecimal` fraction)*

Constructs a GDateBuilder with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, in an unspecified timezone.

GDateBuilder(`CharSequence` *string)*

Constructs a GDateBuilder from a lexical representation.

GDateBuilder(`Calendar` *calendar)*

GDateBuilder(`Date` *date)*

Constructs a GDateBuilder based on a java.util.Date.

Method Summary

`public void` ***addDuration***(*int sign, int year, int month, int day, int hour, int minute, int second,* `BigDecimal` *fraction)*
Adds a given duration to the date/time.

`public void` ***addGDuration***(`GDurationSpecification` *duration)*
Adds a given duration to the date/time.

`public String` ***canonicalString***()
The canonical string representation.

`public void` ***clearDay***()
Clears the day−of−month.

`public void` ***clearMonth***()
Clears the month−of−year.

`public void` ***clearTime***()
Clears the time−of−day.

`public void` ***clearTimeZone***()
Clears the timezone.

`public void` ***clearYear***()
Clears the year.

`public Object` ***clone***()
Builds another GDateBuilder with the same value as this one.

public final int **compareToGDate**(GDateSpecification datespec)
Comparison to another GDate.

public final int **getBuiltinTypeCode**()
Returns the builtin type code for the shape of the information contained in this instance, or 0 if the instance doesn't contain information corresponding to a Schema type.

public XmlCalendar **getCalendar**()
Retrieves the value of the current time as an XmlCalendar.

public Date **getDate**()
Retrieves the value of the current time as a java.util.Date instance.

public final int **getDay**()
Gets the day−of−month.

public int **getFlags**()
Returns a combination of flags indicating the information contained by this GDate.

public final BigDecimal **getFraction**()
Gets the fraction−of−second.

public final int **getHour**()
Gets the hour−of−day.

public final int **getJulianDate**()
Returns the Julian date corresponding to this Gregorian date.

public final int **getMillisecond**()
Gets the rounded millisecond value.

public final int **getMinute**()
Gets the minute−of−hour.

public final int **getMonth**()
Gets the month−of−year.

public final int **getSecond**()
Gets the second−of−minute.

public final int **getTimeZoneHour**()
Gets the time zone hour.

public final int **getTimeZoneMinute**()
Gets the time zone minutes.

public

`final int` ***getTimeZoneSign()***
Gets the time zone sign.

`public final int` ***getYear()***
Gets the year.

`public final boolean` ***hasDate()***
True if this date/time specification specifies a full date (year, month, day)

`public final boolean` ***hasDay()***
True if this date/time specification specifies a day−of−month.

`public final boolean` ***hasMonth()***
True if this date/time specification specifies a month−of−year.

`public final boolean` ***hasTime()***
True if this date/time specification specifies a time−of−day.

`public final boolean` ***hasTimeZone()***
True if this date/time specification specifies a timezone.

`public final boolean` ***hasYear()***
True if this date/time specification specifies a year.

`public boolean` ***isImmutable()***
True if the instance is immutable.

`public boolean` ***isValid()***
True if all date fields lie within their legal ranges.

`public void` ***normalize()***
Normalizes the instance, ensuring date and time fields are within their normal ranges.

`public void` ***normalizeToTimeZone****(int tzSign, int tzHour, int tzMinute)*
If the time and timezone are known, this method changes the timezone to the specified UTC offset, altering minutes, hours, day, month, and year as necessary to ensure that the actual described moment in time is the same.

`public void` ***normalizeToTimeZone****(int tzTotalMinutes)*
Normalizes to a time zone specified by a number of offset minutes rather than sign/hour/minute; for example, normalizeToTimeZone(−60) is the same as normalizeToTimeZone(−1, 1, 0).

`public void` ***setBuiltinTypeCode****(int typeCode)*
Clears the fields in this GDateBuilder that are not applicable for the given SchemaType date code.

`public void` ***setDate***(`Date` *date*)
Sets the current time and date based on a java.util.Date instance.

`public void` ***setDay***(*int day*)
Sets the day−of−month.

`public void` ***setGDate***(`GDateSpecification` *gdate*)
Copies a GDateSpecification, completely replacing the current information in this GDateBuilder.

`public void` ***setJulianDate***(*int julianday*)
Sets the Gregorian date based on the given Julian date.

`public void` ***setMonth***(*int month*)
Sets the month−of−year.

`public void` ***setTime***(*int hour, int minute, int second,* `BigDecimal` *fraction*)
Sets the time.

`public void` ***setTimeZone***(*int tzSign, int tzHour, int tzMinute*)
Sets the time zone without changing the other time fields.

`public void` ***setTimeZone***(*int tzTotalMinutes*)
Sets the time zone based on a number of offset minutes rather than sign/hour/minute; for example, setTimeZone(−60) is the same as setTimeZone(−1, 1, 0).

`public void` ***setYear***(*int year*)
Sets the year.

`public void` ***subtractGDuration***(`GDurationSpecification` *duration*)
Subtracts a given duration from the date/time.

`public GDate` ***toGDate***()
Builds a GDate from this GDateBuilder.

`public final String` ***toString***()
The natural string representation.

Methods from class `java.lang.Object`

`equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Methods from interface `com.bea.xml.GDateSpecification`

`canonicalString, compareToGDate, getBuiltinTypeCode, getCalendar, getDate, getDay, getFlags, getFraction, getHour, getJulianDate,`

`getMillisecond, getMinute, getMonth, getSecond, getTimeZoneHour, getTimeZoneMinute, getTimeZoneSign, getYear, hasDate, hasDay, hasMonth, hasTime, hasTimeZone, hasYear, isImmutable, isValid, toString`

Constructor Detail

GDateBuilder

```
public GDateBuilder()
```

Constructs a GDateBuilder specifying no date or time

GDateBuilder

```
public GDateBuilder(GDateSpecification gdate)
```

Construts a GDateBuilder by copying another GDateSpecificaiton.

GDateBuilder

```
public GDateBuilder(int year,
                    int month,
                    int day,
                    int hour,
                    int minute,
                    int second,
                    BigDecimal fraction,
                    int tzSign,
                    int tzHour,
                    int tzMinute)
```

Constructs an absolute GDateBuilder with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, and in the timezone specified.

Note that you can reexpress the GDateBuilder in any timezone using normalizeToTimeZone(). The normalize() method normalizes to UTC.

If you wish to have a time or date that isn't in a specified timezone, then use the constructor that does not include the timezone arguments.

GDateBuilder

```
public GDateBuilder(int year,
                    int month,
                    int day,
                    int hour,
                    int minute,
                    int second,
                    BigDecimal fraction)
```

Constructs a GDateBuilder with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, in an unspecified timezone.

Note that by not specifying the timezone the GDateBuilder becomes partially unordered with respect to timesthat do have a specified timezone.

GDateBuilder

```
public GDateBuilder(CharSequence string)
```

Constructs a GDateBuilder from a lexical representation. The lexical space contains the union of the lexical spaces of all the schema date/time types (except for duration).

GDateBuilder

```
public GDateBuilder(Calendar calendar)
```

GDateBuilder

```
public GDateBuilder(Date date)
```

Constructs a GDateBuilder based on a java.util.Date.

The current offset of the default timezone is used as the timezone.

For example, if eastern daylight time is in effect at the given date, the timezone on the east coast of the united states translates to GMT−05:00 (EST) + 1:00 (DT offset) == GMT−04:00.

Method Detail

addDuration(int, int, int, int, int, int, int, BigDecimal) Method

```
public void addDuration(int sign,
                        int year,
                        int month,
                        int day,
                        int hour,
                        int minute,
                        int second,
                        BigDecimal fraction)
```

Adds a given duration to the date/time.

Parameters

sign
: +1 to add, −1 to subtract

year
: the number of years to add

month
: the number of months to add

day
: the number of days to add

hour
: the number of hours to add

minute
: the number of minutes to add

second
: the number of seconds to add

fraction
: the number of fractional seconds to add (may be null)

---

addGDuration(GDurationSpecification) Method

```
public void addGDuration(GDurationSpecification duration)
```

Adds a given duration to the date/time.

Parameters

duration
: the duration to add

---

canonicalString() Method

```
public String canonicalString()
```

The canonical string representation. Specific moments or times−of−day in a specified timezone are normalized to UTC time to produce a canonical string form for them. Other recurring time specifications keep their timezone information.

---

clearDay() Method

`public void clearDay()`

Clears the day−of−month. After clearing. hasDay returns false and the value of getDay is undefined.

---

clearMonth() Method

`public void clearMonth()`

Clears the month−of−year. After clearing. hasMonth returns false and the value of getMonth is undefined.

---

clearTime() Method

`public void clearTime()`

Clears the time−of−day. After clearing. hasTime returns false and the value of getTime is undefined.

---

clearTimeZone() Method

`public void clearTimeZone()`

Clears the timezone. After clearing. hasTimeZone returns false and the value of getTimeZoneHour and getTimeZoneMinute are undefined. Does not change the other time fields.

---

clearYear() Method

`public void clearYear()`

Clears the year. After clearing, hasYear returns false and the value of getYear is undefined.

---

clone() Method

`public Object clone()`

Builds another GDateBuilder with the same value as this one.

Overrides

`Object.clone()`

---

compareToGDate(GDateSpecification) Method

`public final int compareToGDate(GDateSpecification` *`datespec`***`)`**

Comparison to another GDate.

- Returns −1 if this < date. (less−than)
- Returns 0 if this == date. (equal)
- Returns 1 if this > date. (greater−than)
- Returns 2 if this date. (incomparable)

Two instances are incomparable if they have different amounts of information.

Parameters

datespec
the date to compare against

---

getBuiltinTypeCode() Method

`public final int getBuiltinTypeCode()`

Returns the builtin type code for the shape of the information contained in this instance, or 0 if the instance doesn't contain information corresponding to a Schema type.

Value will be equal to `SchemaType.BTC_NOT_BUILTIN`, `SchemaType.BTC_G_YEAR`, `SchemaType.BTC_G_YEAR_MONTH`, `SchemaType.BTC_G_MONTH`, `SchemaType.BTC_G_MONTH_DAY`, `SchemaType.BTC_G_DAY`, `SchemaType.BTC_DATE`, `SchemaType.BTC_DATE_TIME`, or `SchemaType.BTC_TIME`.

---

getCalendar() Method

`public XmlCalendar getCalendar()`

Retrieves the value of the current time as an `XmlCalendar`.

`XmlCalendar` is a subclass of `GregorianCalendar` which is slightly customized to match XML schema date rules.

The returned `XmlCalendar` has only those time and date fields set that are reflected in the GDate object. Because of the way the `Calendar` contract works, any information in the isSet() vanishes as soon as you view any unset field using get() methods. This means that if it is important to understand which date fields are set, you must call isSet() first before get().

---

getDate() Method

`public Date getDate()`

Retrieves the value of the current time as a java.util.Date instance.

---

getDay() Method

```
public final int getDay()
```

Gets the day−of−month. The first day of each month is 1.

---

getFlags() Method

```
public int getFlags()
```

Returns a combination of flags indicating the information contained by this GDate. The five flags are HAS_TIMEZONE, HAS_YEAR, HAS_MONTH, HAS_DAY, and HAS_TIME.

---

getFraction() Method

```
public final BigDecimal getFraction()
```

Gets the fraction−of−second. Range from 0 (inclusive) to 1 (exclusive).

---

getHour() Method

```
public final int getHour()
```

Gets the hour−of−day. Midnight is 0, and 11PM is 23.

---

getJulianDate() Method

```
public final int getJulianDate()
```

Returns the Julian date corresponding to this Gregorian date. The Julian date (JD) is a continuous count of days from 1 January 4713 BC (= −4712 January 1).

---

getMillisecond() Method

```
public final int getMillisecond()
```

Gets the rounded millisecond value. Range from 0 to 999

---

getMinute() Method

```
public final int getMinute()
```

Gets the minute−of−hour. Range from 0 to 59.

---

getMonth() Method

```
public final int getMonth()
```

Gets the month−of−year. January is 1.

---

getSecond() Method

```
public final int getSecond()
```

Gets the second−of−minute. Range from 0 to 59.

---

getTimeZoneHour() Method

```
public final int getTimeZoneHour()
```

Gets the time zone hour. This is always positive: for the sign, look at getTimeZoneSign().

---

getTimeZoneMinute() Method

```
public final int getTimeZoneMinute()
```

Gets the time zone minutes. This is always positive: for the sign, look at getTimeZoneSign().

---

getTimeZoneSign() Method

```
public final int getTimeZoneSign()
```

Gets the time zone sign. For time zones east of GMT, this is positive; for time zones west, this is negative.

---

getYear() Method

```
public final int getYear()
```

Gets the year. Should be a four−digit year specification.

---

hasDate() Method

```
public final boolean hasDate()
```

True if this date/time specification specifies a full date (year, month, day)

---

hasDay() Method

```
public final boolean hasDay()
```

True if this date/time specification specifies a day−of−month.

---

hasMonth() Method

```
public final boolean hasMonth()
```

True if this date/time specification specifies a month−of−year.

---

hasTime() Method

```
public final boolean hasTime()
```

True if this date/time specification specifies a time−of−day.

---

hasTimeZone() Method

```
public final boolean hasTimeZone()
```

True if this date/time specification specifies a timezone.

---

hasYear() Method

```
public final boolean hasYear()
```

True if this date/time specification specifies a year.

---

isImmutable() Method

```
public boolean isImmutable()
```

True if the instance is immutable.

---

isValid() Method

```
public boolean isValid()
```

True if all date fields lie within their legal ranges. A GDateBuilder can be invalid, for example, if you change the month to February and the day−of−month is 31.

---

normalize() Method

```
public void normalize()
```

Normalizes the instance, ensuring date and time fields are within their normal ranges.

If no timezone or no time is specified, or if a partial date is specified, this method does nothing, and leaves the timezone information as−is.

If a time or time and date is specified, this method normalizes the timezone to UTC.

---

normalizeToTimeZone(int, int, int) Method

```
public void normalizeToTimeZone(int tzSign,
                                int tzHour,
                                int tzMinute)
```

If the time and timezone are known, this method changes the timezone to the specified UTC offset, altering minutes, hours, day, month, and year as necessary to ensure that the actual described moment in time is the same.

It is an error to operate on instances without a time or timezone, or with a partially specified date.

Parameters

tzSign
: the timezone offset sign, either +1, 0, or −1

tzHour
: the timezone offset hour

tzMinute
: the timezone offset minute

---

normalizeToTimeZone(int) Method

```
public void normalizeToTimeZone(int tzTotalMinutes)
```

Normalizes to a time zone specified by a number of offset minutes rather than sign/hour/minute; for example, normalizeToTimeZone(−60) is the same as normalizeToTimeZone(−1, 1, 0).

---

setBuiltinTypeCode(int) Method

```
public void setBuiltinTypeCode(int typeCode)
```

Clears the fields in this GDateBuilder that are not applicable for the given SchemaType date code. The code should be `SchemaType.BTC_G_YEAR`, `SchemaType.BTC_G_YEAR_MONTH`, `SchemaType.BTC_G_MONTH`, `SchemaType.BTC_G_MONTH_DAY`, `SchemaType.BTC_G_DAY`,

`SchemaType.BTC_DATE`, `SchemaType.BTC_DATE_TIME`, or `SchemaType.BTC_TIME`.

Parameters

typeCode
: the type code to apply

---

setDate(Date) Method

`public void setDate(Date date)`

Sets the current time and date based on a java.util.Date instance.

The timezone offset used is based on the default TimeZone. (The default TimeZone is consulted to incorporate daylight savings offsets if applicable for the current date as well as the base timezone offset.)

If you wish to normalize the timezone, e.g., to UTC, follow this with a call to normalizeToTimeZone.

Parameters

date
: the Date object to copy

---

setDay(int) Method

`public void setDay(int day)`

Sets the day−of−month. The first day of each month is 1.

Parameters

day
: the day of month, from 1−31

---

setGDate(GDateSpecification) Method

`public void setGDate(GDateSpecification gdate)`

Copies a GDateSpecification, completely replacing the current information in this GDateBuilder.

Parameters

gdate
: the GDateSpecification to copy

---

setJulianDate(int) Method

```
public void setJulianDate(int julianday)
```

Sets the Gregorian date based on the given Julian date. The Julian date (JD) is a continuous count of days from 1 January 4713 BC (= −4712 January 1).

Parameters

julianday
: the julian day number

---

setMonth(int) Method

```
public void setMonth(int month)
```

Sets the month−of−year. January is 1.

Parameters

month
: the month, from 1−12

---

setTime(int, int, int, BigDecimal) Method

```
public void setTime(int hour,
                    int minute,
                    int second,
                    BigDecimal fraction)
```

Sets the time. Hours in the day range from 0 to 23; minutes and seconds range from 0 to 59; and fractional seconds range from 0 (inclusive) to 1 (exclusive). The fraction can be null and is assumed to be zero.

Parameters

hour
: the hour of day, from 0−23

minute
: the minute of hour, from 0−59

second
: the second of minute, from 0−59

fraction
: the fraction of second, 0.0 to 0.999... (may be null)

---

setTimeZone(int, int, int) Method

```
public void setTimeZone(int tzSign,
                        int tzHour,
                        int tzMinute)
```

Sets the time zone without changing the other time fields. If you with to adjust other time fields to express the same actual moment in time in a different time zone, use normalizeToTimeZone.

Timezones must be between −14:00 and +14:00. Sign must be −1 or 1 (or 0 for UTC only), and the offset hours and minute arguments must be nonnegative.

Parameters

tzSign
: the timezone offset sign, either +1, 0, or −1

tzHour
: the timezone offset hour

tzMinute
: the timezone offset minute

---

setTimeZone(int) Method

```
public void setTimeZone(int tzTotalMinutes)
```

Sets the time zone based on a number of offset minutes rather than sign/hour/minute; for example, setTimeZone(−60) is the same as setTimeZone(−1, 1, 0).

---

setYear(int) Method

```
public void setYear(int year)
```

Sets the year. Should be a four−digit year specification.

Parameters

year
: the year

---

subtractGDuration(GDurationSpecification) Method

```
public void subtractGDuration(GDurationSpecification duration)
```

Subtracts a given duration from the date/time.

Parameters

duration
the duration to subtract

---

toGDate() Method

`public GDate toGDate()`

Builds a GDate from this GDateBuilder.

---

toString() Method

`public final String toString()`

The natural string representation. This represents the information that is available, including timezone. For types that correspond to defined schema types (schemaBuiltinTypeCode() > 0), this provides the natural lexical representation.

When both time and timezone are specified, this string is not the canonical representation unless the timezone is UTC (Z) (since the same moment in time can be expressed in different timezones). To get a canonical string, use the canonicalString() method.

Overrides
`Object.toString()`

GDateSpecification Interface

public interface ***GDateSpecification***

Represents an XML Schema−compatible Gregorian date.

Both the immutable GDate and the mutable GDateBuilder are GDateSpecifications. Use this interface where you want to allow callers to pass any implementation of a GDate.

Related Topics

`GDate`
`XmlCalendar`

All Known Implementing Classes
`GDateBuilder`, `GDate`

Field Summary

`public static final int` ***HAS_DAY***
Day of month is specified.

`public static final int` ***HAS_MONTH***
Month of year is specified.

`public static final int` ***HAS_TIME***
Time of day is specified.

`public static final int` ***HAS_TIMEZONE***
Timezone is specified.

`public static final int` ***HAS_YEAR***
Year is specified.

Method Summary

`public String` ***canonicalString****()*
The canonical string representation.

`public int` ***compareToGDate****(GDateSpecification gdatespec)*
Comparison to another GDate.

`public int` ***getBuiltinTypeCode()***
Returns the builtin type code for the shape of the information contained in this instance, or 0 if the instance doesn't contain information corresponding to a Schema type.

`public XmlCalendar` ***getCalendar()***
Retrieves the value of the current time as an `XmlCalendar`.

`public Date` ***getDate()***
Retrieves the value of the current time as a java.util.Date instance.

`public int` ***getDay()***
Gets the day−of−month.

`public int` ***getFlags()***
Returns a combination of flags indicating the information contained by this GDate.

`public BigDecimal` ***getFraction()***
Gets the fraction−of−second.

`public int` ***getHour()***
Gets the hour−of−day.

`public int` ***getJulianDate()***
Returns the Julian date corresponding to this Gregorian date.

`public int` ***getMillisecond()***
Gets the rounded millisecond value.

`public int` ***getMinute()***
Gets the minute−of−hour.

`public int` ***getMonth()***
Gets the month−of−year.

`public int` ***getSecond()***
Gets the second−of−minute.

`public int` ***getTimeZoneHour()***
Gets the time zone hour.

`public int` ***getTimeZoneMinute()***
Gets the time zone minutes.

`public int` ***getTimeZoneSign()***
Gets the time zone sign.

`public int`

`getYear()`
Gets the year.

`public boolean hasDate()`
True if this date/time specification specifies a full date (year, month, day)

`public boolean hasDay()`
True if this date/time specification specifies a day−of−month.

`public boolean hasMonth()`
True if this date/time specification specifies a month−of−year.

`public boolean hasTime()`
True if this date/time specification specifies a time−of−day.

`public boolean hasTimeZone()`
True if this date/time specification specifies a timezone.

`public boolean hasYear()`
True if this date/time specification specifies a year.

`public boolean isImmutable()`
True if this GDate specification is immutable.

`public boolean isValid()`
True if this GDate corresponds to a valid gregorian date value in XML schema.

`public String toString()`
The natural string representation.

Field Detail

HAS_DAY

`public static final int HAS_DAY`

Day of month is specified. See `GDateSpecification.getFlags()`.

HAS_MONTH

`public static final int HAS_MONTH`

Month of year is specified. See `GDateSpecification.getFlags()`.

HAS_TIME

`public static final int HAS_TIME`

Time of day is specified. See `GDateSpecification.getFlags()`.

HAS_TIMEZONE

`public static final int HAS_TIMEZONE`

Timezone is specified. See `GDateSpecification.getFlags()`.

HAS_YEAR

`public static final int HAS_YEAR`

Year is specified. See `GDateSpecification.getFlags()`.

Method Detail

canonicalString() Method

`public String canonicalString()`

The canonical string representation. Specific moments or times−of−day in a specified timezone are normalized to UTC time to produce a canonical string form for them. Other recurring time specifications keep their timezone information.

---

compareToGDate(GDateSpecification) Method

`public int compareToGDate(GDateSpecification` *`gdatespec`***`)`**

Comparison to another GDate.

- Returns −1 if this < date. (less−than)
- Returns 0 if this == date. (equal)
- Returns 1 if this > date. (greater−than)
- Returns 2 if this date. (incomparable)

Two instances are incomparable if they have different amounts of information.

Parameters

gdatespec
the date to compare against.

getBuiltinTypeCode() Method

public int getBuiltinTypeCode()

Returns the builtin type code for the shape of the information contained in this instance, or 0 if the instance doesn't contain information corresponding to a Schema type.

Value will be equal to `SchemaType.BTC_NOT_BUILTIN`, `SchemaType.BTC_G_YEAR`, `SchemaType.BTC_G_YEAR_MONTH`, `SchemaType.BTC_G_MONTH`, `SchemaType.BTC_G_MONTH_DAY`, `SchemaType.BTC_G_DAY`, `SchemaType.BTC_DATE`, `SchemaType.BTC_DATE_TIME`, or `SchemaType.BTC_TIME`.

getCalendar() Method

public XmlCalendar getCalendar()

Retrieves the value of the current time as an `XmlCalendar`.

`XmlCalendar` is a subclass of `GregorianCalendar` which is slightly customized to match XML schema date rules.

The returned `XmlCalendar` has only those time and date fields set that are reflected in the GDate object. Because of the way the `Calendar` contract works, any information in the isSet() vanishes as soon as you view any unset field using get() methods. This means that if it is important to understand which date fields are set, you must call isSet() first before get().

getDate() Method

public Date getDate()

Retrieves the value of the current time as a java.util.Date instance.

getDay() Method

public int getDay()

Gets the day−of−month. The first day of each month is 1.

getFlags() Method

public int getFlags()

Returns a combination of flags indicating the information contained by this GDate. The five flags are `GDateSpecification.HAS_TIMEZONE`, `GDateSpecification.HAS_YEAR`,

`GDateSpecification.HAS_MONTH`, `GDateSpecification.HAS_DAY`, and `GDateSpecification.HAS_TIME`.

---

getFraction() Method

`public BigDecimal getFraction()`

Gets the fraction−of−second. Range from 0 (inclusive) to 1 (exclusive).

---

getHour() Method

`public int getHour()`

Gets the hour−of−day. Midnight is 0, and 11PM is 23.

---

getJulianDate() Method

`public int getJulianDate()`

Returns the Julian date corresponding to this Gregorian date. The Julian date (JD) is a continuous count of days from 1 January 4713 BC (= −4712 January 1).

---

getMillisecond() Method

`public int getMillisecond()`

Gets the rounded millisecond value. Range from 0 to 999

---

getMinute() Method

`public int getMinute()`

Gets the minute−of−hour. Range from 0 to 59.

---

getMonth() Method

`public int getMonth()`

Gets the month−of−year. January is 1.

---

getSecond() Method

```
public int getSecond()
```

Gets the second−of−minute. Range from 0 to 59.

---

getTimeZoneHour() Method

```
public int getTimeZoneHour()
```

Gets the time zone hour. This is always positive: for the sign, look at getTimeZoneSign().

---

getTimeZoneMinute() Method

```
public int getTimeZoneMinute()
```

Gets the time zone minutes. This is always positive: for the sign, look at getTimeZoneSign().

---

getTimeZoneSign() Method

```
public int getTimeZoneSign()
```

Gets the time zone sign. For time zones east of GMT, this is positive; for time zones west, this is negative.

---

getYear() Method

```
public int getYear()
```

Gets the year. Should be a four−digit year specification.

---

hasDate() Method

```
public boolean hasDate()
```

True if this date/time specification specifies a full date (year, month, day)

---

hasDay() Method

```
public boolean hasDay()
```

True if this date/time specification specifies a day−of−month.

---

hasMonth() Method

```
public boolean hasMonth()
```

True if this date/time specification specifies a month−of−year.

---

hasTime() Method

```
public boolean hasTime()
```

True if this date/time specification specifies a time−of−day.

---

hasTimeZone() Method

```
public boolean hasTimeZone()
```

True if this date/time specification specifies a timezone.

---

hasYear() Method

```
public boolean hasYear()
```

True if this date/time specification specifies a year.

---

isImmutable() Method

```
public boolean isImmutable()
```

True if this GDate specification is immutable. GDate returns true, and GDateBuilder returns false.

---

isValid() Method

```
public boolean isValid()
```

True if this GDate corresponds to a valid gregorian date value in XML schema.

---

toString() Method

```
public String toString()
```

The natural string representation. This represents the information that is available, including timezone. For types that correspond to defined schema types (schemaBuiltinTypeCode() > 0), this provides the natural lexical representation. When both time and timezone are specified, this string is not the canonical representation unless the timezone is UTC (Z) (since the same moment in time can be expressed in different

timezones). To get a canonical string, use the canonicalString() method.

Overrides

`Object.toString()`

com.bea.xml

GDuration Class

public final class ***GDuration***

extends `Object`
implements `GDurationSpecification`, `Serializable`

Represents an XML Schema−compatible duration.

A duration is made up of a number of years, months, days, hours, minutes, seconds, and fractions of seconds. See the XML Schema specification section on xs:duration for details on the rules for comparing durations and adding durations to dates.

Hierarchy

```
Object
 GDuration
```

All Implemented Interfaces

`GDurationSpecification`, `Serializable`

Constructor Summary

GDuration()

Constructs an empty GDuration representing zero seconds.

GDuration*(`GDurationSpecification` gDuration)*

Constructs a GDuration from another GDurationSpecification.

GDuration*(int sign, int year, int month, int day, int hour, int minute, int second, `BigDecimal` fraction)*

Constructs a GDuration with the specified sign, year, month, day, hours, minutes, seconds, and optional fractional seconds.

GDuration*(`CharSequence` str)*

Constructs a GDuration from a lexical representation.

Method Summary

public GDuration	***add***(GDurationSpecification *duration*) Returns a new GDuration which is the sum of this one and the supplied duration.
public Object	***clone***() Builds another GDate with the same value as this one.
public final int	***compareToGDuration***(GDurationSpecification *duration*) Comparison to another GDuration.
public boolean	***equals***(Object *obj*) Two GDurations are equal if all their fields are equal.
public final int	***getDay***() Gets the day−of−month component.
public BigDecimal	***getFraction***() Gets the fraction−of−second.
public final int	***getHour***() Gets the hour−of−day component.
public final int	***getMinute***() Gets the minute−of−hour component.
public final int	***getMonth***() Gets the month−of−year component.
public final int	***getSecond***() Gets the second−of−minute component.
public final int	***getSign***() Returns the sign of the duration: +1 is forwards and −1 is backwards in time.
public final int	***getYear***() Gets the year component.
public int	***hashCode***()
public final boolean	***isImmutable***() All GDuration instances return true.

`public boolean` ***isValid***()
Returns true if all of the individual components of the duration are nonnegative.

`public GDuration` ***subtract***(`GDurationSpecification` *duration*)
Returns a new GDuration which is the result of subtracting the supplied duration from this one.

`public String` ***toString***()
The natural string representation of the duration.

Methods from class `java.lang.Object`

`finalize, getClass, notify, notifyAll, wait, wait, wait`

Methods from interface `com.bea.xml.GDurationSpecification`

`compareToGDuration, getDay, getFraction, getHour, getMinute, getMonth, getSecond, getSign, getYear, isImmutable, isValid`

Constructor Detail

GDuration

```
public GDuration()
```

Constructs an empty GDuration representing zero seconds.

GDuration

```
public GDuration(GDurationSpecification gDuration)
```

Constructs a GDuration from another GDurationSpecification.

GDuration

```
public GDuration(int sign,
                 int year,
                 int month,
                 int day,
                 int hour,
                 int minute,
                 int second,
                 BigDecimal fraction)
```

Constructs a GDuration with the specified sign, year, month, day, hours, minutes, seconds, and optional fractional seconds.

GDuration

public GDuration(CharSequence *str*)

Constructs a GDuration from a lexical representation. The lexical space contains the union of the lexical spaces of all the schema date/time types (except for duration).

Method Detail

add(GDurationSpecification) Method

public GDuration add(GDurationSpecification *duration*)

Returns a new GDuration which is the sum of this one and the supplied duration. Does a fieldwise addition, with no normalization.

---

clone() Method

public Object clone()

Builds another GDate with the same value as this one.

Overrides

`Object.clone()`

---

compareToGDuration(GDurationSpecification) Method

public final int compareToGDuration(GDurationSpecification *duration*)

Comparison to another GDuration.

- Returns −1 if this < date. (less−than)
- Returns 0 if this == date. (equal)
- Returns 1 if this > date. (greater−than)
- Returns 2 if this date. (incomparable)

Two instances are incomparable if they have different amounts of information.

---

equals(Object) Method

public boolean equals(Object *obj*)

Two GDurations are equal if all their fields are equal. The equals function does not apply normalizatin.

Overrides

`Object.equals(Object)`

---

getDay() Method

`public final int getDay()`

Gets the day−of−month component.

---

getFraction() Method

`public BigDecimal getFraction()`

Gets the fraction−of−second. Range from 0 (inclusive) to 1 (exclusive).

---

getHour() Method

`public final int getHour()`

Gets the hour−of−day component.

---

getMinute() Method

`public final int getMinute()`

Gets the minute−of−hour component.

---

getMonth() Method

`public final int getMonth()`

Gets the month−of−year component.

---

getSecond() Method

`public final int getSecond()`

Gets the second−of−minute component.

---

getSign() Method

`public final int getSign()`

Returns the sign of the duration: +1 is forwards and −1 is backwards in time.

getYear() Method

`public final int getYear()`

Gets the year component.

hashCode() Method

`public int hashCode()`

Overrides

`Object.hashCode()`

isImmutable() Method

`public final boolean isImmutable()`

All GDuration instances return true.

isValid() Method

`public boolean isValid()`

Returns true if all of the individual components of the duration are nonnegative.

subtract(GDurationSpecification) Method

`public GDuration subtract(GDurationSpecification` *`duration`*`)`

Returns a new GDuration which is the result of subtracting the supplied duration from this one. Does a fieldwise subtraction, with no normalization.

toString() Method

`public String toString()`

The natural string representation of the duration.

Any components that are zero are omitted. Note that if the duration is invalid, i.e., it has negative components, those negative components are serialized out here. To check for validity, use the isValid() method; and to normalize most durations to a valid form use the normalize() method.

Overrides

`Object.toString()`

GDurationBuilder Class

public class ***GDurationBuilder***

extends `Object`
implements `GDurationSpecification`, `Serializable`

Used to build `GDuration`.

Hierarchy

```
Object
 GDurationBuilder
```

All Implemented Interfaces

`GDurationSpecification`, `Serializable`

Constructor Summary

GDurationBuilder*()*

Constructs an empty GDurationBuilder representing zero seconds.

GDurationBuilder*(`GDurationSpecification` gDuration)*

Constructs a GDurationBuilder from another GDurationBuilderSpecification.

GDurationBuilder*(int sign, int year, int month, int day, int hour, int minute, int second, `BigDecimal` fraction)*

Constructs a GDurationBuilder with the specified sign, year, month, day, hours, minutes, seconds, and optional fractional seconds.

GDurationBuilder*(`String` s)*

Constructs a GDuration from a lexical representation.

Method Summary

`public void` ***addGDuration****(`GDurationSpecification` duration)*

Adds to this duration.

`public Object` ***clone()***

Builds another GDurationBuilder with the same value as this one.

`public final int` ***compareToGDuration***`(GDurationSpecification` *duration)*

Comparison to another GDuration.

`public final int` ***getDay()***

Gets the day−of−month component.

`public BigDecimal` ***getFraction()***

Gets the fraction−of−second.

`public final int` ***getHour()***

Gets the hour−of−day component.

`public final int` ***getMinute()***

Gets the minute−of−hour component.

`public final int` ***getMonth()***

Gets the month−of−year component.

`public final int` ***getSecond()***

Gets the second−of−minute component.

`public final int` ***getSign()***

Returns the sign of the duration: +1 is forwards and −1 is backwards in time.

`public final int` ***getYear()***

Gets the year component.

`public final boolean` ***isImmutable()***

All GDuration instances return true.

`public boolean` ***isValid()***

Returns true if all of the individual components of the duration are nonnegative.

`public void` ***normalize()***

Normalize a duration value.

`public void` ***setDay***`(int` *day)*

Sets the day component.

`public void` ***setFraction***`(BigDecimal` *fraction)*

Sets the fraction−of−second component.

`public void` ***setHour***(*int hour*)
Sets the hour component.

`public void` ***setMinute***(*int minute*)
Sets the minute component.

`public void` ***setMonth***(*int month*)
Sets the month component.

`public void` ***setSecond***(*int second*)
Sets the second component.

`public final void` ***setSign***(*int sign*)
Sets the sign.

`public void` ***setYear***(*int year*)
Sets the year component.

`public void` ***subtractGDuration***(`GDurationSpecification` *duration*)
Subtracts from this duration.

`public GDuration` ***toGDuration***()
Builds a GDuration from this GDurationBuilder.

`public String` ***toString***()
The natural string representation of the duration.

Methods from class `java.lang.Object`

`equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Methods from interface `com.bea.xml.GDurationSpecification`

`compareToGDuration, getDay, getFraction, getHour, getMinute, getMonth, getSecond, getSign, getYear, isImmutable, isValid`

Constructor Detail

GDurationBuilder

`public GDurationBuilder()`

Constructs an empty GDurationBuilder representing zero seconds.

GDurationBuilder

```
public GDurationBuilder(GDurationSpecification gDuration)
```

Constructs a GDurationBuilder from another GDurationBuilderSpecification.

GDurationBuilder

```
public GDurationBuilder(int sign,
                        int year,
                        int month,
                        int day,
                        int hour,
                        int minute,
                        int second,
                        BigDecimal fraction)
```

Constructs a GDurationBuilder with the specified sign, year, month, day, hours, minutes, seconds, and optional fractional seconds.

GDurationBuilder

```
public GDurationBuilder(String s)
```

Constructs a GDuration from a lexical representation.

Method Detail

addGDuration(GDurationSpecification) Method

```
public void addGDuration(GDurationSpecification duration)
```

Adds to this duration. Does a fieldwise add, with no normalization.

---

clone() Method

```
public Object clone()
```

Builds another GDurationBuilder with the same value as this one.

Overrides

`Object.clone()`

---

compareToGDuration(GDurationSpecification) Method

`public final int compareToGDuration(GDurationSpecification duration)`

Comparison to another GDuration.

- Returns −1 if this < date. (less−than)
- Returns 0 if this == date. (equal)
- Returns 1 if this > date. (greater−than)
- Returns 2 if this date. (incomparable)

Two instances are incomparable if they have different amounts of information.

---

getDay() Method

`public final int getDay()`

Gets the day−of−month component.

---

getFraction() Method

`public BigDecimal getFraction()`

Gets the fraction−of−second. Range from 0 (inclusive) to 1 (exclusive).

---

getHour() Method

`public final int getHour()`

Gets the hour−of−day component.

---

getMinute() Method

`public final int getMinute()`

Gets the minute−of−hour component.

---

getMonth() Method

`public final int getMonth()`

Gets the month−of−year component.

---

getSecond() Method

```
public final int getSecond()
```

Gets the second−of−minute component.

---

getSign() Method

```
public final int getSign()
```

Returns the sign of the duration: +1 is forwards and −1 is backwards in time. This value does not necessarily reflect the true direction of the duration if the duration is not normalized or not normalizable.

---

getYear() Method

```
public final int getYear()
```

Gets the year component.

---

isImmutable() Method

```
public final boolean isImmutable()
```

All GDuration instances return true.

---

isValid() Method

```
public boolean isValid()
```

Returns true if all of the individual components of the duration are nonnegative.

---

normalize() Method

```
public void normalize()
```

Normalize a duration value. This ensures that months, hours, minutes, seconds, and fractions are positive and within the ranges 0..11, 0..23, 0..59, etc. Negative durations are indicated by a negative sign rather than negative components.

Most duration specifications can be normalized to valid durations with all positive components, but not all of them can.

The only situations which cannot be normalized are where the year/month and the day/hour/minute/second offsets are of opposite sign. Days cannot be carried into months since the length of a Gregorian month is variable depending on when the duration is applied. In these cases, this method normalizes the components so

that "day" is the only negative component.

---

setDay(int) Method

public void setDay(int *day*)

Sets the day component.

---

setFraction(BigDecimal) Method

public void setFraction(BigDecimal *fraction*)

Sets the fraction−of−second component.

---

setHour(int) Method

public void setHour(int *hour*)

Sets the hour component.

---

setMinute(int) Method

public void setMinute(int *minute*)

Sets the minute component.

---

setMonth(int) Method

public void setMonth(int *month*)

Sets the month component.

---

setSecond(int) Method

public void setSecond(int *second*)

Sets the second component.

---

setSign(int) Method

public final void setSign(int *sign*)

Sets the sign.

setYear(int) Method

public void setYear(int *year*)

Sets the year component.

subtractGDuration(GDurationSpecification) Method

public void subtractGDuration(GDurationSpecification *duration*)

Subtracts from this duration. Does a fieldwise subtraction, with no normalization.

toGDuration() Method

public GDuration toGDuration()

Builds a GDuration from this GDurationBuilder.

toString() Method

public String toString()

The natural string representation of the duration.

Any components that are zero are omitted. Note that if the duration is invalid, i.e., it has negative components, those negative components are serialized out here. To check for validity, use the isValid() method; and to normalize most durations to a valid form use the normalize() method.

Overrides

`Object.toString()`

com.bea.xml

GDurationSpecification Interface

public interface ***GDurationSpecification***

Represents an XML Schema−compatible duration.

Both the immutable GDuration and the mutable GDurationBuilder are GDurationSpecifications. Use this interface where you want to allow callers to pass any implementation of a GDuration.

Related Topics

`GDuration`

All Known Implementing Classes
`GDurationBuilder`, `GDuration`

Method Summary

`public int` ***compareToGDuration***(`GDurationSpecification` *duration)*
Comparison to another GDuration.

`public int` ***getDay***()
Gets the day−of−month component.

`public BigDecimal` ***getFraction***()
Gets the fraction−of−second.

`public int` ***getHour***()
Gets the hour−of−day component.

`public int` ***getMinute***()
Gets the minute−of−hour component.

`public int` ***getMonth***()
Gets the month−of−year component.

`public int` ***getSecond***()
Gets the second−of−minute component.

`public int` ***getSign***()
Returns the sign of the duration: +1 is forwards and −1 is backwards in time.

`public int` ***getYear***()

Gets the year component.

`public boolean` ***isImmutable()***

True if this instance is immutable.

`public boolean` ***isValid()***

Returns true if all of the individual components of the duration are nonnegative.

Method Detail

compareToGDuration(GDurationSpecification) Method

`public int compareToGDuration(GDurationSpecification` *`duration`***`)`**

Comparison to another GDuration.

- Returns −1 if this < date. (less−than)
- Returns 0 if this == date. (equal)
- Returns 1 if this > date. (greater−than)
- Returns 2 if this date. (incomparable)

Two instances are incomparable if they have different amounts of information.

---

getDay() Method

`public int getDay()`

Gets the day−of−month component.

---

getFraction() Method

`public BigDecimal getFraction()`

Gets the fraction−of−second. Range from 0 (inclusive) to 1 (exclusive).

---

getHour() Method

`public int getHour()`

Gets the hour−of−day component.

---

getMinute() Method

`public int getMinute()`

Gets the minute−of−hour component.

getMonth() Method

```
public int getMonth()
```

Gets the month−of−year component.

getSecond() Method

```
public int getSecond()
```

Gets the second−of−minute component.

getSign() Method

```
public int getSign()
```

Returns the sign of the duration: +1 is forwards and −1 is backwards in time.

getYear() Method

```
public int getYear()
```

Gets the year component.

isImmutable() Method

```
public boolean isImmutable()
```

True if this instance is immutable.

isValid() Method

```
public boolean isValid()
```

Returns true if all of the individual components of the duration are nonnegative.

com.bea.xml

QNameCache Class

public final class ***QNameCache***

extends `Object`

A cache that can be used to pool QName instances. Each thread has one.

Hierarchy

```
Object
 QNameCache
```

Constructor Summary

QNameCache*(int initialCapacity, float loadFactor)*

Creates a QNameCache with the given initialCapacity and loadFactor.

QNameCache*(int initialCapacity)*

Creates a QNameCache with the given initialCapacity.

Method Summary

`public QName` ***`getName`****(`String` uri, `String` localName)*

Fetches a QName with the given namespace and localname.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

QNameCache

```
public QNameCache(int initialCapacity,
                  float loadFactor)
```

Creates a QNameCache with the given initialCapacity and loadFactor.

QNameCache

```
public QNameCache(int initialCapacity)
```

Creates a QNameCache with the given initialCapacity.

Method Detail

getName(String, String) Method

```
public QName getName(String uri,
                     String localName)
```

Fetches a QName with the given namespace and localname. Creates one if one is not found in the cache.

Parameters

uri
the namespace

localName
the localname

Returns

the cached QName

QNameSet Class

public final class ***QNameSet***

extends `Object`
implements `QNameSetSpecification`, `Serializable`

This interface represents a lattice of finite and infinite sets of QNames. The lattice the minimal one that is closed under union, intersection, and inverse, and contains individual QNames as well as entire namespaces. Here is a summary of the two kinds of QNameSets:

- A QNameSet can cover a finite set of namespaces, additionally including a finite set of QNames outside those namespaces, and with the exception of a finite set of QNames excluded from those namespaes:
  - excludedQNamesInIncludedURIs == the set of excluded QNames from coveredURIs namespaces
  - excludedURIs == null
  - includedURIs == the set of covered namespace URIs
  - includedQNamesInExcludedURIs == set of additional QNames outside coveredURIs namespaces
- A QNameSet can cover all namespaces except for a finite number of excluded ones, additionally including a finite set of QNames within the excluded namespaces, and with the exception of a finite set of QNames outside the excluded namespaces:
  - excludedQNamesInIncludedURIs == the set of excluded QNames outside uncoveredURIs namespaces
  - excludedURIs == the set of uncovered namespace URIs
  - includedURIs == null
  - includedQNamesInExcludedURIs == set of additional QNames from uncoveredURIs namespaces

Notice that a finite set of QNames is a degenerate case of the first category outlined above:

- A QnameSet can contain a finite number of QNames:
  - excludedQNamesInIncludedURIs == null
  - excludedURIs == null
  - includedURIs == empty set
  - includedQNamesInExcludedURIs == set of included QNames

Hierarchy

```
Object
 QNameSet
```

All Implemented Interfaces

`QNameSetSpecification`, `Serializable`

Field Summary

public static final QNameSet	***ALL*** The QNameSet containing all QNames.
public static final QNameSet	***EMPTY*** The empty QNameSet.
public static final QNameSet	***LOCAL*** The QNameSet containing all QNames in the local (no−)namespace.
public static final QNameSet	***NONLOCAL*** The QNameSet containing all QNames except for those in the local (no−)namespace.

Method Summary

public boolean	***contains***(*QName name*) True if this ModelTransitionSet contains the given qname.
public boolean	***containsAll***(*QNameSetSpecification set*) True if the given set is a subset of this one.
public Set	***excludedQNamesInIncludedURIs***() The set of QNames excluded from the set even though they are within a namespace that is otherwise fully included in the set.
public Set	***excludedURIs***() Namespaces that are fully excluded from the set except for a finite number of individual QName exceptions.
public static QNameSet	***forArray***(*QName[] includedQNames*) Returns a QNameSet based on the given array of included QNames
public static QNameSet	***forSets***(*Set excludedURIs, Set includedURIs, Set excludedQNamesInIncludedURIs, Set includedQNamesInExcludedURIs*) Returns a QNameSet based on the given sets of excluded URIs, included URIs, excluded QNames in included namespaces, and included QNames in excluded namespaces.
public static QNameSet	***forSpecification***(*QNameSetSpecification spec*) Returns a QNameSet with the same contents as the given QNameSetSpecification.
public static	***forWildcardNamespaceString***(*String wildcard, String targetURI*)

`QNameSet` Returns a QNameSet corresponding to the given wildcard namespace string.

`public Set` ***includedQNamesInExcludedURIs***()
The set of QNames included in the set even though they are within a namespace that is otherwise fully included in the set.

`public Set` ***includedURIs***()
Namespaces that are fully included in set except for a finite number of individual QName exceptions.

`public QNameSet` ***intersect***(`QNameSetSpecification` *set*)
Returns a new QNameSet that is the intersection of this one and another.

`public QNameSet` ***inverse***()
Returns a new QNameSet that is the inverse of this one.

`public boolean` ***isAll***()
True if this ModelTransitionSet contains all QNames.

`public boolean` ***isDisjoint***(`QNameSetSpecification` *set*)
True if the given set is disjoint from this one.

`public boolean` ***isEmpty***()
True if this ModelTransitionSet contains no QNames.

`public static QNameSet` ***singleton***(`QName` *name*)
Returns a QNameSet containing only the given QName.

`public String` ***toString***()
Returns a string representation useful for debugging, subject to change.

`public QNameSet` ***union***(`QNameSetSpecification` *set*)
Returns a new QNameSet that is the union of this one and another.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Methods from interface `com.bea.xml.QNameSetSpecification`

`contains, containsAll, excludedQNamesInIncludedURIs, excludedURIs, includedQNamesInExcludedURIs, includedURIs, intersect, inverse, isAll, isDisjoint, isEmpty, union`

Field Detail

ALL

public static final QNameSet ALL

The QNameSet containing all QNames.

EMPTY

public static final QNameSet EMPTY

The empty QNameSet.

LOCAL

public static final QNameSet LOCAL

The QNameSet containing all QNames in the local (no−)namespace.

NONLOCAL

public static final QNameSet NONLOCAL

The QNameSet containing all QNames except for those in the local (no−)namespace.

Method Detail

contains(QName) Method

public boolean contains(QName *name*)

True if this ModelTransitionSet contains the given qname.

---

containsAll(QNameSetSpecification) Method

public boolean containsAll(QNameSetSpecification *set*)

True if the given set is a subset of this one.

Parameters

set
the set to test

Returns

true if this contains all QNames contained by the given set

---

excludedQNamesInIncludedURIs() Method

```
public Set excludedQNamesInIncludedURIs()
```

The set of QNames excluded from the set even though they are within a namespace that is otherwise fully included in the set.

Returns

the set of excluded QNames from within includedURI namespaces

---

excludedURIs() Method

```
public Set excludedURIs()
```

Namespaces that are fully excluded from the set except for a finite number of individual QName exceptions. Returns null if this set is infinite.

Returns

the set of excluded namespace URI strings

---

forArray(QName[]) Method

```
public static QNameSet forArray(QName[] includedQNames)
```

Returns a QNameSet based on the given array of included QNames

Parameters

includedQNames
the array of included QNames

---

forSets(Set, Set, Set, Set) Method

```
public static QNameSet forSets(Set excludedURIs,
                               Set includedURIs,
                               Set excludedQNamesInIncludedURIs,
```

```
Set includedQNamesInExcludedURIs)
```

Returns a QNameSet based on the given sets of excluded URIs, included URIs, excluded QNames in included namespaces, and included QNames in excluded namespaces.

Parameters

excludedURIs
: the finite set of namespace URI strings to exclude from the set, or null if this set is infinite

includedURIs
: the finite set of namespace URI strings to include in the set, or null if this set is infinite

excludedQNamesInIncludedURIs
: the finite set of exceptional QNames to exclude from the included namespaces

includedQNamesInExcludedURIs
: the finite set of exceptional QNames to include that are in the excluded namespaces

Returns

the constructed QNameSet

---

forSpecification(QNameSetSpecification) Method

```
public static QNameSet forSpecification(QNameSetSpecification spec)
```

Returns a QNameSet with the same contents as the given QNameSetSpecification.

Returns

the copied QNameSet

---

forWildcardNamespaceString(String, String) Method

```
public static QNameSet forWildcardNamespaceString(String wildcard,
                                                  String targetURI)
```

Returns a QNameSet corresponding to the given wildcard namespace string. This is a space−separated list of URIs, plus special tokens as specified in the XML Schema specification (##any, ##other, ##targetNamespace, ##local).

Returns

the constructed QNameSet

---

includedQNamesInExcludedURIs() Method

`public Set includedQNamesInExcludedURIs()`

The set of QNames included in the set even though they are within a namespace that is otherwise fully included in the set.

Returns

the set of included QNames from within excludedURI namespaces

---

includedURIs() Method

`public Set includedURIs()`

Namespaces that are fully included in set except for a finite number of individual QName exceptions. Returns null if this set is infinite.

Returns

the set of included namespace URI strings

---

intersect(QNameSetSpecification) Method

`public QNameSet intersect(QNameSetSpecification` *`set`***`)`**

Returns a new QNameSet that is the intersection of this one and another.

Parameters

set
the set to insersect with

Returns

the intersection

---

inverse() Method

`public QNameSet inverse()`

Returns a new QNameSet that is the inverse of this one.

---

isAll() Method

`public boolean isAll()`

True if this ModelTransitionSet contains all QNames.

---

isDisjoint(QNameSetSpecification) Method

`public boolean isDisjoint(QNameSetSpecification set)`

True if the given set is disjoint from this one.

Parameters

set
the set to test

Returns

true if the set is disjoint from this set

---

isEmpty() Method

`public boolean isEmpty()`

True if this ModelTransitionSet contains no QNames.

---

singleton(QName) Method

`public static QNameSet singleton(QName name)`

Returns a QNameSet containing only the given QName.

Returns

the constructed QNameSet

---

toString() Method

`public String toString()`

Returns a string representation useful for debugging, subject to change.

Overrides
`Object.toString()`

---

union(QNameSetSpecification) Method

`public QNameSet union(QNameSetSpecification set)`

Returns a new QNameSet that is the union of this one and another.

Parameters

set
the set to union with

Returns

the union

com.bea.xml

QNameSetBuilder Class

public class ***QNameSetBuilder***

extends `Object`
implements `QNameSetSpecification`, `Serializable`

Used to build `QNameSet`.

Hierarchy

```
Object
 QNameSetBuilder
```

All Implemented Interfaces

`QNameSetSpecification`, `Serializable`

Constructor Summary

QNameSetBuilder*()*

Constructs an empty QNameSetBuilder.

QNameSetBuilder*(`QNameSetSpecification` set)*

Constructs a QNameSetBuilder whose initial contents are given by another QNameSetSpecification.

QNameSetBuilder*(`String` str, `String` targetURI)*

Constructs a QNameSetBuilder whose initial contents are given as a list of namespace URIs, using the same format used by wildcards in XSD files.

QNameSetBuilder*(`Set` excludedURIs, `Set` includedURIs, `Set` excludedQNamesInIncludedURIs, `Set` includedQNamesInExcludedURIs)*

Constructs a QNameSetBuilder whose inital contents are given by the four sets.

Method Summary

`public`
`void` ***add****(`QName` qname)*

Adds a single QName to this QNameSetBuilder.

public void **addAll**(QNameSetSpecification *set*)

Adds the contents of another QNameSet to this QNameSetBuilder.

public void **addNamespace**(String *uri*)

Adds an entire namespace URI of QNames to this QNameSetBuilder.

public void **clear**()

Clears this QNameSetBuilder

public boolean **contains**(QName *name*)

True if this ModelTransitionSet contains the given qname.

public boolean **containsAll**(QNameSetSpecification *set*)

True if the parameter is a subset of this set.

public Set **excludedQNamesInIncludedURIs**()

The finite set of QNames that are excluded from the set within namespaces that are otherwise included.

public Set **excludedURIs**()

The finite set of namespace URIs that are almost completely excluded from the set (that is, each namespace URI that included in the set with with a finite number of QName exceptions).

public Set **includedQNamesInExcludedURIs**()

The finite set of QNames that are included in the set within namespaces that are otherwise excluded.

public Set **includedURIs**()

The finite set of namespace URIs that are almost completely included in the set (that is, each namespace URI that included in the set with with a finite number of QName exceptions).

public QNameSet **intersect**(QNameSetSpecification *set*)

Returns a new QNameSet that is the intersection of this one and another.

public QNameSet **inverse**()

Returns a new QNameSet that is the inverse of this one.

public void **invert**()

Inverts this QNameSetBuilder.

public boolean **isAll**()

True if this ModelTransitionSet contains all QNames.

public

`boolean` ***isDisjoint***(`QNameSetSpecification` *set*)
True if the given set is disjoint from this one.

`public boolean` ***isEmpty***()
True if this ModelTransitionSet contains no QNames.

`public void` ***remove***(`QName` *qname*)
Removes the given qname from this QNameSetBuilder.

`public void` ***removeAll***(`QNameSetSpecification` *set*)
Removes all contents of a given QNameSet from this QNameSetBuilder.

`public void` ***removeNamespace***(`String` *uri*)
Removes an entire namespace URI from this QNameSetBuilder.

`public void` ***restrict***(`QNameSetSpecification` *set*)
Restricts the contents of this QNameSetBuilder to be a subset of the given QNameSet.

`public QNameSet` ***toQNameSet***()
Returns a `QNameSet` equivalent to the current state of this QNameSetBuilder.

`public String` ***toString***()
Returns a string representation useful for debugging, subject to change.

`public QNameSet` ***union***(`QNameSetSpecification` *set*)
Returns a new QNameSet that is the union of this one and another.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Methods from interface `com.bea.xml.QNameSetSpecification`

`contains, containsAll, excludedQNamesInIncludedURIs, excludedURIs, includedQNamesInExcludedURIs, includedURIs, intersect, inverse, isAll, isDisjoint, isEmpty, union`

Constructor Detail

QNameSetBuilder

`public QNameSetBuilder()`

Constructs an empty QNameSetBuilder.

QNameSetBuilder

```
public QNameSetBuilder(QNameSetSpecification set)
```

Constructs a QNameSetBuilder whose initial contents are given by another QNameSetSpecification.

QNameSetBuilder

```
public QNameSetBuilder(String str,
                       String targetURI)
```

Constructs a QNameSetBuilder whose initial contents are given as a list of namespace URIs, using the same format used by wildcards in XSD files.

QNameSetBuilder

```
public QNameSetBuilder(Set excludedURIs,
                       Set includedURIs,
                       Set excludedQNamesInIncludedURIs,
                       Set includedQNamesInExcludedURIs)
```

Constructs a QNameSetBuilder whose inital contents are given by the four sets. Exactly one of either excludedURIs or includedURIs must be non−null.

Method Detail

add(QName) Method

```
public void add(QName qname)
```

Adds a single QName to this QNameSetBuilder.

---

addAll(QNameSetSpecification) Method

```
public void addAll(QNameSetSpecification set)
```

Adds the contents of another QNameSet to this QNameSetBuilder.

---

addNamespace(String) Method

```
public void addNamespace(String uri)
```

Adds an entire namespace URI of QNames to this QNameSetBuilder. The empty string is used to signifiy the (local) no−namespace.

clear() Method

`public void clear()`

Clears this QNameSetBuilder

contains(QName) Method

`public boolean contains(QName` *`name`*`)`

True if this ModelTransitionSet contains the given qname.

containsAll(QNameSetSpecification) Method

`public boolean containsAll(QNameSetSpecification` *`set`*`)`

True if the parameter is a subset of this set.

excludedQNamesInIncludedURIs() Method

`public Set excludedQNamesInIncludedURIs()`

Description copied from `QNameSetSpecification.excludedQNamesInIncludedURIs()`
The finite set of QNames that are excluded from the set within namespaces that are otherwise included. Should only contain QNames within namespace that are within the set includedURIs() (or any URI, if includedURIs() is null, which means that all URIs are almost completely included).

Never null.

The same set as inverse().includedQNames().

excludedURIs() Method

`public Set excludedURIs()`

Description copied from `QNameSetSpecification.excludedURIs()`
The finite set of namespace URIs that are almost completely excluded from the set (that is, each namespace URI that included in the set with with a finite number of QName exceptions). Null if the set of namespaceURIs that are almost completely included is infinite.

Null (meaning almost all URIs excluded) if includedURIs() is non−null; non−null otherwise.

The same set as inverse().includedURIs().

includedQNamesInExcludedURIs() Method

`public Set includedQNamesInExcludedURIs()`

Description copied from `QNameSetSpecification.includedQNamesInExcludedURIs()`
The finite set of QNames that are included in the set within namespaces that are otherwise excluded. Should only contain QNames within namespace that are within the set excludedURIs() (or any URI, if excludedURIs() is null, which means that all URIs are almost completely excluded).

Never null.

The same as inverse().excludedQNames().

---

includedURIs() Method

`public Set includedURIs()`

Description copied from `QNameSetSpecification.includedURIs()`
The finite set of namespace URIs that are almost completely included in the set (that is, each namespace URI that included in the set with with a finite number of QName exceptions). Null if the set of namespaceURIs that are almost completely included is infinite.

Null (meaning almost all URIs included) if excludedURIs() is non−null; non−null otherwise.

The same as inverse.excludedURIs().

---

intersect(QNameSetSpecification) Method

`public QNameSet intersect(QNameSetSpecification set)`

Returns a new QNameSet that is the intersection of this one and another.

---

inverse() Method

`public QNameSet inverse()`

Returns a new QNameSet that is the inverse of this one.

---

invert() Method

`public void invert()`

Inverts this QNameSetBuilder.

---

isAll() Method

`public boolean isAll()`

True if this ModelTransitionSet contains all QNames.

---

isDisjoint(QNameSetSpecification) Method

`public boolean isDisjoint(QNameSetSpecification` *`set`*`)`

True if the given set is disjoint from this one.

---

isEmpty() Method

`public boolean isEmpty()`

True if this ModelTransitionSet contains no QNames.

---

remove(QName) Method

`public void remove(QName` *`qname`*`)`

Removes the given qname from this QNameSetBuilder.

---

removeAll(QNameSetSpecification) Method

`public void removeAll(QNameSetSpecification` *`set`*`)`

Removes all contents of a given QNameSet from this QNameSetBuilder.

---

removeNamespace(String) Method

`public void removeNamespace(String` *`uri`*`)`

Removes an entire namespace URI from this QNameSetBuilder.

---

restrict(QNameSetSpecification) Method

`public void restrict(QNameSetSpecification` *`set`*`)`

Restricts the contents of this QNameSetBuilder to be a subset of the given QNameSet. In other words, computes an intersection.

---

toQNameSet() Method

`public QNameSet toQNameSet()`

Returns a `QNameSet` equivalent to the current state of this QNameSetBuilder.

---

toString() Method

`public String toString()`

Returns a string representation useful for debugging, subject to change.

Overrides

`Object.toString()`

---

union(QNameSetSpecification) Method

`public QNameSet union(QNameSetSpecification` *`set`***`)`**

Returns a new QNameSet that is the union of this one and another.

QNameSetSpecification Interface

public interface ***QNameSetSpecification***

Represents a lattice of finite and infinite sets of QNames.

Related Topics

`QNameSet`

All Known Implementing Classes

`QNameSetBuilder`, `QNameSet`

Method Summary

`public boolean` ***contains****(QName name)*

True if the set contains the given QName.

`public boolean` ***containsAll****(QNameSetSpecification set)*

True if the parameter is a subset of this set.

`public Set` ***excludedQNamesInIncludedURIs****()*

The finite set of QNames that are excluded from the set within namespaces that are otherwise included.

`public Set` ***excludedURIs****()*

The finite set of namespace URIs that are almost completely excluded from the set (that is, each namespace URI that included in the set with with a finite number of QName exceptions).

`public Set` ***includedQNamesInExcludedURIs****()*

The finite set of QNames that are included in the set within namespaces that are otherwise excluded.

`public Set` ***includedURIs****()*

The finite set of namespace URIs that are almost completely included in the set (that is, each namespace URI that included in the set with with a finite number of QName exceptions).

`public QNameSet` ***intersect****(QNameSetSpecification set)*

Returns the intersection with another QNameSet.

`public QNameSet` ***inverse****()*

Return the inverse of this QNameSet.

public boolean ***isAll()***

True if the set is the set of all QNames.

public boolean ***isDisjoint****(QNameSetSpecification set)*

True if is disjoint from the specified set.

public boolean ***isEmpty()***

True if the set is empty.

public QNameSet ***union****(QNameSetSpecification set)*

Returns the union with another QNameSet.

Method Detail

contains(QName) Method

public boolean contains(QName *name*)

True if the set contains the given QName. Roughly equivalent to: (includedURIs() == null ? excludedURIs().contains(namespace) : includedURIs().contains(namespace)) ? !excludedQNamesInIncludedURIs().contains(name) : includedQNamesInExcludedURIs().contains(name)

---

containsAll(QNameSetSpecification) Method

public boolean containsAll(QNameSetSpecification *set*)

True if the parameter is a subset of this set.

---

excludedQNamesInIncludedURIs() Method

public Set excludedQNamesInIncludedURIs()

The finite set of QNames that are excluded from the set within namespaces that are otherwise included. Should only contain QNames within namespace that are within the set includedURIs() (or any URI, if includedURIs() is null, which means that all URIs are almost completely included).

Never null.

The same set as inverse().includedQNames().

---

excludedURIs() Method

public Set excludedURIs()

The finite set of namespace URIs that are almost completely excluded from the set (that is, each namespace URI that included in the set with with a finite number of QName exceptions). Null if the set of namespaceURIs that are almost completely included is infinite.

Null (meaning almost all URIs excluded) if includedURIs() is non−null; non−null otherwise.

The same set as inverse().includedURIs().

---

includedQNamesInExcludedURIs() Method

```
public Set includedQNamesInExcludedURIs()
```

The finite set of QNames that are included in the set within namespaces that are otherwise excluded. Should only contain QNames within namespace that are within the set excludedURIs() (or any URI, if excludedURIs() is null, which means that all URIs are almost completely excluded).

Never null.

The same as inverse().excludedQNames().

---

includedURIs() Method

```
public Set includedURIs()
```

The finite set of namespace URIs that are almost completely included in the set (that is, each namespace URI that included in the set with with a finite number of QName exceptions). Null if the set of namespaceURIs that are almost completely included is infinite.

Null (meaning almost all URIs included) if excludedURIs() is non−null; non−null otherwise.

The same as inverse.excludedURIs().

---

intersect(QNameSetSpecification) Method

```
public QNameSet intersect(QNameSetSpecification set)
```

Returns the intersection with another QNameSet.

---

inverse() Method

```
public QNameSet inverse()
```

Return the inverse of this QNameSet. That is the QNameSet which contains all the QNames not contained in this set. In other words for which set.contains(name) != set.inverse().contains(name) for all names.

---

isAll() Method

```
public boolean isAll()
```

True if the set is the set of all QNames.

---

isDisjoint(QNameSetSpecification) Method

```
public boolean isDisjoint(QNameSetSpecification set)
```

True if is disjoint from the specified set.

---

isEmpty() Method

```
public boolean isEmpty()
```

True if the set is empty.

---

union(QNameSetSpecification) Method

```
public QNameSet union(QNameSetSpecification set)
```

Returns the union with another QNameSet.

com.bea.xml

SchemaAttributeGroup Interface

public interface ***SchemaAttributeGroup***

extends `SchemaComponent`

Represents an attribute group.

An attribute group is a syntactic construct, not a part of the logical model. For example, types declared within an attribute group become local to the type that uses the group − they're not local to the attribute group itself. Therefore in the logical model of a schema type system, an attribute group doesn't represent anything. Its contents are merged into the logical model at parse time.

Related Topics

`SchemaTypeLoader.findAttributeGroup(QName)`

All Superinterfaces

`SchemaComponent`

Field Summary

Fields from interface com.bea.xml.`SchemaComponent`

`ATTRIBUTE, ATTRIBUTE_GROUP, ELEMENT, IDENTITY_CONSTRAINT, MODEL_GROUP, NOTATION, TYPE`

Method Summary

`public int` ***`getComponentType`***`()`
Returns SchemaComponent.ATTRIBUTE_GROUP.

`public QName` ***`getName`***`()`
The name of the model group.

Methods from interface `com.bea.xml.SchemaComponent`

`getComponentRef, getTypeSystem`

Method Detail

getComponentType() Method

```
public int getComponentType()
```

Returns SchemaComponent.ATTRIBUTE_GROUP.

---

getName() Method

```
public QName getName()
```

The name of the model group.

com.bea.xml

SchemaAttributeModel Interface

public interface ***SchemaAttributeModel***

Represents the attribute structure allowed on a complex type.

Related Topics

`SchemaType.getAttributeModel()`

Field Summary

`public static final int` ***LAX***	Lax wildcard processing.
`public static final int` ***NONE***	See `SchemaAttributeModel.getWildcardProcess()`
`public static final int` ***SKIP***	Skip wildcard processing.
`public static final int` ***STRICT***	Strict wildcard processing.

Method Summary

`public SchemaLocalAttribute` ***getAttribute****(`QName` name)*	Returns the attribute with the given name.
`public SchemaLocalAttribute[]` ***getAttributes****()*	Returns an array containing all the attributes in the model.
`public int` ***getWildcardProcess****()*	Returns the processing code (`SchemaAttributeModel.STRICT`, `SchemaAttributeModel.LAX`, `SchemaAttributeModel.SKIP`).
`public QNameSet` ***getWildcardSet****()*	QNameSet representing the attribute wildcard specification.

Field Detail

LAX

`public static final int LAX`

Lax wildcard processing. See `SchemaAttributeModel.getWildcardProcess()`

NONE

`public static final int NONE`

See `SchemaAttributeModel.getWildcardProcess()`

SKIP

`public static final int SKIP`

Skip wildcard processing. See `SchemaAttributeModel.getWildcardProcess()`

STRICT

`public static final int STRICT`

Strict wildcard processing. See `SchemaAttributeModel.getWildcardProcess()`

Method Detail

getAttribute(QName) Method

`public SchemaLocalAttribute getAttribute(QName` *`name`***`)`**

Returns the attribute with the given name.

---

getAttributes() Method

`public SchemaLocalAttribute[] getAttributes()`

Returns an array containing all the attributes in the model.

---

getWildcardProcess() Method

```
public int getWildcardProcess()
```

Returns the processing code (`SchemaAttributeModel.STRICT`, `SchemaAttributeModel.LAX`, `SchemaAttributeModel.SKIP`). Returns 0 (`SchemaAttributeModel.NONE`) if no wildcard specified.

---

getWildcardSet() Method

```
public QNameSet getWildcardSet()
```

QNameSet representing the attribute wildcard specification.

SchemaComponent Interface

public interface ***SchemaComponent***

Represents a global Schema Component. That is, a type, element, attribute, model group, attribute group, or identity constraint.

Note that not all types, elements, and attributes are global; local types, element, and attributes do not appear in the global lookup table. Also note that other information items such as particles, facets, and so on are not globally indexed, so are not SchemaComponents.

Related Topics

`SchemaType`
`SchemaGlobalElement`
`SchemaGlobalAttribute`
`SchemaAttributeGroup`
`SchemaModelGroup`
`SchemaIdentityConstraint`

All Known Subinterfaces

`SchemaAttributeGroup`, `SchemaGlobalAttribute`, `SchemaGlobalElement`, `SchemaIdentityConstraint`, `SchemaModelGroup`, `SchemaType`

Field Summary

`public static final int` ***ATTRIBUTE***
An attribute definition.

`public static final int` ***ATTRIBUTE_GROUP***
An attribute group definition.

`public static final int` ***ELEMENT***
An element definition.

`public static final int` ***IDENTITY_CONSTRAINT***
An identity constraint definition.

`public static final int` ***MODEL_GROUP***
A model group definition.

`public static final int` ***NOTATION***
A notation definition.

public static final int ***TYPE***	A type definition.

Method Summary

public SchemaComponent.Ref ***getComponentRef()***	Used for on−demand loading of schema components.
public int ***getComponentType()***	Returns the type code for the schema object, either `SchemaComponent.TYPE`, `SchemaComponent.ELEMENT`, `SchemaComponent.ATTRIBUTE`, `SchemaComponent.ATTRIBUTE_GROUP`, `SchemaComponent.MODEL_GROUP`, `SchemaComponent.IDENTITY_CONSTRAINT`, or `SchemaComponent.NOTATION`.
public QName ***getName()***	The name of the schema component
public SchemaTypeSystem ***getTypeSystem()***	Returns the typesystem within which this component definition resides

Field Detail

ATTRIBUTE

public static final int ATTRIBUTE

An attribute definition. See `SchemaComponent.getComponentType()`

ATTRIBUTE_GROUP

public static final int ATTRIBUTE_GROUP

An attribute group definition. See `SchemaComponent.getComponentType()`

ELEMENT

public static final int ELEMENT

An element definition. See `SchemaComponent.getComponentType()`

IDENTITY_CONSTRAINT

`public static final int IDENTITY_CONSTRAINT`

An identity constraint definition. See `SchemaComponent.getComponentType()`

MODEL_GROUP

`public static final int MODEL_GROUP`

A model group definition. See `SchemaComponent.getComponentType()`

NOTATION

`public static final int NOTATION`

A notation definition. See `SchemaComponent.getComponentType()`

TYPE

`public static final int TYPE`

A type definition. See `SchemaComponent.getComponentType()`

Method Detail

getComponentRef() Method

`public SchemaComponent.Ref getComponentRef()`

Used for on−demand loading of schema components.

---

getComponentType() Method

`public int getComponentType()`

Returns the type code for the schema object, either `SchemaComponent.TYPE`, `SchemaComponent.ELEMENT`, `SchemaComponent.ATTRIBUTE`, `SchemaComponent.ATTRIBUTE_GROUP`, `SchemaComponent.MODEL_GROUP`, `SchemaComponent.IDENTITY_CONSTRAINT`, or `SchemaComponent.NOTATION`.

---

getName() Method

```
public QName getName()
```

The name of the schema component

---

getTypeSystem() Method

```
public SchemaTypeSystem getTypeSystem()
```

Returns the typesystem within which this component definition resides

SchemaField Interface

public interface ***SchemaField***

Represents an element or an attribute declaration.

Related Topics

`SchemaType.getContainerField()`
`SchemaLocalElement`
`SchemaLocalAttribute`

All Known Subinterfaces

`SchemaGlobalAttribute`, `SchemaGlobalElement`, `SchemaLocalAttribute`, `SchemaLocalElement`

Method Summary

`public String` ***getDefaultText()***
The default value as plain text.

`public XmlAnySimpleType` ***getDefaultValue()***
The default value as a strongly−typed value.

`public BigInteger` ***getMaxOccurs()***
Returns the maxOccurs value for this particle, or null if it is unbounded.

`public BigInteger` ***getMinOccurs()***
Returns the minOccurs value for this particle.

`public QName` ***getName()***
Returns the form−unqualified−or−qualified name.

`public SchemaType` ***getType()***
Returns the type of this use.

`public boolean` ***isAttribute()***
True if this use is an attribute

`public boolean` ***isDefault()***
True if a default is supplied.

`public boolean` ***`isFixed`***`()`
True if the value is fixed.

`public boolean` ***`isNillable`***`()`
True if nillable; always false for attributes.

Method Detail

getDefaultText() Method

`public String getDefaultText()`

The default value as plain text. See `SchemaField.isDefault()` and `SchemaField.isFixed()`.

---

getDefaultValue() Method

`public XmlAnySimpleType getDefaultValue()`

The default value as a strongly−typed value. See `SchemaField.isDefault()` and `SchemaField.isFixed()`.

---

getMaxOccurs() Method

`public BigInteger getMaxOccurs()`

Returns the maxOccurs value for this particle, or null if it is unbounded. If it is not specified explicitly, this defaults to BigInteger.ONE.

---

getMinOccurs() Method

`public BigInteger getMinOccurs()`

Returns the minOccurs value for this particle. If it is not specified explicitly, this defaults to BigInteger.ONE.

---

getName() Method

`public QName getName()`

Returns the form−unqualified−or−qualified name.

---

getType() Method

`public SchemaType getType()`

Returns the type of this use.

isAttribute() Method

`public boolean isAttribute()`

True if this use is an attribute

isDefault() Method

`public boolean isDefault()`

True if a default is supplied. If `SchemaField.isFixed()`, then isDefault is always true.

isFixed() Method

`public boolean isFixed()`

True if the value is fixed.

isNillable() Method

`public boolean isNillable()`

True if nillable; always false for attributes.

com.bea.xml

SchemaGlobalAttribute Interface

public interface ***SchemaGlobalAttribute***

extends `SchemaComponent`, `SchemaLocalAttribute`

Represents a global attribute definition

Related Topics

`SchemaTypeLoader.findAttribute(QName)`

All Superinterfaces

`SchemaComponent`, `SchemaField`, `SchemaLocalAttribute`

Field Summary

Fields from interface com.bea.xml.`SchemaComponent`

`ATTRIBUTE, ATTRIBUTE_GROUP, ELEMENT, IDENTITY_CONSTRAINT, MODEL_GROUP, NOTATION, TYPE`

Fields from interface com.bea.xml.`SchemaLocalAttribute`

`OPTIONAL, PROHIBITED, REQUIRED`

Method Summary

`public SchemaGlobalAttribute.Ref` ***getRef()***

Retruns a SchemaGlobalElement.Ref pointing to this element itself.

`public String` ***getSourceName()***

The name of the source .xsd file within which this attribute was defined

Methods from interface `com.bea.xml.SchemaComponent`

`getComponentRef, getComponentType, getName, getTypeSystem`

Methods from interface `com.bea.xml.SchemaField`

`getDefaultText, getDefaultValue, getMaxOccurs, getMinOccurs, getName, getType, isAttribute, isDefault, isFixed, isNillable`

Methods from interface `com.bea.xml.SchemaLocalAttribute`

`getUse`

Method Detail

getRef() Method

`public SchemaGlobalAttribute.Ref getRef()`

Retruns a SchemaGlobalElement.Ref pointing to this element itself.

---

getSourceName() Method

`public String getSourceName()`

The name of the source .xsd file within which this attribute was defined

com.bea.xml

SchemaGlobalElement Interface

public interface ***SchemaGlobalElement***

extends `SchemaComponent`, `SchemaLocalElement`

Represents a global element definition.

Related Topics

`SchemaTypeLoader.findElement(QName)`

All Superinterfaces

`SchemaComponent`, `SchemaField`, `SchemaLocalElement`

Field Summary

Fields from interface com.bea.xml.`SchemaComponent`

`ATTRIBUTE, ATTRIBUTE_GROUP, ELEMENT, IDENTITY_CONSTRAINT, MODEL_GROUP, NOTATION, TYPE`

Method Summary

`public boolean` ***finalExtension()***
True if using this element as the head of a substitution group for a substitution via type extension is prohibited.

`public boolean` ***finalRestriction()***
True if using this element as the head of a substitution group for a substitution via type restriction is prohibited.

`public SchemaGlobalElement.Ref` ***getRef()***
Retruns a SchemaGlobalElement.Ref pointing to this element itself.

`public String` ***getSourceName()***
The name of the source .xsd file within which this attribute was defined

`public SchemaGlobalElement` ***substitutionGroup()***

The element that is the head of this element's substitution group, or `null` if this element is not a member of a substitution group.

`public QName[]`
substitutionGroupMembers()

Set of QNames for elements that are the members of the substitution group for which this element is the head, not including this element.

Methods from interface `com.bea.xml.SchemaComponent`

`getComponentRef, getComponentType, getName, getTypeSystem`

Methods from interface `com.bea.xml.SchemaField`

`getDefaultText, getDefaultValue, getMaxOccurs, getMinOccurs, getName, getType, isAttribute, isDefault, isFixed, isNillable`

Methods from interface `com.bea.xml.SchemaLocalElement`

`blockExtension, blockRestriction, blockSubstitution, getIdentityConstraints, isAbstract`

Method Detail

finalExtension() Method

`public boolean finalExtension()`

True if using this element as the head of a substitution group for a substitution via type extension is prohibited. If both finalExtension and finalRestriction are true, this element cannot be head of a substitution group. Sensible only for global elements.

---

finalRestriction() Method

`public boolean finalRestriction()`

True if using this element as the head of a substitution group for a substitution via type restriction is prohibited. If both finalExtension and finalRestriction are true, this element cannot be head of a substitution group. Sensible only for global elements.

---

getRef() Method

`public SchemaGlobalElement.Ref getRef()`

Retruns a SchemaGlobalElement.Ref pointing to this element itself.

---

getSourceName() Method

```
public String getSourceName()
```

The name of the source .xsd file within which this attribute was defined

---

substitutionGroup() Method

```
public SchemaGlobalElement substitutionGroup()
```

The element that is the head of this element's substitution group, or `null` if this element is not a member of a substitution group.

---

substitutionGroupMembers() Method

```
public QName[] substitutionGroupMembers()
```

Set of QNames for elements that are the members of the substitution group for which this element is the head, not including this element.

com.bea.xml

SchemaIdentityConstraint Interface

public interface ***SchemaIdentityConstraint***

extends `SchemaComponent`

Represents an identity constraint definition.

All Superinterfaces

`SchemaComponent`

Field Summary

`public static final int` ***CC_KEY***
A xs:key constraint.

`public static final int` ***CC_KEYREF***
A xs:keyRef constraint.

`public static final int` ***CC_UNIQUE***
A xs:unique constraint.

Fields from interface com.bea.xml.`SchemaComponent`

`ATTRIBUTE, ATTRIBUTE_GROUP, ELEMENT, IDENTITY_CONSTRAINT, MODEL_GROUP, NOTATION, TYPE`

Method Summary

`public int` ***getConstraintCategory()***
Return the constraint category.

`public Object` ***getFieldPath****(int index)*
Return a compiled xpath object for the field.

`public String[]` ***getFields()***
Return (a copy of) the xpaths for all the fields.

`public Map` ***getNSMap()***

Return a read−only copy of the namespace map.

`public SchemaIdentityConstraint` ***getReferencedKey()***

Returns the key that a key ref refers to.

`public String` ***getSelector()***

Return the selector xpath as a string.

`public Object` ***getSelectorPath()***

Return a compiled xpath object for the selector.

Methods from interface `com.bea.xml.SchemaComponent`

`getComponentRef, getComponentType, getName, getTypeSystem`

Field Detail

CC_KEY

`public static final int CC_KEY`

A xs:key constraint. See `SchemaIdentityConstraint.getConstraintCategory()`.

CC_KEYREF

`public static final int CC_KEYREF`

A xs:keyRef constraint. See `SchemaIdentityConstraint.getConstraintCategory()`.

CC_UNIQUE

`public static final int CC_UNIQUE`

A xs:unique constraint. See `SchemaIdentityConstraint.getConstraintCategory()`.

Method Detail

getConstraintCategory() Method

`public int getConstraintCategory()`

Return the constraint category. Either `SchemaIdentityConstraint.CC_KEY`, `SchemaIdentityConstraint.CC_KEYREF`, or `SchemaIdentityConstraint.CC_UNIQUE`.

---

getFieldPath(int) Method

`public Object getFieldPath(int index)`

Return a compiled xpath object for the field.

---

getFields() Method

`public String[] getFields()`

Return (a copy of) the xpaths for all the fields.

---

getNSMap() Method

`public Map getNSMap()`

Return a read−only copy of the namespace map. This is the set of prefix to URI mappings that were in scope in the schema at the point at which this constraint was declared

---

getReferencedKey() Method

`public SchemaIdentityConstraint getReferencedKey()`

Returns the key that a key ref refers to. Only valid for keyrefs.

---

getSelector() Method

`public String getSelector()`

Return the selector xpath as a string.

---

getSelectorPath() Method

`public Object getSelectorPath()`

Return a compiled xpath object for the selector.

com.bea.xml

SchemaLocalAttribute Interface

public interface ***SchemaLocalAttribute***

extends `SchemaField`

Represents a local or global attribute definition.

All Superinterfaces

`SchemaField`

All Known Subinterfaces

`SchemaGlobalAttribute`

Field Summary

`public static final int` ***OPTIONAL***
An optional attribute.

`public static final int` ***PROHIBITED***
A prohibited attribute.

`public static final int` ***REQUIRED***
A required attribute.

Method Summary

`public int` ***getUse()***
Returns `SchemaLocalAttribute.PROHIBITED`, `SchemaLocalAttribute.OPTIONAL`, or `SchemaLocalAttribute.REQUIRED`.

Methods from interface `com.bea.xml.SchemaField`

`getDefaultText, getDefaultValue, getMaxOccurs, getMinOccurs, getName, getType, isAttribute, isDefault, isFixed, isNillable`

Field Detail

OPTIONAL

`public static final int OPTIONAL`

An optional attribute. See `SchemaLocalAttribute.getUse()`.

PROHIBITED

`public static final int PROHIBITED`

A prohibited attribute. See `SchemaLocalAttribute.getUse()`.

REQUIRED

`public static final int REQUIRED`

A required attribute. See `SchemaLocalAttribute.getUse()`.

Method Detail

getUse() Method

`public int getUse()`

Returns `SchemaLocalAttribute.PROHIBITED`, `SchemaLocalAttribute.OPTIONAL`, or `SchemaLocalAttribute.REQUIRED`. (Actually, never returns PROHIBITED because the schema specificaion states that a prohibited attribute is equivalent to no attribute at all, so a prohibited attribute will never be present in the compiled model.)

com.bea.xml

SchemaLocalElement Interface

public interface ***SchemaLocalElement***

extends `SchemaField`

Represents a local or global element definition.

All Superinterfaces

`SchemaField`

All Known Subinterfaces

`SchemaGlobalElement`

Method Summary

`public boolean` ***blockExtension()***
True if extension is blocked.

`public boolean` ***blockRestriction()***
True if restriction is blocked.

`public boolean` ***blockSubstitution()***
True if element substitution is blocked.

`public SchemaIdentityConstraint[]` ***getIdentityConstraints()***
Returns all the Key, KeyRef, and Unique constraints on this element.

`public boolean` ***isAbstract()***
True if this element is prohibited in content.

Methods from interface `com.bea.xml.SchemaField`

`getDefaultText, getDefaultValue, getMaxOccurs, getMinOccurs, getName, getType, isAttribute, isDefault, isFixed, isNillable`

Method Detail

blockExtension() Method

```
public boolean blockExtension()
```

True if extension is blocked.

---

blockRestriction() Method

```
public boolean blockRestriction()
```

True if restriction is blocked.

---

blockSubstitution() Method

```
public boolean blockSubstitution()
```

True if element substitution is blocked.

---

getIdentityConstraints() Method

```
public SchemaIdentityConstraint[] getIdentityConstraints()
```

Returns all the Key, KeyRef, and Unique constraints on this element.

---

isAbstract() Method

```
public boolean isAbstract()
```

True if this element is prohibited in content. Only sensible if this is the head of a substitution group; then only substitution group members can appear.

Although local elements cannot be abstract, if an element use is a ref to a global element, then the flag from the global element is copied in to the local element where the use occurs.

com.bea.xml

SchemaModelGroup Interface

public interface ***SchemaModelGroup***

extends `SchemaComponent`

Represents a model group.

A model group is a syntactic construct, not a part of the logical model. For example, types declared within a model group become local to the type that uses the group − they're not local to the model group itself. Therefore in the logical model of a schema type system, a model group doesn't represent anything. Its contents are merged into the logical model at parse time.

All Superinterfaces

`SchemaComponent`

Field Summary

Fields from interface com.bea.xml.`SchemaComponent`

`ATTRIBUTE, ATTRIBUTE_GROUP, ELEMENT, IDENTITY_CONSTRAINT, MODEL_GROUP, NOTATION, TYPE`

Method Summary

`public int` ***`getComponentType`***`()`
Returns SchemaComponent.GROUP.

`public QName` ***`getName`***`()`
The name of the model group.

Methods from interface `com.bea.xml.SchemaComponent`

`getComponentRef, getTypeSystem`

Method Detail

getComponentType() Method

```
public int getComponentType()
```

Returns SchemaComponent.GROUP.

---

getName() Method

```
public QName getName()
```

The name of the model group.

SchemaParticle Interface

public interface ***SchemaParticle***

Represents a Schema particle definition.

The content model of a complex type is a tree of particles. Each particle is either an `SchemaParticle.ALL`, `SchemaParticle.CHOICE`, `SchemaParticle.SEQUENCE`, `SchemaParticle.ELEMENT`, or `SchemaParticle.WILDCARD`. All, choice and sequence particles are groups that can have child particles; elements and wildcards are always leaves of the particle tree.

The tree of particles available on a schema type is minimized, that is, it already has removed "pointless" particles such as empty sequences, nonrepeating sequences with only one item, and so on. (Pointless particles are defined precisely in the XML Schema specification.)

Related Topics

`SchemaType.getContentModel()`

Field Summary

`public static final int` ***ALL***
An xs:all group.

`public static final int` ***CHOICE***
A xs:choice group.

`public static final int` ***ELEMENT***
An xs:element particle.

`public static final int` ***LAX***
Lax wildcard processing.

`public static final int` ***SEQUENCE***
A xs:sequence group.

`public static final int` ***SKIP***
Skip wildcard processing.

`public static final int` ***STRICT***
Strict wildcard processing.

`public static final int` ***WILDCARD***

An xs:any particle, also known as an element wildcard.

Method Summary

`public QNameSet` ***acceptedStartNames()***
Returns the QNameSet of element names that can be accepted at the beginning of this particle.

`public boolean` ***canStartWithElement(QName name)***
True if this particle can start with the given element (taking into account the structure of all child particles of course).

`public int` ***countOfParticleChild()***
The number of children.

`public String` ***getDefaultText()***
For elements only: returns the default (or fixed) text value

`public XmlAnySimpleType` ***getDefaultValue()***
For elements only: returns the default (or fixed) strongly−typed value

`public int` ***getIntMaxOccurs()***
Returns the maxOccurs value, pegged to a 32−bit int for convenience of a validating state machine that doesn't count higher than MAX_INT anyway.

`public int` ***getIntMinOccurs()***
Returns the minOccurs value, pegged to a 32−bit int for convenience of a validating state machine that doesn't count higher than MAX_INT anyway.

`public BigInteger` ***getMaxOccurs()***
Returns the maxOccurs value for this particle, or null if it is unbounded.

`public BigInteger` ***getMinOccurs()***
Returns the minOccurs value for this particle.

`public QName` ***getName()***
For elements only: the QName for the element use.

`public SchemaParticle` ***getParticleChild(int i)***
Another way to access the particle children.

`public SchemaParticle[]` ***getParticleChildren()***

Applies to sequence, choice, and all particles only: returns an array of all the particle children in order.

`public int` ***getParticleType()***

Returns the particle type (`SchemaParticle.ALL`, `SchemaParticle.CHOICE`, `SchemaParticle.SEQUENCE`, `SchemaParticle.ELEMENT`, or `SchemaParticle.WILDCARD`).

`public SchemaType` ***getType()***

For elements only: returns the type of the element.

`public int` ***getWildcardProcess()***

For wildcards, returns the processing code (`SchemaParticle.STRICT`, `SchemaParticle.LAX`, `SchemaParticle.SKIP`).

`public QNameSet` ***getWildcardSet()***

For wildcards, returns a QNameSet representing the wildcard.

`public boolean` ***isDefault()***

For elements only: True if has default.

`public boolean` ***isFixed()***

For elements only: true if is fixed value.

`public boolean` ***isNillable()***

For elements only: true if nillable.

`public boolean` ***isSingleton()***

One if minOccurs == maxOccurs == 1.

`public boolean` ***isSkippable()***

True if this particle can be skipped (taking into account both the minOcurs as well as the structure of all the child particles)

Field Detail

ALL

`public static final int ALL`

An xs:all group. See `SchemaParticle.getParticleType()`.

CHOICE

public static final int CHOICE

A xs:choice group. See `SchemaParticle.getParticleType()`.

ELEMENT

public static final int ELEMENT

An xs:element particle. This code means the particle can be coerced to `SchemaLocalElement`. See `SchemaParticle.getParticleType()`.

LAX

public static final int LAX

Lax wildcard processing. See `SchemaParticle.getWildcardProcess()`

SEQUENCE

public static final int SEQUENCE

A xs:sequence group. See `SchemaParticle.getParticleType()`.

SKIP

public static final int SKIP

Skip wildcard processing. See `SchemaParticle.getWildcardProcess()`

STRICT

public static final int STRICT

Strict wildcard processing. See `SchemaParticle.getWildcardProcess()`

WILDCARD

public static final int WILDCARD

An xs:any particle, also known as an element wildcard. See `SchemaParticle.getParticleType()`.

Method Detail

acceptedStartNames() Method

```
public QNameSet acceptedStartNames()
```

Returns the QNameSet of element names that can be accepted at the beginning of this particle.

---

canStartWithElement(QName) Method

```
public boolean canStartWithElement(QName name)
```

True if this particle can start with the given element (taking into account the structure of all child particles of course).

---

countOfParticleChild() Method

```
public int countOfParticleChild()
```

The number of children.

---

getDefaultText() Method

```
public String getDefaultText()
```

For elements only: returns the default (or fixed) text value

---

getDefaultValue() Method

```
public XmlAnySimpleType getDefaultValue()
```

For elements only: returns the default (or fixed) strongly−typed value

---

getIntMaxOccurs() Method

```
public int getIntMaxOccurs()
```

Returns the maxOccurs value, pegged to a 32−bit int for convenience of a validating state machine that doesn't count higher than MAX_INT anyway. Unbounded is given as MAX_INT.

---

getIntMinOccurs() Method

```
public int getIntMinOccurs()
```

Returns the minOccurs value, pegged to a 32−bit int for convenience of a validating state machine that doesn't count higher than MAX_INT anyway.

---

getMaxOccurs() Method

```
public BigInteger getMaxOccurs()
```

Returns the maxOccurs value for this particle, or null if it is unbounded. If it's not specified explicitly, this returns BigInteger.ONE.

---

getMinOccurs() Method

```
public BigInteger getMinOccurs()
```

Returns the minOccurs value for this particle. If it's not specified explicitly, this returns BigInteger.ONE.

---

getName() Method

```
public QName getName()
```

For elements only: the QName for the element use. May be unqualified version of referenced element's name.

---

getParticleChild(int) Method

```
public SchemaParticle getParticleChild(int i)
```

Another way to access the particle children.

---

getParticleChildren() Method

```
public SchemaParticle[] getParticleChildren()
```

Applies to sequence, choice, and all particles only: returns an array of all the particle children in order.

---

getParticleType() Method

```
public int getParticleType()
```

Returns the particle type (`SchemaParticle.ALL`, `SchemaParticle.CHOICE`, `SchemaParticle.SEQUENCE`, `SchemaParticle.ELEMENT`, or `SchemaParticle.WILDCARD`).

getType() Method

```
public SchemaType getType()
```

For elements only: returns the type of the element.

getWildcardProcess() Method

```
public int getWildcardProcess()
```

For wildcards, returns the processing code (`SchemaParticle.STRICT`, `SchemaParticle.LAX`, `SchemaParticle.SKIP`).

getWildcardSet() Method

```
public QNameSet getWildcardSet()
```

For wildcards, returns a QNameSet representing the wildcard.

isDefault() Method

```
public boolean isDefault()
```

For elements only: True if has default. If isFixed, then isDefault is always true.

isFixed() Method

```
public boolean isFixed()
```

For elements only: true if is fixed value.

isNillable() Method

```
public boolean isNillable()
```

For elements only: true if nillable.

isSingleton() Method

```
public boolean isSingleton()
```

One if minOccurs == maxOccurs == 1.

isSkippable() Method

```
public boolean isSkippable()
```

True if this particle can be skipped (taking into account both the minOcurs as well as the structure of all the child particles)

SchemaProperty Interface

public interface ***SchemaProperty***

Represents a summary of similar SchemaFields in a complex type.

In a schema type, every element with the same name must have the same type. Therefore, all together, elements with the same name form a coherent collection of similar elements. Similarly, attributes can only be defined once, so each attribute obviously is a coherent group on its own.

A SchemaProperty represents a summary of the the elements with a given name or the attribute with a given name. It represents the summary cardinality of the fields, the summary default and fixed values, and so on. When inferring information about an element or attribute, it is typically easier to consult then SchemaProperty than to hunt for the exact SchemaField in the particle tree or attribute model.

Related Topics

```
SchemaType.getProperties()
SchemaType.getAttributeProperties()
SchemaType.getElementProperties()
SchemaType.getAttributeProperty(QName)
SchemaType.getElementProperty(QName)
```

Field Summary

`public static final int` ***CONSISTENTLY***
Applies to all elements for this property.

`public static final int` ***JAVA_BIG_DECIMAL***
A `BigDecimal`.

`public static final int` ***JAVA_BIG_INTEGER***
A `BigInteger`.

`public static final int` ***JAVA_BOOLEAN***
A Java boolean.

`public static final int` ***JAVA_BYTE***
A Java byte.

`public static final int` ***JAVA_BYTE_ARRAY***
A byte[].

`public static`

final int ***JAVA_CALENDAR***
A `Calendar`.

public static final int ***JAVA_DATE***
A `Date`.

public static final int ***JAVA_DOUBLE***
A Java double.

public static final int ***JAVA_ENUM***
A generated `StringEnumAbstractBase` subclass.

public static final int ***JAVA_FIRST_PRIMITIVE***
Java primitive type codes (for non−nullable Java types) are between JAVA_FIRST_PRIMITIVE and JAVA_LAST_PRIMITIVE, inclusive.

public static final int ***JAVA_FLOAT***
A Java float.

public static final int ***JAVA_GDATE***
A `GDate`.

public static final int ***JAVA_GDURATION***
A `GDuration`.

public static final int ***JAVA_INT***
A Java int.

public static final int ***JAVA_LAST_PRIMITIVE***
Java primitive type codes (for non−nullable Java types) are between JAVA_FIRST_PRIMITIVE and JAVA_LAST_PRIMITIVE, inclusive.

public static final int ***JAVA_LIST***
A `List`.

public static final int ***JAVA_LONG***
A Java long.

public static final int ***JAVA_OBJECT***
A `Object`, used for some simple type unions.

public static final int ***JAVA_QNAME***
A `QName`.

public static final int ***JAVA_SHORT***
A Java short.

public static

`final int` ***JAVA_STRING***
A `String`.

`public static final int` ***NEVER***
Applies to no elements for this property.

`public static final int` ***VARIABLE***
Applies to some, but not other elements for this property.

`public static final int` ***XML_OBJECT***
An XML Bean type that inherits from `XmlObject`.

Method Summary

`public QName[]` ***acceptedNames()***
For element properties the set of names that are accepted for this property if this element is the head of a substitution group.

`public boolean` ***extendsJavaArray()***
True if there is a Java getter that returns an array.

`public boolean` ***extendsJavaOption()***
True if there is an Java isSet method that tests for presence.

`public boolean` ***extendsJavaSingleton()***
True if there is a Java getter that returns a singleton.

`public SchemaType` ***getContainerType()***
The type within which this property appears

`public String` ***getDefaultText()***
Returns the default or fixed value, if it is consistent.

`public XmlAnySimpleType` ***getDefaultValue()***
Returns the default or fixed value as a strongly−typed value, if it is consistent.

`public String` ***getJavaPropertyName()***
The Java name for this property.

`public QNameSet` ***getJavaSetterDelimiter()***
Returns the set of element names which should appear strictly after all occurences of the elements described by this property.

`public int` ***getJavaTypeCode()***

Returns the natural Java type for this property.

`public BigInteger` ***getMaxOccurs()***
Returns a summarized minimum occurrance number.

`public BigInteger` ***getMinOccurs()***
Returns a summarized minimum occurrance number.

`public QName` ***getName()***
The name of this element or attribute

`public SchemaType` ***getType()***
The schema type for the property.

`public int` ***hasDefault()***
Returns `SchemaProperty.NEVER`, `SchemaProperty.VARIABLE`, or `SchemaProperty.CONSISTENTLY` defaulted, depending on the defaults present in the elements in this property.

`public int` ***hasFixed()***
Returns `SchemaProperty.NEVER`, `SchemaProperty.VARIABLE`, or `SchemaProperty.CONSISTENTLY` fixed, depending on the fixed constraints present in the elements in this property.

`public int` ***hasNillable()***
Returns `SchemaProperty.NEVER`, `SchemaProperty.VARIABLE`, or `SchemaProperty.CONSISTENTLY` nillable, depending on the nillability of the elements in this property.

`public boolean` ***isAttribute()***
True for attributes.

`public boolean` ***isReadOnly()***
True for read−only properties.

`public SchemaType` ***javaBasedOnType()***
The schema type returned from the Java getter for this property.

Field Detail

CONSISTENTLY

`public static final int CONSISTENTLY`

Applies to all elements for this property. See `SchemaProperty.hasNillable()`, `SchemaProperty.hasDefault()`, `SchemaProperty.hasFixed()`

JAVA_BIG_DECIMAL

public static final int JAVA_BIG_DECIMAL

A `BigDecimal`. See `SchemaProperty.getJavaTypeCode()`.

JAVA_BIG_INTEGER

public static final int JAVA_BIG_INTEGER

A `BigInteger`. See `SchemaProperty.getJavaTypeCode()`.

JAVA_BOOLEAN

public static final int JAVA_BOOLEAN

A Java boolean. See `SchemaProperty.getJavaTypeCode()`.

JAVA_BYTE

public static final int JAVA_BYTE

A Java byte. See `SchemaProperty.getJavaTypeCode()`.

JAVA_BYTE_ARRAY

public static final int JAVA_BYTE_ARRAY

A byte[]. See `SchemaProperty.getJavaTypeCode()`.

JAVA_CALENDAR

public static final int JAVA_CALENDAR

A `Calendar`. See `SchemaProperty.getJavaTypeCode()`.

JAVA_DATE

public static final int JAVA_DATE

A `Date`. See `SchemaProperty.getJavaTypeCode()`.

JAVA_DOUBLE

public static final int JAVA_DOUBLE

A Java double. See `SchemaProperty.getJavaTypeCode()`.

JAVA_ENUM

public static final int JAVA_ENUM

A generated `StringEnumAbstractBase` subclass. See `SchemaProperty.getJavaTypeCode()`.

JAVA_FIRST_PRIMITIVE

public static final int JAVA_FIRST_PRIMITIVE

Java primitive type codes (for non−nullable Java types) are between JAVA_FIRST_PRIMITIVE and JAVA_LAST_PRIMITIVE, inclusive.

JAVA_FLOAT

public static final int JAVA_FLOAT

A Java float. See `SchemaProperty.getJavaTypeCode()`.

JAVA_GDATE

public static final int JAVA_GDATE

A `GDate`. See `SchemaProperty.getJavaTypeCode()`.

JAVA_GDURATION

public static final int JAVA_GDURATION

A `GDuration`. See `SchemaProperty.getJavaTypeCode()`.

JAVA_INT

public static final int JAVA_INT

A Java int. See `SchemaProperty.getJavaTypeCode()`.

JAVA_LAST_PRIMITIVE

public static final int JAVA_LAST_PRIMITIVE

Java primitive type codes (for non−nullable Java types) are between JAVA_FIRST_PRIMITIVE and JAVA_LAST_PRIMITIVE, inclusive.

JAVA_LIST

public static final int JAVA_LIST

A `List`. See `SchemaProperty.getJavaTypeCode()`.

JAVA_LONG

public static final int JAVA_LONG

A Java long. See `SchemaProperty.getJavaTypeCode()`.

JAVA_OBJECT

public static final int JAVA_OBJECT

A `Object`, used for some simple type unions. See `SchemaProperty.getJavaTypeCode()`.

JAVA_QNAME

public static final int JAVA_QNAME

A `QName`. See `SchemaProperty.getJavaTypeCode()`.

JAVA_SHORT

public static final int JAVA_SHORT

A Java short. See `SchemaProperty.getJavaTypeCode()`.

JAVA_STRING

public static final int JAVA_STRING

A `String`. See `SchemaProperty.getJavaTypeCode()`.

NEVER

public static final int NEVER

Applies to no elements for this property. See `SchemaProperty.hasNillable()`, `SchemaProperty.hasDefault()`, `SchemaProperty.hasFixed()`

VARIABLE

public static final int VARIABLE

Applies to some, but not other elements for this property. See `SchemaProperty.hasNillable()`, `SchemaProperty.hasDefault()`, `SchemaProperty.hasFixed()`

XML_OBJECT

public static final int XML_OBJECT

An XML Bean type that inherits from `XmlObject`. See `SchemaProperty.getJavaTypeCode()`.

Method Detail

acceptedNames() Method

public QName[] acceptedNames()

For element properties the set of names that are accepted for this property if this element is the head of a substitution group. This will always have at least one element, ie, the property's name.

---

extendsJavaArray() Method

public boolean extendsJavaArray()

True if there is a Java getter that returns an array.

---

extendsJavaOption() Method

public boolean extendsJavaOption()

True if there is an Java isSet method that tests for presence.

---

extendsJavaSingleton() Method

`public boolean extendsJavaSingleton()`

True if there is a Java getter that returns a singleton.

---

getContainerType() Method

`public SchemaType getContainerType()`

The type within which this property appears

---

getDefaultText() Method

`public String getDefaultText()`

Returns the default or fixed value, if it is consistent. If it is not consistent, then returns null. See `SchemaProperty.hasDefault()` and `SchemaProperty.hasFixed()`.

---

getDefaultValue() Method

`public XmlAnySimpleType getDefaultValue()`

Returns the default or fixed value as a strongly−typed value, if it is consistent. If it is not consistent, then returns null. See `SchemaProperty.hasDefault()` and `SchemaProperty.hasFixed()`.

---

getJavaPropertyName() Method

`public String getJavaPropertyName()`

The Java name for this property. For example, if the method to access this property is called getFirstName, then this method returns the string "FirstName". May be null if the schema type has not been compiled to Java.

---

getJavaSetterDelimiter() Method

`public QNameSet getJavaSetterDelimiter()`

Returns the set of element names which should appear strictly after all occurences of the elements described by this property. For element properties only.

---

getJavaTypeCode() Method

```
public int getJavaTypeCode()
```

Returns the natural Java type for this property. Returns either XML_OBJECT (for complex types) or one of the JAVA_* constants described in this interface.

---

getMaxOccurs() Method

```
public BigInteger getMaxOccurs()
```

Returns a summarized minimum occurrance number. For example, a sequence containing a nonoptional singleton element repeated twice will result in a property getMaxOccurs() of 2.

---

getMinOccurs() Method

```
public BigInteger getMinOccurs()
```

Returns a summarized minimum occurrance number. For example, a sequence containing a nonoptional singleton element repeated twice will result in a property getMinOccurs() of 2.

---

getName() Method

```
public QName getName()
```

The name of this element or attribute

---

getType() Method

```
public SchemaType getType()
```

The schema type for the property.

---

hasDefault() Method

```
public int hasDefault()
```

Returns `SchemaProperty.NEVER`, `SchemaProperty.VARIABLE`, or `SchemaProperty.CONSISTENTLY` defaulted, depending on the defaults present in the elements in this property.

---

hasFixed() Method

```
public int hasFixed()
```

Returns `SchemaProperty.NEVER`, `SchemaProperty.VARIABLE`, or `SchemaProperty.CONSISTENTLY` fixed, depending on the fixed constraints present in the elements in this property.

---

hasNillable() Method

```
public int hasNillable()
```

Returns `SchemaProperty.NEVER`, `SchemaProperty.VARIABLE`, or `SchemaProperty.CONSISTENTLY` nillable, depending on the nillability of the elements in this property.

---

isAttribute() Method

```
public boolean isAttribute()
```

True for attributes.

---

isReadOnly() Method

```
public boolean isReadOnly()
```

True for read−only properties.

---

javaBasedOnType() Method

```
public SchemaType javaBasedOnType()
```

The schema type returned from the Java getter for this property. Applies only to types that have been code generated to Java; may be a base type of getType().

SchemaStringEnumEntry Interface

public interface ***SchemaStringEnumEntry***

Describes a code−generated string enumeration

Related Topics

`SchemaType.getStringEnumEntries()`

Method Summary

`public String` ***getEnumName()***	A capitalized name to use for the enumeration constant name
`public int` ***getIntValue()***	An integer to represent this choice in the enumeration
`public String` ***getString()***	The enumerated string value

Method Detail

getEnumName() Method

`public String getEnumName()`

A capitalized name to use for the enumeration constant name

---

getIntValue() Method

`public int getIntValue()`

An integer to represent this choice in the enumeration

---

getString() Method

`public String getString()`

The enumerated string value

com.bea.xml

SchemaType Interface

public interface ***SchemaType***

extends `SchemaComponent`

Represents a schema type.

SchemaType is the metadata "type" class for `XmlObject`, and it plays the analogous role for `XmlObject` that `Class` plays for `Object`.

Every XML Bean class corresponds to a singleton SchemaType object obtainable by ClassName.type (e.g., `XmlNonPositiveInteger.type`), and every XML Bean instance has an actual SchemaType, obtainable by `XmlObject.schemaType()`. The ClassName.type and schemaType() mechanisms are analogous to the ordinary Java ClassName.class and obj.getClass() mechanisms.

All schema types are represented by a SchemaType, this includes all types regardless of whether they are built−in or user−defined, compiled or uncompiled, simple or complex.

In addition, a compiled XML Bean type system includes special "document" schema types each representing a document containing nothing but a single element for each global element, and special "attribute" schema types each representing a fragment containing nothing but a single global attribute for global attribute.

Do not confuse Schema Types with other types of Schema Components such as `SchemaGlobalElement`, `SchemaGlobalAttribute`, `SchemaModelGroup`, or `SchemaAttributeGroup`. SchemaType represents a Type component, not any of the other kinds of components. There are different kinds of metadata objects for the different Schema components.

The space of SchemaTypes is divided as follows:

- First, there is the universal base type and the universal subtype. These are `XmlObject.type` (corresponding to xs:anyType) and `XmlBeans.NO_TYPE`, respectively. The first type is a base type of all other types. The other type indicates the absence of type information and, at least in set−theoretic terms, is a subtype of all other types.
- There is another universal base type that is the base type for all simple types. This is the `XmlAnySimpleType.type`, corresponding to xs:anySimpleType. Only XmlObject.type and XmlAnySimpleType.type return `true` for `SchemaType.isURType()`, and only XmlBeans.NO_TYPE returns `true` for `SchemaType.isNoType()`.
- The two "special" kinds of types that are generated that do not formally exist in the actual Schema specification are document types and global attribute types (corresponding to documents that contain a global element, or fragments that contain a global attribute). They can be detected using `SchemaType.isDocumentType()` and `SchemaType.isAttributeType()`. Other than their anonymity (lack of a type name) and their appearance only at the root of an instance, they are otherwise just like ordinary complex types.
- Simple types can be detected using `SchemaType.isSimpleType()`. Complex types are consdered to be all the types that are not simple.
- Simple types are divided into three varieties: atomic types, list types, and union types. Which variety

of simple type you have can be discoverd using `SchemaType.getSimpleVariety()`. It will return either `SchemaType.ATOMIC`, `SchemaType.LIST`, or `SchemaType.UNION`.

- An `SchemaType.ATOMIC` simple type is always based on one of the 20 built−in primitive schema types. You can determine the underlying primitive type for an atomic simple type by calling `SchemaType.getPrimitiveType()`. An atomic type may add facet restrictions on top of the primitive type, and these facets can be explored using `SchemaType.getFacet(int)`, `SchemaType.getWhiteSpaceRule()`, `SchemaType.matchPatternFacet(String)`, `SchemaType.getEnumerationValues()`, and related methods.
- A `SchemaType.LIST` simple type is always based on another non−list simple type. The underlying list item type can be obtained by using `SchemaType.getListItemType()`.
- A `SchemaType.UNION` simple type is always composed out of a number of other simple types. The direct members of the union can be obtained by `SchemaType.getUnionMemberTypes()`. When unions consist of other unions, it is useful to know useful to know the "leaves of the union tree", so the set of non−union types making up the union can be obtained by `SchemaType.getUnionConstituentTypes()`. The closure of the entire "union tree" is `SchemaType.getUnionSubTypes()` (this includes the type itself). For simple unions that do not consist of other unions, all three of these sets are the same.
- Complex types have nested structure. They are divided into four content types: empty content, simple content, element−only content, and mixed content. All kinds of complex types may have attributes. The content type for a complex type can be dermined using `SchemaType.getContentType()`. This will return `SchemaType.EMPTY_CONTENT`, `SchemaType.SIMPLE_CONTENT`, `SchemaType.ELEMENT_CONTENT`, or `SchemaType.MIXED_CONTENT`.
- If a complex type has `SchemaType.EMPTY_CONTENT`, the content model will be null.
- If a complex type has `SchemaType.SIMPLE_CONTENT`, then it will extend the simple type that describes the content. In addition, the type may impose additional simple type facet restrictions; these can be determined in the same way they are for a simple type.
- If a complex type has `SchemaType.ELEMENT_CONTENT` or `SchemaType.MIXED_CONTENT`, then the detailed content model can be determined by examining the particle tree (which may be null for MIXED_CONTENT). The particle tree can be obtained via `SchemaType.getContentModel()`.
- When working with a complex type, most users will find it sufficient to discover the summarized shape of the content model and attribute model using `SchemaType.getElementProperties()`, `SchemaType.getAttributeProperties()`, and related methods rather than examining the particle tree and attribute model directly.

Related Topics

```
SchemaTypeLoader
XmlObject.schemaType()
SimpleValue.instanceType()
```

All Superinterfaces

`SchemaComponent`

Field Summary

public static final int	***ATOMIC*** Atomic type.
public static final int	***BTC_ANY_SIMPLE*** xs:anySimpleType, aka `XmlAnySimpleType.type`
public static final int	***BTC_ANY_TYPE*** xs:anyType, aka `XmlObject.type`
public static final int	***BTC_ANY_URI*** xs:anyURI, aka `XmlAnyURI.type`
public static final int	***BTC_BASE_64_BINARY*** xs:base64Binary, aka `XmlBase64Binary.type`
public static final int	***BTC_BOOLEAN*** xs:boolean, aka `XmlBoolean.type`
public static final int	***BTC_BYTE*** xs:byte, aka `XmlByte.type`
public static final int	***BTC_DATE*** xs:date, aka `XmlDate.type`
public static final int	***BTC_DATE_TIME*** xs:dateTime, aka `XmlDateTime.type`
public static final int	***BTC_DECIMAL*** xs:decimal, aka `XmlDecimal.type`
public static final int	***BTC_DOUBLE*** xs:double, aka `XmlDouble.type`
public static final int	***BTC_DURATION*** xs:duration, aka `XmlDuration.type`
public static final int	***BTC_ENTITIES*** xs:ENTITIES, aka `XmlENTITIES.type`
public static final int	***BTC_ENTITY*** xs:ENTITY, aka `XmlENTITY.type`
public static final int	***BTC_FIRST_PRIMITIVE***

The primitive types have codes between BTC_FIRST_PRIMITIVE and BTC_LAST_PRIMITIVE inclusive

public static final int **BTC_FLOAT**
xs:float, aka `XmlFloat.type`

public static final int **BTC_G_DAY**
xs:gDay, aka `XmlGDay.type`

public static final int **BTC_G_MONTH**
xs:gMonth, aka `XmlGMonth.type`

public static final int **BTC_G_MONTH_DAY**
xs:gMonthDay, aka `XmlGMonthDay.type`

public static final int **BTC_G_YEAR**
xs:gYear, aka `XmlGYear.type`

public static final int **BTC_G_YEAR_MONTH**
xs:gYearMonth, aka `XmlGYearMonth.type`

public static final int **BTC_HEX_BINARY**
xs:hexBinary, aka `XmlBase64Binary.type`

public static final int **BTC_ID**
xs:ID, aka `XmlID.type`

public static final int **BTC_IDREF**
xs:IDREF, aka `XmlIDREF.type`

public static final int **BTC_IDREFS**
xs:IDREFS, aka `XmlIDREFS.type`

public static final int **BTC_INT**
xs:int, aka `XmlInt.type`

public static final int **BTC_INTEGER**
xs:integer, aka `XmlInteger.type`

public static final int **BTC_LANGUAGE**
xs:language, aka `XmlLanguage.type`

public static final int **BTC_LAST_BUILTIN**
int

public static final int **BTC_LAST_PRIMITIVE**
The primitive types have codes between BTC_FIRST_PRIMITIVE and

BTC_LAST_PRIMITIVE inclusive

public static final int **BTC_LONG**
xs:long, aka `XmlLong.type`

public static final int **BTC_NAME**
xs:Name, aka `XmlName.type`

public static final int **BTC_NCNAME**
xs:NCName, aka `XmlNCName.type`

public static final int **BTC_NEGATIVE_INTEGER**
xs:NegativeInteger, aka `XmlNegativeInteger.type`

public static final int **BTC_NMTOKEN**
xs:NMTOKEN, aka `XmlNMTOKEN.type`

public static final int **BTC_NMTOKENS**
xs:NMTOKENS, aka `XmlNMTOKENS.type`

public static final int **BTC_NON_NEGATIVE_INTEGER**
xs:nonNegativeInteger, aka `XmlNonNegativeInteger.type`

public static final int **BTC_NON_POSITIVE_INTEGER**
xs:nonPositiveInteger, aka `XmlNonPositiveInteger.type`

public static final int **BTC_NORMALIZED_STRING**
xs:normalizedString, aka `XmlNormalizedString.type`

public static final int **BTC_NOT_BUILTIN**
Not a builtin type

public static final int **BTC_NOTATION**
xs:NOTATION, aka `XmlNOTATION.type`

public static final int **BTC_POSITIVE_INTEGER**
xs:positiveInteger, aka `XmlPositiveInteger.type`

public static final int **BTC_QNAME**
xs:QName, aka `XmlQName.type`

public static final int **BTC_SHORT**
xs:short, aka `XmlShort.type`

public static final int **BTC_STRING**
xs:string, aka `XmlString.type`

public

static final int **BTC_TIME**
xs:time, aka `XmlTime.type`

public static final int **BTC_TOKEN**
xs:token, aka `XmlToken.type`

public static final int **BTC_UNSIGNED_BYTE**
xs:unsignedByte, aka `XmlUnsignedByte.type`

public static final int **BTC_UNSIGNED_INT**
xs:unsignedInt, aka `XmlUnsignedInt.type`

public static final int **BTC_UNSIGNED_LONG**
xs:unsignedLong, aka `XmlUnsignedLong.type`

public static final int **BTC_UNSIGNED_SHORT**
xs:unsignedShort, aka `XmlUnsignedShort.type`

public static final int **DT_EXTENSION**
Derived by extension.

public static final int **DT_NOT_DERIVED**
Not derived.

public static final int **DT_RESTRICTION**
Derived by restriction.

public static final int **ELEMENT_CONTENT**
Element−only content.

public static final int **EMPTY_CONTENT**
Empty content.

public static final int **FACET_ENUMERATION**
xs:enumeration facet − use `SchemaType.getEnumerationValues()` instead

public static final int **FACET_FRACTION_DIGITS**
xs:fractionDigits facet

public static final int **FACET_LENGTH**
xs:length facet

public static final int **FACET_MAX_EXCLUSIVE**
xs:maxExclusive facet

public static final int **FACET_MAX_INCLUSIVE**
xs:maxInclusive facet

public static final int **FACET_MAX_LENGTH**
xs:maxLength facet

public static final int **FACET_MIN_EXCLUSIVE**
xs:minExclusive facet

public static final int **FACET_MIN_INCLUSIVE**
xs:minInclusive facet

public static final int **FACET_MIN_LENGTH**
xs:minLength facet

public static final int **FACET_PATTERN**
xs:pattern facet − use `SchemaType.matchPatternFacet(String)` instead

public static final int **FACET_TOTAL_DIGITS**
xs:totalDigits facet

public static final int **FACET_WHITE_SPACE**
xs:whiteSpace facet − use `SchemaType.getWhiteSpaceRule()` instead

public static final int **LAST_BASIC_FACET**
int

public static final int **LAST_FACET**
The last ordinary facet code

public static final int **LAST_PROPERTY**
The last property code

public static final int **LIST**
Simple list type.

public static final int **MIXED_CONTENT**
Mixed content.

public static final int **NOT_COMPLEX_TYPE**
Not a complex type.

public static final int **NOT_DECIMAL**
Not a decimal restriction.

public static final int **NOT_SIMPLE**
Not a simple type or simple content.

public static final int **PARTIAL_ORDER**
Partially ordered.

public static final int **PROPERTY_BOUNDED**
int

public static final int **PROPERTY_CARDINALITY**
int

public static final int **PROPERTY_NUMERIC**
int

public static final int **PROPERTY_ORDERED**
int

public static final int **SIMPLE_CONTENT**
Simple content.

public static final int **SIZE_BIG_DECIMAL**
Fits in a `BigDecimal`.

public static final int **SIZE_BIG_INTEGER**
Fits in a `BigInteger`.

public static final int **SIZE_BYTE**
Fits in a byte.

public static final int **SIZE_INT**
Fits in an int.

public static final int **SIZE_LONG**
Fits in a long.

public static final int **SIZE_SHORT**
Fits in a short.

public static final int **TOTAL_ORDER**
Totally ordered.

public static final int **UNION**
Union type.

public static final int **UNORDERED**
Unordered.

public static final int **WS_COLLAPSE**
Whitespace collapsed and trimmed.

public static final int **WS_PRESERVE**
Whitespace preserved.

`public static final int` ***WS_REPLACE***
Whitespace replaced by ordinary space.

`public static final int` ***WS_UNSPECIFIED***
Whitespace rule unspecified.

Fields from interface com.bea.xml.`SchemaComponent`

`ATTRIBUTE, ATTRIBUTE_GROUP, ELEMENT, IDENTITY_CONSTRAINT, MODEL_GROUP, NOTATION, TYPE`

Method Summary

`public boolean` ***blockExtension()***
True if extensions of this type cannot be substituted for this type

`public boolean` ***blockRestriction()***
True if restrictions of this type cannot be substituted for this type

`public SchemaStringEnumEntry` ***enumEntryForString***(`String` *s)*
Returns the string enum entry corresponding to the given enumerated string, or null if there is no match or this type is not a string enumeration.

`public StringEnumAbstractBase` ***enumForInt***(*int i)*
Returns the string enum value corresponding to the given enumerated string, or null if there is no match or this type is not a string enumeration.

`public StringEnumAbstractBase` ***enumForString***(`String` *s)*
Returns the string enum value corresponding to the given enumerated string, or null if there is no match or this type is not a string enumeration.

`public boolean` ***finalExtension()***
True if other types cannot extend this type (only for complex types)

`public boolean` ***finalList()***
True if list derivation of this type is prohibited (only for simple types)

`public boolean` ***finalRestriction()***
True if other types cannot restrict this type

`public boolean` ***finalUnion()***
True if union derivation of this type is prohibited (only for simple types)

`public SchemaType[]` ***getAnonymousTypes()***
The array of inner (anonymous) types defined within this type.

`public int` ***getAnonymousUnionMemberOrdinal()***
For anonymous types defined inside a union only: gets the integer indicating the declaration order of this type within the outer union type, or zero if this is not applicable.

`public SchemaAttributeModel` ***getAttributeModel()***
Returns the attribute model for this complex type (with simple or complex content).

`public SchemaProperty[]` ***getAttributeProperties()***
Returns all the SchemaProperties corresponding to attributes.

`public SchemaProperty` ***getAttributeProperty(QName attrName)***
Returns a SchemaProperty corresponding to an attribute within this complex type by looking up the attribute name.

`public SchemaType` ***getAttributeType(QName eltName, SchemaTypeLoader wildcardTypeLoader)***
Returns the type of an attribute based on the attribute name and the type system within which (wildcard) names are resolved.

`public QName` ***getAttributeTypeAttributeName()***
Returns the attribute qname if this is a attribute type, or null otherwise.

`public SchemaType` ***getBaseEnumType()***
If this is a string enumeration, returns the most basic base schema type that this enuemration is based on.

`public SchemaType` ***getBaseType()***
Returns base restriction or extension type.

`public int` ***getBuiltinTypeCode()***
Returns an integer for builtin types that can be used for quick comparison.

`public SchemaType` ***getCommonBaseType(SchemaType type)***
Returns common base type with the given type.

`public SchemaField`

getContainerField()
The parent schema element.

`public SchemaParticle` ***getContentModel()***
Returns the complex content model for this complex type (with complex content).

`public int` ***getContentType()***
Returns `SchemaType.EMPTY_CONTENT`, `SchemaType.SIMPLE_CONTENT`, `SchemaType.ELEMENT_CONTENT`, or `SchemaType.MIXED_CONTENT` for complex types.

`public int` ***getDecimalSize()***
For atomic numeric restrictions of decimal only: the numeric size category.

`public int` ***getDerivationType()***
Returns an integer for the derivation type, either `SchemaType.DT_EXTENSION`, `SchemaType.DT_RESTRICTION`, `SchemaType.DT_NOT_DERIVED`.

`public SchemaProperty[]` ***getDerivedProperties()***
Returns the SchemaProperties defined by this complex type, exclusive of the base type (if any).

`public QName` ***getDocumentElementName()***
Returns the document element name if this is a document type, or null otherwise.

`public SchemaProperty[]` ***getElementProperties()***
Returns all the SchemaProperties corresponding to elements.

`public SchemaProperty` ***getElementProperty****(`QName` eltName)*
Returns a SchemaProperty corresponding to an element within this complex type by looking up the element name.

`public SchemaTypeElementSequencer` ***getElementSequencer()***
Returns a `SchemaTypeElementSequencer` object, which can then be used to validate complex content inside this element.

`public SchemaType` ***getElementType****(`QName` eltName, `QName` xsiType, `SchemaTypeLoader` wildcardTypeLoader)*
Returns the type of a child element based on the element name and an xsi:type attribute (and the type system within which names are resolved).

`public XmlAnySimpleType[]` ***getEnumerationValues()***
Returns the array of valid objects from the enumeration facet, null if no enumeration defined.

`public Class` ***getEnumJavaClass()***
The Java class corresponding to the enumeration type for this schema type, if applicable (or null if not an enumeration).

`public XmlAnySimpleType` ***getFacet****(int facetCode)*
Returns the value of the given facet, or null if none is set.

`public String` ***getFullJavaImplName()***
The fully−qualified Java type name of the implementation class.

`public String` ***getFullJavaName()***
The fully−qualified Java type name of the class.

`public Class` ***getJavaClass()***
The Java class corresponding to this schema type.

`public SchemaType` ***getListItemType()***
For list types only: get the item type.

`public QName` ***getName()***
The name used to describe the type in the schema.

`public SchemaType` ***getOuterType()***
The outer schema type.

`public String[]` ***getPatterns()***
True

`public SchemaType` ***getPrimitiveType()***
For atomic types only: get the primitive type underlying this one.

`public SchemaProperty[]` ***getProperties()***
Returns all the SchemaProperties within this complex type, elements followed by attributes.

`public SchemaType.Ref` ***getRef()***
Retruns a SchemaType.Ref pointing to this schema type itself.

`public String` ***getShortJavaImplName()***
The short unqualfiied Java name for the implementation class.

`public String` ***getShortJavaName()***
The short unqualfiied Java name for the class.

`public int` ***getSimpleVariety()***
Returns whether the simple type is ATOMIC, UNION, or LIST.

`public String` ***getSourceName()***
Returns the filename for the XSD file from which this type was loaded.

`public SchemaStringEnumEntry[]` ***getStringEnumEntries()***
Returns the array of SchemaStringEnumEntries for this type: this array includes information about the java constant names used for each string enum entry.

`public SchemaTypeSystem` ***getTypeSystem()***
Returns the SchemaTypeLoader in which this type was defined.

`public SchemaType` ***getUnionCommonBaseType()***
For union types only: get the most specific common base type of the constituent member types.

`public SchemaType[]` ***getUnionConstituentTypes()***
For union types only: get the constituent member types.

`public SchemaType[]` ***getUnionMemberTypes()***
For union types only: get the shallow member types.

`public SchemaType[]` ***getUnionSubTypes()***
For union types only: gets the full tree of member types.

`public int` ***getWhiteSpaceRule()***
For nonunion simple types: get the whitespace rule.

`public boolean` ***hasAllContent()***
True if the complex content model for this complex type is an "all" group.

`public boolean` ***hasAttributeWildcards()***
True if this type permits wildcard attributes.

`public boolean` ***hasElementWildcards()***
True if this type permits element wildcards.

`public boolean` ***hasPatternFacet()***
True if there are regex pattern facents

`public boolean` ***hasStringEnumValues()***

True if this is a string enum where an integer is assigned to each enumerated value.

`public boolean` ***isAbstract()***

True if this type cannot be used directly in instances

`public boolean` ***isAnonymousType()***

True if the Xsd type is anonymous (i.e., not top−level).

`public boolean` ***isAssignableFrom(`SchemaType` type)***

True if the specified type derives from this type (or if it is the same type).

`public boolean` ***isAttributeType()***

True if this is a attribute type.

`public boolean` ***isBounded()***

True if bounded.

`public boolean` ***isBuiltinType()***

True for any of the 40+ built−in types.

`public boolean` ***isCompiled()***

True if this schema type was compiled to have a corresponding Java class.

`public boolean` ***isDocumentType()***

True if this is a document type.

`public boolean` ***isFacetFixed(int facetCode)***

True if the given facet is fixed.

`public boolean` ***isFinite()***

True if finite.

`public boolean` ***isNoType()***

True for the type object that represents a the absence of a determined type.

`public boolean` ***isNumeric()***

True if numeric.

`public boolean` ***isOrderSensitive()***

True if particles have same defaults, nillability, etc, that are invariant when order changes.

`public boolean`

isPrimitiveType()
True for any of the 20 primitive types (plus anySimpleType)

public boolean
isSimpleType()
True for the anySimpleType and any restrictions/unions/lists.

public boolean
isSkippedAnonymousType()
True if this anonymous type has no corresponding Java type.

public boolean
isURType()
True for anyType and anySimpleType.

public boolean
isValidSubstitution(QName *name*)
For document types, true if the given name can be substituted for the document element name.

public boolean
matchPatternFacet(String *s*)
True if the given string matches the pattern facets.

public XmlAnySimpleType
newValue(Object *v*)
Creates an immutable simple type value that does not reside in a tree.

public int
ordered()
True if ordered.

Methods from interface com.bea.xml.SchemaComponent

getComponentRef, getComponentType,

Field Detail

ATOMIC

public static final int ATOMIC

Atomic type. See SchemaType.getSimpleVariety()

BTC_ANY_SIMPLE

public static final int BTC_ANY_SIMPLE

xs:anySimpleType, aka XmlAnySimpleType.type

BTC_ANY_TYPE

public static final int BTC_ANY_TYPE

xs:anyType, aka `XmlObject.type`

BTC_ANY_URI

public static final int BTC_ANY_URI

xs:anyURI, aka `XmlAnyURI.type`

BTC_BASE_64_BINARY

public static final int BTC_BASE_64_BINARY

xs:base64Binary, aka `XmlBase64Binary.type`

BTC_BOOLEAN

public static final int BTC_BOOLEAN

xs:boolean, aka `XmlBoolean.type`

BTC_BYTE

public static final int BTC_BYTE

xs:byte, aka `XmlByte.type`

BTC_DATE

public static final int BTC_DATE

xs:date, aka `XmlDate.type`

BTC_DATE_TIME

public static final int BTC_DATE_TIME

xs:dateTime, aka `XmlDateTime.type`

BTC_DECIMAL

public static final int BTC_DECIMAL

xs:decimal, aka `XmlDecimal.type`

BTC_DOUBLE

public static final int BTC_DOUBLE

xs:double, aka `XmlDouble.type`

BTC_DURATION

public static final int BTC_DURATION

xs:duration, aka `XmlDuration.type`

BTC_ENTITIES

public static final int BTC_ENTITIES

xs:ENTITIES, aka `XmlENTITIES.type`

BTC_ENTITY

public static final int BTC_ENTITY

xs:ENTITY, aka `XmlENTITY.type`

BTC_FIRST_PRIMITIVE

public static final int BTC_FIRST_PRIMITIVE

The primitive types have codes between BTC_FIRST_PRIMITIVE and BTC_LAST_PRIMITIVE inclusive

---

BTC_FLOAT

public static final int BTC_FLOAT

xs:float, aka `XmlFloat.type`

BTC_G_DAY

public static final int BTC_G_DAY

xs:gDay, aka `XmlGDay.type`

BTC_G_MONTH

public static final int BTC_G_MONTH

xs:gMonth, aka `XmlGMonth.type`

BTC_G_MONTH_DAY

public static final int BTC_G_MONTH_DAY

xs:gMonthDay, aka `XmlGMonthDay.type`

BTC_G_YEAR

public static final int BTC_G_YEAR

xs:gYear, aka `XmlGYear.type`

BTC_G_YEAR_MONTH

public static final int BTC_G_YEAR_MONTH

xs:gYearMonth, aka `XmlGYearMonth.type`

BTC_HEX_BINARY

public static final int BTC_HEX_BINARY

xs:hexBinary, aka `XmlBase64Binary.type`

BTC_ID

public static final int BTC_ID

xs:ID, aka `XmlID.type`

BTC_IDREF

public static final int BTC_IDREF

xs:IDREF, aka `XmlIDREF.type`

BTC_IDREFS

public static final int BTC_IDREFS

xs:IDREFS, aka `XmlIDREFS.type`

BTC_INT

public static final int BTC_INT

xs:int, aka `XmlInt.type`

BTC_INTEGER

public static final int BTC_INTEGER

xs:integer, aka `XmlInteger.type`

BTC_LANGUAGE

public static final int BTC_LANGUAGE

xs:language, aka `XmlLanguage.type`

BTC_LAST_BUILTIN

public static final int BTC_LAST_BUILTIN

BTC_LAST_PRIMITIVE

public static final int BTC_LAST_PRIMITIVE

The primitive types have codes between BTC_FIRST_PRIMITIVE and BTC_LAST_PRIMITIVE inclusive

---

BTC_LONG

public static final int BTC_LONG

xs:long, aka `XmlLong.type`

BTC_NAME

public static final int BTC_NAME

xs:Name, aka `XmlName.type`

BTC_NCNAME

public static final int BTC_NCNAME

xs:NCName, aka `XmlNCName.type`

BTC_NEGATIVE_INTEGER

public static final int BTC_NEGATIVE_INTEGER

xs:NegativeInteger, aka `XmlNegativeInteger.type`

BTC_NMTOKEN

public static final int BTC_NMTOKEN

xs:NMTOKEN, aka `XmlNMTOKEN.type`

BTC_NMTOKENS

public static final int BTC_NMTOKENS

xs:NMTOKENS, aka `XmlNMTOKENS.type`

BTC_NON_NEGATIVE_INTEGER

public static final int BTC_NON_NEGATIVE_INTEGER

xs:nonNegativeInteger, aka `XmlNonNegativeInteger.type`

BTC_NON_POSITIVE_INTEGER

public static final int BTC_NON_POSITIVE_INTEGER

xs:nonPositiveInteger, aka `XmlNonPositiveInteger.type`

BTC_NORMALIZED_STRING

public static final int BTC_NORMALIZED_STRING

xs:normalizedString, aka `XmlNormalizedString.type`

BTC_NOT_BUILTIN

public static final int BTC_NOT_BUILTIN

Not a builtin type

BTC_NOTATION

public static final int BTC_NOTATION

xs:NOTATION, aka `XmlNOTATION.type`

BTC_POSITIVE_INTEGER

public static final int BTC_POSITIVE_INTEGER

xs:positiveInteger, aka `XmlPositiveInteger.type`

BTC_QNAME

public static final int BTC_QNAME

xs:QName, aka `XmlQName.type`

BTC_SHORT

public static final int BTC_SHORT

xs:short, aka `XmlShort.type`

BTC_STRING

public static final int BTC_STRING

xs:string, aka `XmlString.type`

BTC_TIME

public static final int BTC_TIME

xs:time, aka `XmlTime.type`

BTC_TOKEN

public static final int BTC_TOKEN

xs:token, aka `XmlToken.type`

BTC_UNSIGNED_BYTE

public static final int BTC_UNSIGNED_BYTE

xs:unsignedByte, aka `XmlUnsignedByte.type`

BTC_UNSIGNED_INT

public static final int BTC_UNSIGNED_INT

xs:unsignedInt, aka `XmlUnsignedInt.type`

BTC_UNSIGNED_LONG

public static final int BTC_UNSIGNED_LONG

xs:unsignedLong, aka `XmlUnsignedLong.type`

BTC_UNSIGNED_SHORT

public static final int BTC_UNSIGNED_SHORT

xs:unsignedShort, aka `XmlUnsignedShort.type`

DT_EXTENSION

public static final int DT_EXTENSION

Derived by extension. See `SchemaType.getDerivationType()`.

DT_NOT_DERIVED

public static final int DT_NOT_DERIVED

Not derived. True for XmlObject.type only. See `SchemaType.getDerivationType()`.

DT_RESTRICTION

public static final int DT_RESTRICTION

Derived by restriction. See `SchemaType.getDerivationType()`.

ELEMENT_CONTENT

public static final int ELEMENT_CONTENT

Element−only content. See `SchemaType.getContentType()`.

EMPTY_CONTENT

public static final int EMPTY_CONTENT

Empty content. See `SchemaType.getContentType()`.

FACET_ENUMERATION

public static final int FACET_ENUMERATION

xs:enumeration facet − use `SchemaType.getEnumerationValues()` instead

FACET_FRACTION_DIGITS

public static final int FACET_FRACTION_DIGITS

xs:fractionDigits facet

FACET_LENGTH

public static final int FACET_LENGTH

xs:length facet

FACET_MAX_EXCLUSIVE

public static final int FACET_MAX_EXCLUSIVE

xs:maxExclusive facet

FACET_MAX_INCLUSIVE

public static final int FACET_MAX_INCLUSIVE

xs:maxInclusive facet

FACET_MAX_LENGTH

public static final int FACET_MAX_LENGTH

xs:maxLength facet

FACET_MIN_EXCLUSIVE

public static final int FACET_MIN_EXCLUSIVE

xs:minExclusive facet

FACET_MIN_INCLUSIVE

public static final int FACET_MIN_INCLUSIVE

xs:minInclusive facet

FACET_MIN_LENGTH

public static final int FACET_MIN_LENGTH

xs:minLength facet

FACET_PATTERN

public static final int FACET_PATTERN

xs:pattern facet − use `SchemaType.matchPatternFacet(String)` instead

FACET_TOTAL_DIGITS

public static final int FACET_TOTAL_DIGITS

xs:totalDigits facet

FACET_WHITE_SPACE

public static final int FACET_WHITE_SPACE

xs:whiteSpace facet − use `SchemaType.getWhiteSpaceRule()` instead

LAST_BASIC_FACET

public static final int LAST_BASIC_FACET

LAST_FACET

public static final int LAST_FACET

The last ordinary facet code

LAST_PROPERTY

public static final int LAST_PROPERTY

The last property code

LIST

public static final int LIST

Simple list type. See `SchemaType.getSimpleVariety()`

MIXED_CONTENT

public static final int MIXED_CONTENT

Mixed content. See `SchemaType.getContentType()`.

NOT_COMPLEX_TYPE

public static final int NOT_COMPLEX_TYPE

Not a complex type. See `SchemaType.getContentType()`.

NOT_DECIMAL

public static final int NOT_DECIMAL

Not a decimal restriction. See `SchemaType.getDecimalSize()`.

NOT_SIMPLE

public static final int NOT_SIMPLE

Not a simple type or simple content. See `SchemaType.getSimpleVariety()`.

PARTIAL_ORDER

public static final int PARTIAL_ORDER

Partially ordered. See `SchemaType.ordered()`.

PROPERTY_BOUNDED

public static final int PROPERTY_BOUNDED

Related Topics

`SchemaType.isBounded()`

PROPERTY_CARDINALITY

public static final int PROPERTY_CARDINALITY

Related Topics

`SchemaType.isFinite()`

PROPERTY_NUMERIC

`public static final int PROPERTY_NUMERIC`

Related Topics

`SchemaType.isNumeric()`

PROPERTY_ORDERED

`public static final int PROPERTY_ORDERED`

Related Topics

`SchemaType.ordered()`

SIMPLE_CONTENT

`public static final int SIMPLE_CONTENT`

Simple content. See `SchemaType.getContentType()`.

SIZE_BIG_DECIMAL

`public static final int SIZE_BIG_DECIMAL`

Fits in a `BigDecimal`. See `SchemaType.getDecimalSize()`.

SIZE_BIG_INTEGER

`public static final int SIZE_BIG_INTEGER`

Fits in a `BigInteger`. See `SchemaType.getDecimalSize()`.

SIZE_BYTE

`public static final int SIZE_BYTE`

Fits in a byte. See `SchemaType.getDecimalSize()`.

SIZE_INT

public static final int SIZE_INT

Fits in an int. See `SchemaType.getDecimalSize()`.

SIZE_LONG

public static final int SIZE_LONG

Fits in a long. See `SchemaType.getDecimalSize()`.

SIZE_SHORT

public static final int SIZE_SHORT

Fits in a short. See `SchemaType.getDecimalSize()`.

TOTAL_ORDER

public static final int TOTAL_ORDER

Totally ordered. See `SchemaType.ordered()`.

UNION

public static final int UNION

Union type. See `SchemaType.getSimpleVariety()`

UNORDERED

public static final int UNORDERED

Unordered. See `SchemaType.ordered()`.

WS_COLLAPSE

public static final int WS_COLLAPSE

Whitespace collapsed and trimmed. See `SchemaType.getWhiteSpaceRule()`.

WS_PRESERVE

`public static final int WS_PRESERVE`

Whitespace preserved. See `SchemaType.getWhiteSpaceRule()`.

WS_REPLACE

`public static final int WS_REPLACE`

Whitespace replaced by ordinary space. See `SchemaType.getWhiteSpaceRule()`.

WS_UNSPECIFIED

`public static final int WS_UNSPECIFIED`

Whitespace rule unspecified. See `SchemaType.getWhiteSpaceRule()`.

Method Detail

blockExtension() Method

`public boolean blockExtension()`

True if extensions of this type cannot be substituted for this type

---

blockRestriction() Method

`public boolean blockRestriction()`

True if restrictions of this type cannot be substituted for this type

---

enumEntryForString(String) Method

`public SchemaStringEnumEntry enumEntryForString(String s)`

Returns the string enum entry corresponding to the given enumerated string, or null if there is no match or this type is not a string enumeration.

---

enumForInt(int) Method

```
public StringEnumAbstractBase enumForInt(int i)
```

Returns the string enum value corresponding to the given enumerated string, or null if there is no match or this type is not a string enumeration.

---

enumForString(String) Method

```
public StringEnumAbstractBase enumForString(String s)
```

Returns the string enum value corresponding to the given enumerated string, or null if there is no match or this type is not a string enumeration.

---

finalExtension() Method

```
public boolean finalExtension()
```

True if other types cannot extend this type (only for complex types)

---

finalList() Method

```
public boolean finalList()
```

True if list derivation of this type is prohibited (only for simple types)

---

finalRestriction() Method

```
public boolean finalRestriction()
```

True if other types cannot restrict this type

---

finalUnion() Method

```
public boolean finalUnion()
```

True if union derivation of this type is prohibited (only for simple types)

---

getAnonymousTypes() Method

```
public SchemaType[] getAnonymousTypes()
```

The array of inner (anonymous) types defined within this type.

---

getAnonymousUnionMemberOrdinal() Method

```
public int getAnonymousUnionMemberOrdinal()
```

For anonymous types defined inside a union only: gets the integer indicating the declaration order of this type within the outer union type, or zero if this is not applicable. The first anonymous union member within a union type is numbered "1". Used to differentiate between different anonymous types.

---

getAttributeModel() Method

```
public SchemaAttributeModel getAttributeModel()
```

Returns the attribute model for this complex type (with simple or complex content).

---

getAttributeProperties() Method

```
public SchemaProperty[] getAttributeProperties()
```

Returns all the SchemaProperties corresponding to attributes.

---

getAttributeProperty(QName) Method

```
public SchemaProperty getAttributeProperty(QName attrName)
```

Returns a SchemaProperty corresponding to an attribute within this complex type by looking up the attribute name.

---

getAttributeType(QName, SchemaTypeLoader) Method

```
public SchemaType getAttributeType(QName eltName,
                                   SchemaTypeLoader wildcardTypeLoader)
```

Returns the type of an attribute based on the attribute name and the type system within which (wildcard) names are resolved.

---

getAttributeTypeAttributeName() Method

```
public QName getAttributeTypeAttributeName()
```

Returns the attribute qname if this is a attribute type, or null otherwise.

---

getBaseEnumType() Method

`public SchemaType getBaseEnumType()`

If this is a string enumeration, returns the most basic base schema type that this enuemration is based on. Otherwise returns null.

---

getBaseType() Method

`public SchemaType getBaseType()`

Returns base restriction or extension type. Unions and lists return the anySimpleType.

---

getBuiltinTypeCode() Method

`public int getBuiltinTypeCode()`

Returns an integer for builtin types that can be used for quick comparison.

---

getCommonBaseType(SchemaType) Method

`public SchemaType getCommonBaseType(SchemaType` *`type`***`)`**

Returns common base type with the given type. The returned type is the most specific declared base type of both types.

---

getContainerField() Method

`public SchemaField getContainerField()`

The parent schema element. Null for top−level (named) types and document types.

---

getContentModel() Method

`public SchemaParticle getContentModel()`

Returns the complex content model for this complex type (with complex content).

---

getContentType() Method

`public int getContentType()`

Returns `SchemaType.EMPTY_CONTENT`, `SchemaType.SIMPLE_CONTENT`, `SchemaType.ELEMENT_CONTENT`, or `SchemaType.MIXED_CONTENT` for complex types. For

noncomplex types, returns `SchemaType.NOT_COMPLEX_TYPE`.

---

getDecimalSize() Method

`public int getDecimalSize()`

For atomic numeric restrictions of decimal only: the numeric size category. Takes into account min and max restrictions as well as totalDigits and fractionDigits facets.

Returns either `SchemaType.NOT_DECIMAL`, `SchemaType.SIZE_BYTE`, `SchemaType.SIZE_SHORT`, `SchemaType.SIZE_INT`, `SchemaType.SIZE_LONG`, `SchemaType.SIZE_BIG_INTEGER`, or `SchemaType.SIZE_BIG_DECIMAL`.

---

getDerivationType() Method

`public int getDerivationType()`

Returns an integer for the derivation type, either `SchemaType.DT_EXTENSION`, `SchemaType.DT_RESTRICTION`, `SchemaType.DT_NOT_DERIVED`.

---

getDerivedProperties() Method

`public SchemaProperty[] getDerivedProperties()`

Returns the SchemaProperties defined by this complex type, exclusive of the base type (if any).

---

getDocumentElementName() Method

`public QName getDocumentElementName()`

Returns the document element name if this is a document type, or null otherwise.

---

getElementProperties() Method

`public SchemaProperty[] getElementProperties()`

Returns all the SchemaProperties corresponding to elements.

---

getElementProperty(QName) Method

`public SchemaProperty getElementProperty(QName` *`eltName`*`)`

Returns a SchemaProperty corresponding to an element within this complex type by looking up the element name.

getElementSequencer() Method

```
public SchemaTypeElementSequencer getElementSequencer()
```

Returns a `SchemaTypeElementSequencer` object, which can then be used to validate complex content inside this element. This is useful for example for trying out different names and see which one would be valid as a child of this element.

getElementType(QName, QName, SchemaTypeLoader) Method

```
public SchemaType getElementType(QName eltName,
                                 QName xsiType,
                                 SchemaTypeLoader wildcardTypeLoader)
```

Returns the type of a child element based on the element name and an xsi:type attribute (and the type system within which names are resolved).

getEnumerationValues() Method

```
public XmlAnySimpleType[] getEnumerationValues()
```

Returns the array of valid objects from the enumeration facet, null if no enumeration defined.

getEnumJavaClass() Method

```
public Class getEnumJavaClass()
```

The Java class corresponding to the enumeration type for this schema type, if applicable (or null if not an enumeration).

getFacet(int) Method

```
public XmlAnySimpleType getFacet(int facetCode)
```

Returns the value of the given facet, or null if none is set.

getFullJavaImplName() Method

```
public String getFullJavaImplName()
```

The fully−qualified Java type name of the implementation class.

getFullJavaName() Method

```
public String getFullJavaName()
```

The fully−qualified Java type name of the class.

---

getJavaClass() Method

```
public Class getJavaClass()
```

The Java class corresponding to this schema type.

---

getListItemType() Method

```
public SchemaType getListItemType()
```

For list types only: get the item type. This is the atomic or union type that is the type of every entry in the list.

Returns null if this type is not a list.

---

getName() Method

```
public QName getName()
```

The name used to describe the type in the schema. Null if the type is anonymous (nested), or if it is a document type.

---

getOuterType() Method

```
public SchemaType getOuterType()
```

The outer schema type. Null for top−level (named) types.

---

getPatterns() Method

```
public String[] getPatterns()
```

True

---

getPrimitiveType() Method

```
public SchemaType getPrimitiveType()
```

For atomic types only: get the primitive type underlying this one.

Returns null if this is not an atomic type.

---

getProperties() Method

`public SchemaProperty[] getProperties()`

Returns all the SchemaProperties within this complex type, elements followed by attributes.

---

getRef() Method

`public SchemaType.Ref getRef()`

Retruns a SchemaType.Ref pointing to this schema type itself.

---

getShortJavaImplName() Method

`public String getShortJavaImplName()`

The short unqualfiied Java name for the implementation class.

---

getShortJavaName() Method

`public String getShortJavaName()`

The short unqualfiied Java name for the class.

---

getSimpleVariety() Method

`public int getSimpleVariety()`

Returns whether the simple type is ATOMIC, UNION, or LIST. Returns `SchemaType.NOT_SIMPLE`, `SchemaType.ATOMIC`, `SchemaType.UNION`, or `SchemaType.LIST`.

---

getSourceName() Method

`public String getSourceName()`

Returns the filename for the XSD file from which this type was loaded. Intended for use when debugging.

---

getStringEnumEntries() Method

`public SchemaStringEnumEntry[] getStringEnumEntries()`

Returns the array of SchemaStringEnumEntries for this type: this array includes information about the java constant names used for each string enum entry.

---

getTypeSystem() Method

```
public SchemaTypeSystem getTypeSystem()
```

Returns the SchemaTypeLoader in which this type was defined. Complex types are defined and used in exactly one schema type system, but simple types are defined in one type system and can be used in any number of type systems. The most common case is the builtin types, which are defined in the builtin type system and used elsewhere.

---

getUnionCommonBaseType() Method

```
public SchemaType getUnionCommonBaseType()
```

For union types only: get the most specific common base type of the constituent member types. May return a UR type.

Returns null if this type is not a union.

---

getUnionConstituentTypes() Method

```
public SchemaType[] getUnionConstituentTypes()
```

For union types only: get the constituent member types. This returns only non−union types, so, for example, for unions of unions, this returns the flattened list of individual member types within the innermost unions.

Returns null if this type is not a union.

---

getUnionMemberTypes() Method

```
public SchemaType[] getUnionMemberTypes()
```

For union types only: get the shallow member types. This returns the declared member types of the union, so, for example if the type contains another union, the nested members of that union are NOT returned here.

Returns null if this type is not a union.

---

getUnionSubTypes() Method

```
public SchemaType[] getUnionSubTypes()
```

For union types only: gets the full tree of member types. This computes the closure of the set returned by getUnionMemberTypes(), so, for example, it returns all the types nested within unions of unions as well as the

top−level members; the set also includes the type itself. If you are seeking only the basic non−union consituents, use getUnionConstituentTypes.

Returns null if this type is not a union.

---

getWhiteSpaceRule() Method

`public int getWhiteSpaceRule()`

For nonunion simple types: get the whitespace rule. This is either `SchemaType.WS_PRESERVE`, `SchemaType.WS_REPLACE`, or `SchemaType.WS_COLLAPSE`. Returns `SchemaType.WS_UNSPECIFIED` for unions and complex types.

---

hasAllContent() Method

`public boolean hasAllContent()`

True if the complex content model for this complex type is an "all" group.

---

hasAttributeWildcards() Method

`public boolean hasAttributeWildcards()`

True if this type permits wildcard attributes. See the attribute model for more information about which wildcards are allowed.

---

hasElementWildcards() Method

`public boolean hasElementWildcards()`

True if this type permits element wildcards. See the content model for more information about which wildcards are allowed, and where.

---

hasPatternFacet() Method

`public boolean hasPatternFacet()`

True if there are regex pattern facents

---

hasStringEnumValues() Method

`public boolean hasStringEnumValues()`

True if this is a string enum where an integer is assigned to each enumerated value.

isAbstract() Method

`public boolean isAbstract()`

True if this type cannot be used directly in instances

isAnonymousType() Method

`public boolean isAnonymousType()`

True if the Xsd type is anonymous (i.e., not top−level).

isAssignableFrom(SchemaType) Method

`public boolean isAssignableFrom(SchemaType` *`type`***`)`**

True if the specified type derives from this type (or if it is the same type). Note that XmlObject.type (the anyType) is assignable from all type, and the XmlBeans.noType (the absence of a type) is assignable to all types.

isAttributeType() Method

`public boolean isAttributeType()`

True if this is a attribute type.

Attribute types are generated for every global attribute. An attribute type is an unnamed complex type that contains exactly one attribute: we define these types, because they are the types of the "attribute documents" which contain the defined global attribute, and they all turn into Java types. (Named AttributenameAttribute.)

isBounded() Method

`public boolean isBounded()`

True if bounded.

isBuiltinType() Method

`public boolean isBuiltinType()`

True for any of the 40+ built−in types.

isCompiled() Method

`public boolean isCompiled()`

True if this schema type was compiled to have a corresponding Java class.

---

isDocumentType() Method

`public boolean isDocumentType()`

True if this is a document type.

Document types are generated for every global element. A document type is an unnamed complex type that contains exactly one element: we define these types, because they are the types of the "documents" which contain the defined global elements, and they all turn into Java types. (Named ElementnameDocument.)

---

isFacetFixed(int) Method

`public boolean isFacetFixed(int` *`facetCode`***`)`**

True if the given facet is fixed.

---

isFinite() Method

`public boolean isFinite()`

True if finite.

---

isNoType() Method

`public boolean isNoType()`

True for the type object that represents a the absence of a determined type. XML Objects whose type isNoType() are never valid.

---

isNumeric() Method

`public boolean isNumeric()`

True if numeric.

---

isOrderSensitive() Method

```
public boolean isOrderSensitive()
```

True if particles have same defaults, nillability, etc, that are invariant when order changes. Computed only for Javaized types.

---

isPrimitiveType() Method

```
public boolean isPrimitiveType()
```

True for any of the 20 primitive types (plus anySimpleType)

---

isSimpleType() Method

```
public boolean isSimpleType()
```

True for the anySimpleType and any restrictions/unions/lists.

---

isSkippedAnonymousType() Method

```
public boolean isSkippedAnonymousType()
```

True if this anonymous type has no corresponding Java type. True for anonymous types nested within simple type restrictions.

---

isURType() Method

```
public boolean isURType()
```

True for anyType and anySimpleType.

---

isValidSubstitution(QName) Method

```
public boolean isValidSubstitution(QName name)
```

For document types, true if the given name can be substituted for the document element name.

---

matchPatternFacet(String) Method

```
public boolean matchPatternFacet(String s)
```

True if the given string matches the pattern facets. Always true if there are no pattern facets.

---

newValue(Object) Method

`public XmlAnySimpleType newValue(Object v)`

Creates an immutable simple type value that does not reside in a tree.

---

ordered() Method

`public int ordered()`

True if ordered. Returns either `SchemaType.UNORDERED`, `SchemaType.PARTIAL_ORDER`, or `SchemaType.TOTAL_ORDER`.

SchemaTypeElementSequencer Interface

public interface ***SchemaTypeElementSequencer***

This class is used to programatically validate the contents of an XML element.Call to both `SchemaTypeElementSequencer.next(QName)` and `SchemaTypeElementSequencer.peek(QName)` will return true if the element with the provided name is allowed at the current position in the element content, the difference being that `SchemaTypeElementSequencer.next(QName)` will advance the current position, while `SchemaTypeElementSequencer.peek(QName)` won't.

Related Topics

`SchemaType.getElementSequencer()`

Method Summary

`public boolean` ***next****(QName elementName)*
Returns true if the element with the given name is valid at the current position.

`public boolean` ***peek****(QName elementName)*
Return true if the element with the given name is valid at the current position.

Method Detail

next(QName) Method

`public boolean next(QName` *`elementName`***`)`**

Returns true if the element with the given name is valid at the current position. Advances the current position.

---

peek(QName) Method

`public boolean peek(QName` *`elementName`***`)`**

Return true if the element with the given name is valid at the current position. Does not advance the current position.

com.bea.xml

SchemaTypeLoader Interface

public interface ***SchemaTypeLoader***

Represents a searchable set of XML Schema component definitions.

SchemaTypeLoader is somewhat analogous to `ClassLoader`, because it is responsible for finding `SchemaComponent` definitions by name, yet it is not responsible for being able to enumerate all the component definitons available. (If you wish to enumerate component definitions, see `SchemaTypeSystem`.) There are some ways in which SchemaTypeSystems are dissimilar from ClassLoaders, however. Since XML Schema has a number of instance−oriented typing mechanisms (such as wildcards) that do not exist in Java, a SchemaTypeLoader is not associated with a type; instead, a SchemaTypeLoader is associated with each XML instance.

Every XML instance is loaded within the context of a SchemaTypeLoader; the SchemaTypeLoader for an instance is used to resolve all type definitions within the instance and for applying type−sensitive methods such as `XmlObject.validate()`.

Normally the SchemaTypeLoader being used for all instances is the context type loader (that is, the SchemaTypeLoader returned from `XmlBeans.getContextTypeLoader()`). The context type loader consults the thread's context ClassLoader (see `Thread.getContextClassLoader()`) to find schema type defintions that are available on the classpath. The net result is that you can use schema types simply by putting their compiled schema JARs on your classpath. If you wish to load instances using a different SchemaTypeLoader, then you must call `SchemaTypeLoader.parse(String, SchemaType, XmlOptions)` methods on the SchemaTypeLoader instance explicitly rather than using the normal convenient Factory methods.

A SchemaTypeLoader can be obtained by dynamically loading XSD files using `XmlBeans.loadXsd(XmlObject[])`, or by assembling other SchemaTypeLoaders or SchemaTypeSystems on a path using `XmlBeans.typeLoaderUnion(SchemaTypeLoader[])`.

Related Topics

```
XmlBeans.loadXsd(XmlObject[])
XmlBeans.getContextTypeLoader()
XmlBeans.typeLoaderUnion(SchemaTypeLoader[])
SchemaTypeSystem
```

All Known Subinterfaces

`SchemaTypeSystem`

Method Summary

`public String` ***compilePath***(`String` *pathExpr*, `XmlOptions` *options*)

Compiles an XPath

`public String` **`compileQuery`**(`String` *queryExpr*, `XmlOptions` *options*)
Compiles an XQuery

`public SchemaGlobalAttribute` **`findAttribute`**(`QName` *name*)
Returns the global attribute defintion with the given name, or null if none.

`public SchemaAttributeGroup` **`findAttributeGroup`**(`QName` *name*)
Returns the attribute group defintion with the given name, or null if none.

`public SchemaAttributeGroup.Ref` **`findAttributeGroupRef`**(`QName` *name*)
Used for on−demand loading.

`public SchemaGlobalAttribute.Ref` **`findAttributeRef`**(`QName` *name*)
Used for on−demand loading.

`public SchemaType` **`findAttributeType`**(`QName` *name*)
Returns the attribute type containing the given attribute name, or null if none.

`public SchemaType.Ref` **`findAttributeTypeRef`**(`QName` *name*)
Used for on−demand loading.

`public SchemaType` **`findDocumentType`**(`QName` *name*)
Returns the document type rooted at the given element name, or null if none.

`public SchemaType.Ref` **`findDocumentTypeRef`**(`QName` *name*)
Used for on−demand loading.

`public SchemaGlobalElement` **`findElement`**(`QName` *name*)
Returns the global element defintion with the given name, or null if none.

`public SchemaGlobalElement.Ref` **`findElementRef`**(`QName` *name*)
Used for on−demand loading.

`public SchemaIdentityConstraint.Ref` **`findIdentityConstraintRef`**(`QName` *name*)
Used for on−demand loading.

`public SchemaModelGroup` **`findModelGroup`**(`QName` *name*)
Returns the model group defintion with the given name, or null if none.

`public SchemaModelGroup.Ref`

findModelGroupRef(QName *name*)
Used for on−demand loading.

public SchemaType
findType(QName *name*)
Returns the type with the given name, or null if none.

public SchemaType.Ref
findTypeRef(QName *name*)
Used for on−demand loading.

public InputStream
getSourceAsStream(String *sourceName*)
Loads original XSD source as a stream.

public boolean
isNamespaceDefined(String *namespace*)
True if the typeloader contains any definitions in the given namespace.

public XmlObject
newInstance(SchemaType *type*, XmlOptions *options*)
Creates an instance of the given type.

public XMLInputStream
newValidatingXMLInputStream(XMLInputStream *xis*, SchemaType *type*, XmlOptions *options*)
Returns a validating XMLInputStream that will throw an exception if the XML is not valid

public XmlSaxHandler
newXmlSaxHandler(SchemaType *type*, XmlOptions *options*)
Returns an XmlSaxHandler that can parse an instance of the given type.

public XmlObject
parse(String *xmlText*, SchemaType *type*, XmlOptions *options*)
Parses an instance of the given type.

public XmlObject
parse(File *file*, SchemaType *type*, XmlOptions *options*)
Parses an instance of the given type.

public XmlObject
parse(URL *file*, SchemaType *type*, XmlOptions *options*)
Parses an instance of the given type.

public XmlObject
parse(InputStream *jiois*, SchemaType *type*, XmlOptions *options*)
Parses an instance of the given type.

public XmlObject
parse(Reader *jior*, SchemaType *type*, XmlOptions *options*)
Parses an instance of the given type.

public XmlObject
parse(Node *node*, SchemaType *type*, XmlOptions *options*)

Parses an instance of the given type.

`public XmlObject` **`parse`**`(XMLInputStream xis, SchemaType type, XmlOptions options)`

Parses an instance of the given type.

`public SchemaType` **`typeForClassname`**`(String classname)`

Finds a type for a given fully−qualified XML Bean classname

`public SchemaType` **`typeForSignature`**`(String signature)`

Finds a type for a given signature string

Method Detail

compilePath(String, XmlOptions) Method

```
public String compilePath(String pathExpr,
                          XmlOptions options)
     throws XmlException
```

Compiles an XPath

Exceptions

XmlException

---

compileQuery(String, XmlOptions) Method

```
public String compileQuery(String queryExpr,
                           XmlOptions options)
     throws XmlException
```

Compiles an XQuery

Exceptions

XmlException

---

findAttribute(QName) Method

```
public SchemaGlobalAttribute findAttribute(QName name)
```

Returns the global attribute defintion with the given name, or null if none.

---

findAttributeGroup(QName) Method

public SchemaAttributeGroup findAttributeGroup(QName *name*)

Returns the attribute group defintion with the given name, or null if none.

---

findAttributeGroupRef(QName) Method

public SchemaAttributeGroup.Ref findAttributeGroupRef(QName *name*)

Used for on−demand loading.

---

findAttributeRef(QName) Method

public SchemaGlobalAttribute.Ref findAttributeRef(QName *name*)

Used for on−demand loading.

---

findAttributeType(QName) Method

public SchemaType findAttributeType(QName *name*)

Returns the attribute type containing the given attribute name, or null if none.

---

findAttributeTypeRef(QName) Method

public SchemaType.Ref findAttributeTypeRef(QName *name*)

Used for on−demand loading.

---

findDocumentType(QName) Method

public SchemaType findDocumentType(QName *name*)

Returns the document type rooted at the given element name, or null if none.

---

findDocumentTypeRef(QName) Method

public SchemaType.Ref findDocumentTypeRef(QName *name*)

Used for on−demand loading.

---

findElement(QName) Method

public SchemaGlobalElement findElement(QName *name*)

Returns the global element defintion with the given name, or null if none.

---

findElementRef(QName) Method

public SchemaGlobalElement.Ref findElementRef(QName *name*)

Used for on−demand loading.

---

findIdentityConstraintRef(QName) Method

public SchemaIdentityConstraint.Ref findIdentityConstraintRef(QName *name*)

Used for on−demand loading.

---

findModelGroup(QName) Method

public SchemaModelGroup findModelGroup(QName *name*)

Returns the model group defintion with the given name, or null if none.

---

findModelGroupRef(QName) Method

public SchemaModelGroup.Ref findModelGroupRef(QName *name*)

Used for on−demand loading.

---

findType(QName) Method

public SchemaType findType(QName *name*)

Returns the type with the given name, or null if none.

---

findTypeRef(QName) Method

public SchemaType.Ref findTypeRef(QName *name*)

Used for on−demand loading.

---

getSourceAsStream(String) Method

```
public InputStream getSourceAsStream(String sourceName)
```

Loads original XSD source as a stream. See `SchemaType.getSourceName()`.

---

isNamespaceDefined(String) Method

```
public boolean isNamespaceDefined(String namespace)
```

True if the typeloader contains any definitions in the given namespace.

---

newInstance(SchemaType, XmlOptions) Method

```
public XmlObject newInstance(SchemaType type,
                             XmlOptions options)
```

Creates an instance of the given type.

---

newValidatingXMLInputStream(XMLInputStream, SchemaType, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                  SchemaType type,
                                                  XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream that will throw an exception if the XML is not valid

Exceptions

XmlException
XMLStreamException

---

newXmlSaxHandler(SchemaType, XmlOptions) Method

```
public XmlSaxHandler newXmlSaxHandler(SchemaType type,
                                      XmlOptions options)
```

Returns an XmlSaxHandler that can parse an instance of the given type.

---

parse(String, SchemaType, XmlOptions) Method

```
public XmlObject parse(String xmlText,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException
```

Parses an instance of the given type.

Exceptions

XmlException

---

parse(File, SchemaType, XmlOptions) Method

```
public XmlObject parse(File file,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException, IOException
```

Parses an instance of the given type.

Exceptions

XmlException
IOException

---

parse(URL, SchemaType, XmlOptions) Method

```
public XmlObject parse(URL file,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException, IOException
```

Parses an instance of the given type.

Exceptions

XmlException
IOException

---

parse(InputStream, SchemaType, XmlOptions) Method

```
public XmlObject parse(InputStream jiois,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException, IOException
```

Parses an instance of the given type.

Exceptions

XmlException
IOException

---

parse(Reader, SchemaType, XmlOptions) Method

```
public XmlObject parse(Reader jior,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException, IOException
```

Parses an instance of the given type.

Exceptions

XmlException
IOException

---

parse(Node, SchemaType, XmlOptions) Method

```
public XmlObject parse(Node node,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException
```

Parses an instance of the given type.

Exceptions

XmlException

---

parse(XMLInputStream, SchemaType, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public XmlObject parse(XMLInputStream xis,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses an instance of the given type.

Exceptions

XmlException
XMLStreamException

---

typeForClassname(String) Method

```
public SchemaType typeForClassname(String classname)
```

Finds a type for a given fully−qualified XML Bean classname

---

typeForSignature(String) Method

```
public SchemaType typeForSignature(String signature)
```

Finds a type for a given signature string

com.bea.xml

SchemaTypeLoaderException Class

public class ***SchemaTypeLoaderException***

extends `XmlRuntimeException`

An exception that is thrown if there is corruption or a version mismatch in a compiled schema type system.

Hierarchy

```
Object
 Throwable
  Exception
   RuntimeException
    XmlRuntimeException
     SchemaTypeLoaderException
```

All Implemented Interfaces

`Serializable`

Field Summary

public static final int	***BAD_HANDLE*** int
public static final int	***BAD_PARTICLE_TYPE*** int
public static final int	***INT_TOO_LARGE*** int
public static final int	***IO_EXCEPTION*** int
public static final int	***MALFORMED_CONTENT_MODEL*** int
public static final int	***NESTED_EXCEPTION*** int
public static final int	***NO_RESOURCE*** int

public static final int ***NOT_WRITEABLE***
int

public static final int ***UNRECOGNIZED_INDEX_ENTRY***
int

public static final int ***WRONG_FILE_TYPE***
int

public static final int ***WRONG_MAGIC_COOKIE***
int

public static final int ***WRONG_MAJOR_VERSION***
int

public static final int ***WRONG_MINOR_VERSION***
int

public static final int ***WRONG_PROPERTY_TYPE***
int

public static final int ***WRONG_SIMPLE_VARIETY***
int

Constructor Summary

SchemaTypeLoaderException(`String` *message*, `String` *name*, `String` *handle*, *int code*, `Exception` *cause*)

Constructs an exception with the given message, filename, extension, code, and cause

SchemaTypeLoaderException(`String` *message*, `String` *name*, `String` *handle*, *int code*)

Constructs an exception with the given message, filename, extension, and code

Method Summary

public int ***getCode***()
Returns the reason for the failure, given by one of the numeric constants in this class

Methods from `com.bea.xml.XmlRuntimeException`

getError, getErrors

Methods from `java.lang.Throwable`

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

BAD_HANDLE

public static final int BAD_HANDLE

BAD_PARTICLE_TYPE

public static final int BAD_PARTICLE_TYPE

INT_TOO_LARGE

public static final int INT_TOO_LARGE

IO_EXCEPTION

public static final int IO_EXCEPTION

MALFORMED_CONTENT_MODEL

public static final int MALFORMED_CONTENT_MODEL

NESTED_EXCEPTION

public static final int NESTED_EXCEPTION

NO_RESOURCE

public static final int NO_RESOURCE

NOT_WRITEABLE

```
public static final int NOT_WRITEABLE
```

UNRECOGNIZED_INDEX_ENTRY

```
public static final int UNRECOGNIZED_INDEX_ENTRY
```

WRONG_FILE_TYPE

```
public static final int WRONG_FILE_TYPE
```

WRONG_MAGIC_COOKIE

```
public static final int WRONG_MAGIC_COOKIE
```

WRONG_MAJOR_VERSION

```
public static final int WRONG_MAJOR_VERSION
```

WRONG_MINOR_VERSION

```
public static final int WRONG_MINOR_VERSION
```

WRONG_PROPERTY_TYPE

```
public static final int WRONG_PROPERTY_TYPE
```

WRONG_SIMPLE_VARIETY

```
public static final int WRONG_SIMPLE_VARIETY
```

Constructor Detail

SchemaTypeLoaderException

```
public SchemaTypeLoaderException(String message,
                                 String name,
                                 String handle,
                                 int code,
                                 Exception cause)
```

Constructs an exception with the given message, filename, extension, code, and cause

---

SchemaTypeLoaderException

```
public SchemaTypeLoaderException(String message,
                                 String name,
                                 String handle,
                                 int code)
```

Constructs an exception with the given message, filename, extension, and code

Method Detail

getCode() Method

```
public int getCode()
```

Returns the reason for the failure, given by one of the numeric constants in this class

com.bea.xml

SchemaTypeSystem Interface

public interface ***SchemaTypeSystem***

extends `SchemaTypeLoader`

A finite set of XML Schema component definitions.

Every `SchemaComponent` such as a `SchemaType`, `SchemaGlobalElement`, `SchemaGlobalAttribute`, `SchemaModelGroup`, `SchemaAttributeGroup`, or `SchemaIdentityConstraint`, is defined in exactly one SchemaTypeSystem. (See `SchemaComponent.getTypeSystem()`.) A single SchemaTypeSystem can include definitions from any number of namespaces; one SchemaTypeSystem consists simply of a set of component definitions that were compiled together.

Since every component is defined in a single SchemaTypeSystem, no SchemaTypeSystem other than `XmlBeans.getBuiltinTypeSystem()` includes any of the the built−in types. That means you cannot ordinarily load instances using a single SchemaTypeSystem by itself. Instead, you will want to combine a path of SchemaTypeSystems together using `XmlBeans.typeLoaderUnion(SchemaTypeLoader[])` to form a SchemaTypeLoader that can be used for loading instances.

For example, the following code compiles the schema in myXSDFile in the presence of only the minimal builtin type system. The resulting SchemaTypeSystem `sts` contains only the definitions from myXSD file. In order to load and validate an instance within the context of those types, we must next construct a `SchemaTypeLoader stl` that contains both the builtin type system and the types defined within the myXSD file.

```
SchemaTypeSystem sts = XmlBeans.compileXsd(new XmlObject[]
   { XmlObject.Factory.parse(myXSDFile) },
   XmlBeans.getBuiltinTypeSystem(),
   null);
SchemaTypeLoader stl = XmlBeans.typeLoaderUnion(new SchemaTypeLoader[]
   { sts, XmlBeans.getBuiltinTypeSystem() });
XmlObject mydoc = stl.parse(instanceFile, null, null);
System.out.println("Document valid: " + mydoc.validate());
```

As you can see, for working with instances, you typically want to work with a SchemaTypeLoader constructed from a path rather than a solitary SchemaTypeSystem. See `XmlBeans.loadXsd(XmlObject[])` for a convenient alternative to `XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`.

A SchemaTypeSystem is useful when you need to enumerate the exact set of component definitions derived from a set of XSD files, for example, when you are analyzing the contents of the XSD files themselves. Here is how to use a SchemaTypeSystem to inspect a set of schema definitions:

1. First, use `XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)` to compile any number of schema files. If the schema files are valid, result will be a SchemaTypeSystem that contains all the component definitions from those files. It will contain no

other component definitions.

2. Alternatively, call `SchemaComponent.getTypeSystem()` on a precompiled schema component to discover the SchemaTypeSystem within which that component was originally compiled.
3. Once you have a SchemaTypeSystem, call:
   - `SchemaTypeSystem.globalTypes()` for all the global type definitions.
   - `SchemaTypeSystem.globalElements()` for all the global element definitions.
   - `SchemaTypeSystem.globalAttributes()` for all the global attribute definitions.
   - `SchemaTypeSystem.modelGroups()` for all the named model group definitions.
   - `SchemaTypeSystem.attributeGroups()` for all the attribute group definitions.
4. In addition, there are special types generated for XML Beans thare are not formally part of the Schema specification:
   - `SchemaTypeSystem.documentTypes()` returns all the document types.
   - `SchemaTypeSystem.attributeTypes()` returns all the attribute types.

A document type is a type that contains a single global element; there is one document type for each global element definition in a SchemaTypeSystem. In an instance document, only the root XmlObject can have a document type as its type.

Similarly, an attribute type is a type that contains a single global attribute, and there is one attribute type for each global attribute definition in a SchemaTypeSystem. It is possible to have a root XmlObject representing a fragment whose type is an attribute type, but attribute types are present mainly for symmetry and to simplify code such as the type−tree−walking code below.

The global component methods above only provide a view of the top−level components of a SchemaTypeSystem and do not include any nested definitions. To view all the nested definitions, you will want to traverse the entire tree of `SchemaType` defintions within a SchemaTypeSystem by examining the `SchemaType.getAnonymousTypes()` within each `SchemaType` recursively.

The following code is a standard treewalk that visits every `SchemaType` in the SchemaTypeSystem once, including nested definitions.

```
List allSeenTypes = new ArrayList();
allSeenTypes.addAll(Arrays.asList(typeSystem.documentTypes()));
allSeenTypes.addAll(Arrays.asList(typeSystem.attributeTypes()));
allSeenTypes.addAll(Arrays.asList(typeSystem.globalTypes()));
for (int i = 0; i <allSeenTypes.size(); i++)
{
    SchemaType sType = (SchemaType)allSeenTypes.get(i);
    System.out.prinlnt("Visiting " + sType.toString());
    allSeenTypes.addAll(Arrays.asList(sType.getAnonymousTypes()));
}
```

Related Topics

`SchemaType`
`SchemaTypeLoader`
`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`
`XmlBeans.typeLoaderUnion(SchemaTypeLoader[])`
`XmlBeans.getBuiltinTypeSystem()`

All Superinterfaces

`SchemaTypeLoader`

Method Summary

`public SchemaAttributeGroup[]`	***attributeGroups()*** Returns the attribute groups defined in this loader.
`public SchemaType[]`	***attributeTypes()*** Returns the attribute types defined in this loader.
`public SchemaType[]`	***documentTypes()*** Returns the document types defined in this loader.
`public ClassLoader`	***getClassLoader()*** Returns the classloader used by this loader for resolving types.
`public String`	***getName()*** Returns the name of this loader.
`public SchemaGlobalAttribute[]`	***globalAttributes()*** Returns the global attributes defined in this loader.
`public SchemaGlobalElement[]`	***globalElements()*** Returns the global elements defined in this loader.
`public SchemaType[]`	***globalTypes()*** Returns the global types defined in this loader.
`public SchemaModelGroup[]`	***modelGroups()*** Returns the model groups defined in this loader.
`public void`	***resolve()*** Initializes a type system (resolves all handles within the type system).
`public SchemaComponent`	***resolveHandle***(`String` *handle)* Locates a type, element, or attribute using the handle.
`public void`	***saveToDirectory***(`File` *classDir)* Saves this type to a directory.
`public SchemaType`	***typeForHandle***(`String` *handle)* Locates a type, element, or attribute using the handle.

Methods from interface `com.bea.xml.SchemaTypeLoader`

`compilePath, compileQuery, findAttribute, findAttributeGroup, findAttributeGroupRef, findAttributeRef, findAttributeType, findAttributeTypeRef, findDocumentType, findDocumentTypeRef, findElement, findElementRef, findIdentityConstraintRef, findModelGroup, findModelGroupRef, findType, findTypeRef, getSourceAsStream, isNamespaceDefined, newInstance, newValidatingXMLInputStream, newXmlSaxHandler, parse, parse, parse, parse, parse, parse, parse, typeForClassname, typeForSignature`

Method Detail

attributeGroups() Method

`public SchemaAttributeGroup[] attributeGroups()`

Returns the attribute groups defined in this loader.

---

attributeTypes() Method

`public SchemaType[] attributeTypes()`

Returns the attribute types defined in this loader.

---

documentTypes() Method

`public SchemaType[] documentTypes()`

Returns the document types defined in this loader.

---

getClassLoader() Method

`public ClassLoader getClassLoader()`

Returns the classloader used by this loader for resolving types.

---

getName() Method

`public String getName()`

Returns the name of this loader.

---

globalAttributes() Method

`public SchemaGlobalAttribute[] globalAttributes()`

Returns the global attributes defined in this loader.

---

globalElements() Method

`public SchemaGlobalElement[] globalElements()`

Returns the global elements defined in this loader.

---

globalTypes() Method

`public SchemaType[] globalTypes()`

Returns the global types defined in this loader.

---

modelGroups() Method

`public SchemaModelGroup[] modelGroups()`

Returns the model groups defined in this loader.

---

resolve() Method

`public void resolve()`

Initializes a type system (resolves all handles within the type system).

---

resolveHandle(String) Method

`public SchemaComponent resolveHandle(String` *`handle`***`)`**

Locates a type, element, or attribute using the handle.

---

saveToDirectory(File) Method

`public void saveToDirectory(File` *`classDir`***`)`**

Saves this type to a directory.

---

typeForHandle(String) Method

public SchemaType typeForHandle(String *handle***)**

Locates a type, element, or attribute using the handle.

SimpleValue Interface

public interface ***SimpleValue***

extends `XmlObject`

All XmlObject implementations can be coerced to SimpleValue. For any given schema type, only a subset of the conversion methods will work. Others will throw an exception.

SimpleValue is useful for declaring variables which can hold more than one similar schema type that may not happen to have a common XML base type, for example, two list types, or two unrelated integer restrictions that happen to fit into an int.

All Known Implementing Classes

`FilterXmlObject`

All Superinterfaces

`XmlObject`, `XmlTokenSource`

Nested Class Summary

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public BigDecimal`
`bigDecimalValue`*()*
Returns the value as a `BigDecimal`

`public BigInteger`
`bigIntegerValue`*()*
Returns the value as a `BigInteger`

`public boolean` ***booleanValue()***
Returns the value as a boolean

`public byte[]` ***byteArrayValue()***
Returns the value as a byte array

`public byte` ***byteValue()***
Returns the value as a byte

`public Calendar` ***calendarValue()***
Returns the value as a `Calendar`

`public Date` ***dateValue()***
Returns the value as a `Date`

`public double` ***doubleValue()***
Returns the value as a double

`public StringEnumAbstractBase` ***enumValue()***
Returns the value as a `StringEnumAbstractBase`

`public float` ***floatValue()***
Returns the value as a float

`public GDate` ***gDateValue()***
Returns the value as a `GDate`

`public GDuration` ***gDurationValue()***
Returns the value as a `GDuration`

`public BigDecimal` ***getBigDecimalValue()***
Returns the value as a `BigDecimal`.

`public BigInteger` ***getBigIntegerValue()***
Returns the value as a `BigInteger`.

`public boolean` ***getBooleanValue()***
Returns the value as a boolean.

`public byte[]` ***getByteArrayValue()***
Returns the value as a byte array.

`public byte` ***getByteValue()***
Returns the value as a byte.

`public Calendar` ***getCalendarValue()***
Returns the value as a `Calendar`.

`public Date` ***getDateValue()***
Returns the value as a `Date`.

`public double` ***getDoubleValue()***
Returns the value as a double.

`public StringEnumAbstractBase` ***getEnumValue()***
Returns the value as a `StringEnumAbstractBase`.

`public float` ***getFloatValue()***
Returns the value as a float.

`public GDate` ***getGDateValue()***
Returns the value as a `GDate`.

`public GDuration` ***getGDurationValue()***
Returns the value as a `GDuration`.

`public int` ***getIntValue()***
Returns the value as an int.

`public List` ***getListValue()***
Returns the value as a `List` of friendly Java objects (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration).

`public long` ***getLongValue()***
Returns the value as a long.

`public Object` ***getObjectValue()***
Returns a union value as a its natural friendly Java object (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration).

`public QName` ***getQNameValue()***
Returns the value as a `QName`.

`public short` ***getShortValue()***
Returns the value as a short.

`public String` ***getStringValue()***
Returns the value as a `String`.

`public SchemaType` ***instanceType()***
The same as getSchemaType unless this is a union instance or nil value.

`public int`

intValue()
Returns the value as an int

`public List`
listValue()
Returns the value as a `List` of friendly Java objects (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration)

`public long`
longValue()
Returns the value as a long

`public void`
objectSet(Object obj)
Sets the value as an arbitrary `Object`.

`public Object`
objectValue()
Returns a union value as a its natural friendly Java object (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration)

`public QName`
qNameValue()
Returns the value as a `QName`

`public void`
set(String obj)
Sets the value as a `String`

`public void`
set(boolean v)
Sets the value as a boolean

`public void`
set(byte v)
Sets the value as a byte.

`public void`
set(short v)
Sets the value as a short.

`public void`
set(int v)
Sets the value as an int.

`public void`
set(long v)
Sets the value as a long.

`public void`
set(BigInteger obj)
Sets the value as a `BigInteger`.

`public void`
set(BigDecimal obj)
Sets the value as a `BigDecimal`

`public void`
set(float v)

Sets the value as a float.

public void
set(*double v*)
Sets the value as a double.

public void
set(*byte[] obj*)
Sets the value as a byte array.

public void
set(*StringEnumAbstractBase obj*)
Sets the value as a `StringEnumAbstractBase`.

public void
set(*Calendar obj*)
Sets the value as a `Calendar`.

public void
set(*Date obj*)
Sets the value as a `Date`.

public void
set(*GDateSpecification obj*)
Sets the value as a `GDate`.

public void
set(*GDurationSpecification obj*)
Sets the value as a `GDuration`.

public void
set(*QName obj*)
Sets the value as a `QName`.

public void
set(*List obj*)
Sets the value as a `List`.

public void
setBigDecimalValue(*BigDecimal obj*)
Sets the value as a `BigDecimal`.

public void
setBigIntegerValue(*BigInteger obj*)
Sets the value as a `BigInteger`.

public void
setBooleanValue(*boolean v*)
Sets the value as a boolean.

public void
setByteArrayValue(*byte[] obj*)
Sets the value as a byte array.

public void
setByteValue(*byte v*)
Sets the value as a byte.

public void
setCalendarValue(*Calendar obj*)
Sets the value as a `Calendar`.

public void

setDateValue(Date obj)
Sets the value as a Date.

public void
setDoubleValue(double v)
Sets the value as a double.

public void
setEnumValue(StringEnumAbstractBase obj)
Sets the value as a StringEnumAbstractBase.

public void
setFloatValue(float v)
Sets the value as a float.

public void
setGDateValue(GDate obj)
Sets the value as a GDate.

public void
setGDurationValue(GDuration obj)
Sets the value as a GDuration.

public void
setIntValue(int v)
Sets the value as an int.

public void
setListValue(List obj)
Sets the value as a List.

public void
setLongValue(long v)
Sets the value as a long.

public void
setObjectValue(Object obj)
Sets the value as an arbitrary Object.

public void
setQNameValue(QName obj)
Sets the value as a QName.

public void
setShortValue(short v)
Sets the value as a short.

public void
setStringValue(String obj)
Sets the value as a String.

public short
shortValue()
Returns the value as a short

public String
stringValue()
Returns the value as a String

public List
xgetListValue()
Returns the value as a List of XmlAnySimpleType objects.

`public List` ***xlistValue()***
Returns the value as a `List` of XmlAnySimpleType objects

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Method Detail

bigDecimalValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getBigDecimalValue()`

`public BigDecimal bigDecimalValue()`

Returns the value as a `BigDecimal`. *

---

bigIntegerValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getBigIntegerValue()`

`public BigInteger bigIntegerValue()`

Returns the value as a `BigInteger`. *

---

booleanValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getBooleanValue()`

`public boolean booleanValue()`

Returns the value as a boolean. *

---

byteArrayValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getByteArrayValue()`

`public byte[] byteArrayValue()`

Returns the value as a byte array. *

---

byteValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getByteValue()`

`public byte byteValue()`

Returns the value as a byte. *

---

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getCalendarValue()`

`public Calendar calendarValue()`

Returns the value as a `Calendar`. *

---

dateValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getDateValue()`

`public Date dateValue()`

Returns the value as a `Date`. *

---

doubleValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getDoubleValue()`

`public double doubleValue()`

Returns the value as a double. *

---

enumValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getEnumValue()`

`public StringEnumAbstractBase enumValue()`

Returns the value as a `StringEnumAbstractBase`. *

---

floatValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getFloatValue()`

`public float floatValue()`

Returns the value as a float. *

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getGDateValue()`

`public GDate gDateValue()`

Returns the value as a `GDate`. *

---

gDurationValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getGDurationValue()`

`public GDuration gDurationValue()`

Returns the value as a `GDuration`. *

---

getBigDecimalValue() Method

`public BigDecimal getBigDecimalValue()`

Returns the value as a `BigDecimal`.

---

getBigIntegerValue() Method

`public BigInteger getBigIntegerValue()`

Returns the value as a `BigInteger`.

---

getBooleanValue() Method

`public boolean getBooleanValue()`

Returns the value as a boolean.

---

getByteArrayValue() Method

`public byte[] getByteArrayValue()`

Returns the value as a byte array.

---

getByteValue() Method

`public byte getByteValue()`

Returns the value as a byte.

---

getCalendarValue() Method

`public Calendar getCalendarValue()`

Returns the value as a `Calendar`.

---

getDateValue() Method

`public Date getDateValue()`

Returns the value as a `Date`.

---

getDoubleValue() Method

`public double getDoubleValue()`

Returns the value as a double.

---

getEnumValue() Method

`public StringEnumAbstractBase getEnumValue()`

Returns the value as a `StringEnumAbstractBase`.

---

getFloatValue() Method

`public float getFloatValue()`

Returns the value as a float.

---

getGDateValue() Method

`public GDate getGDateValue()`

Returns the value as a `GDate`.

---

getGDurationValue() Method

`public GDuration getGDurationValue()`

Returns the value as a `GDuration`.

---

getIntValue() Method

`public int getIntValue()`

Returns the value as an int.

---

getListValue() Method

`public List getListValue()`

Returns the value as a `List` of friendly Java objects (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration).

---

getLongValue() Method

`public long getLongValue()`

Returns the value as a long.

---

getObjectValue() Method

`public Object getObjectValue()`

Returns a union value as a its natural friendly Java object (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration).

---

getQNameValue() Method

`public QName getQNameValue()`

Returns the value as a `QName`.

---

getShortValue() Method

`public short getShortValue()`

Returns the value as a short.

---

getStringValue() Method

`public String getStringValue()`

Returns the value as a `String`.

---

instanceType() Method

`public SchemaType instanceType()`

The same as getSchemaType unless this is a union instance or nil value.

For unions, this returns the non−union consituent type of this instance. This type may change if setters are called that cause the instance to change to another constituent type of the union.

For nil values, this returns null.

---

intValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getIntValue()`

`public int intValue()`

Returns the value as an int. *

---

listValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getListValue()`

`public List listValue()`

Returns the value as a `List` of friendly Java objects (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration). *

---

longValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getLongValue()`

`public long longValue()`

Returns the value as a long. *

---

objectSet(Object) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setObjectValue(java.lang.Object)`

`public void objectSet(Object obj)`

Sets the value as an arbitrary `Object`.

---

objectValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getObjectValue()`

`public Object objectValue()`

Returns a union value as a its natural friendly Java object (String, Integer, Byte, Short, Long, BigInteger, Decimal, Float, Double, byte[], Calendar, GDuration). *

---

qNameValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getQNameValue()`

`public QName qNameValue()`

Returns the value as a `QName`. *

---

set(String) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setStringValue(java.lang.String)`

`public void set(String obj)`

Sets the value as a `String`. *

---

set(boolean) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setBooleanValue(boolean)`

`public void set(boolean v)`

Sets the value as a boolean. *

---

set(byte) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setByteValue(byte)`

`public void set(byte v)`

Sets the value as a byte.

---

set(short) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setShortValue(short)`

`public void set(short v)`

Sets the value as a short.

---

set(int) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setIntValue(int)`

`public void set(int v)`

Sets the value as an int.

---

set(long) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setLongValue(long)`

`public void set(long v)`

Sets the value as a long.

---

set(BigInteger) Method

DEPRECATED replaced with
`com.bea.xml.SimpleValue.setBigIntegerValue(java.math.BigInteger)`

`public void set(BigInteger obj)`

Sets the value as a `BigInteger`.

---

set(BigDecimal) Method

DEPRECATED replaced with
`com.bea.xml.SimpleValue.setBigDecimalValue(java.math.BigDecimal)`

public void set(BigDecimal *obj*)

Sets the value as a `BigDecimal`

---

set(float) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setFloatValue(float)`

public void set(float *v*)

Sets the value as a float.

---

set(double) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setDoubleValue(double)`

public void set(double *v*)

Sets the value as a double.

---

set(byte[]) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setByteArrayValue(byte[])`

public void set(byte[] *obj*)

Sets the value as a byte array.

---

set(StringEnumAbstractBase) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setEnumValue(com.bea.xml.StringEnumAbstractBase)`

public void set(StringEnumAbstractBase *obj*)

Sets the value as a `StringEnumAbstractBase`.

---

set(Calendar) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setCalendarValue(java.util.Calendar)`

public void set(Calendar *obj*)

Sets the value as a `Calendar`.

set(Date) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setDateValue(java.util.Date)`

`public void set(Date obj)`

Sets the value as a `Date`.

set(GDateSpecification) Method

DEPRECATED replaced with
`com.bea.xml.SimpleValue.setGDateValue(com.bea.xml.GDate)`

`public void set(GDateSpecification obj)`

Sets the value as a `GDate`.

set(GDurationSpecification) Method

DEPRECATED replaced with
`com.bea.xml.SimpleValue.setGDurationValue(com.bea.xml.GDuration)`

`public void set(GDurationSpecification obj)`

Sets the value as a `GDuration`.

set(QName) Method

DEPRECATED replaced with
`com.bea.xml.SimpleValue.setQNameValue(javax.xml.namespace.QName)`

`public void set(QName obj)`

Sets the value as a `QName`.

set(List) Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.setListValue(java.util.List)`

`public void set(List obj)`

Sets the value as a `List`.

setBigDecimalValue(BigDecimal) Method

`public void setBigDecimalValue(BigDecimal obj)`

Sets the value as a `BigDecimal`.

---

setBigIntegerValue(BigInteger) Method

`public void setBigIntegerValue(BigInteger obj)`

Sets the value as a `BigInteger`.

---

setBooleanValue(boolean) Method

`public void setBooleanValue(boolean v)`

Sets the value as a boolean.

---

setByteArrayValue(byte[]) Method

`public void setByteArrayValue(byte[] obj)`

Sets the value as a byte array.

---

setByteValue(byte) Method

`public void setByteValue(byte v)`

Sets the value as a byte.

---

setCalendarValue(Calendar) Method

`public void setCalendarValue(Calendar obj)`

Sets the value as a `Calendar`.

---

setDateValue(Date) Method

`public void setDateValue(Date obj)`

Sets the value as a `Date`.

---

setDoubleValue(double) Method

public void setDoubleValue(double *v*)

Sets the value as a double.

---

setEnumValue(StringEnumAbstractBase) Method

public void setEnumValue(StringEnumAbstractBase *obj*)

Sets the value as a `StringEnumAbstractBase`.

---

setFloatValue(float) Method

public void setFloatValue(float *v*)

Sets the value as a float.

---

setGDateValue(GDate) Method

public void setGDateValue(GDate *obj*)

Sets the value as a `GDate`.

---

setGDurationValue(GDuration) Method

public void setGDurationValue(GDuration *obj*)

Sets the value as a `GDuration`.

---

setIntValue(int) Method

public void setIntValue(int *v*)

Sets the value as an int.

---

setListValue(List) Method

public void setListValue(List *obj*)

Sets the value as a `List`.

---

setLongValue(long) Method

`public void setLongValue(long v)`

Sets the value as a long.

---

setObjectValue(Object) Method

`public void setObjectValue(Object obj)`

Sets the value as an arbitrary `Object`.

---

setQNameValue(QName) Method

`public void setQNameValue(QName obj)`

Sets the value as a `QName`.

---

setShortValue(short) Method

`public void setShortValue(short v)`

Sets the value as a short.

---

setStringValue(String) Method

`public void setStringValue(String obj)`

Sets the value as a `String`.

---

shortValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getShortValue()`

`public short shortValue()`

Returns the value as a short. *

---

stringValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getStringValue()`

`public String stringValue()`

Returns the value as a `String`. *

---

xgetListValue() Method

`public List xgetListValue()`

Returns the value as a `List` of XmlAnySimpleType objects.

---

xlistValue() Method

DEPRECATED replaced with `com.bea.xml.SimpleValue.getListValue()`

`public List xlistValue()`

Returns the value as a `List` of XmlAnySimpleType objects. *

com.bea.xml

StringEnumAbstractBase Class

public class ***StringEnumAbstractBase***

extends `Object`
implements `Serializable`

The base class for code−generated string enumeration value classes.

Subclasses are intended to be final types with a finite set of singleton instances. Each instance has a string value, which it returns via `StringEnumAbstractBase.toString()`, and an int value for the purpose of switching in case statements, returned via `StringEnumAbstractBase.intValue()`.

Each subclass manages an instance of `StringEnumAbstractBase.Table`, which holds all the singleton instances for the subclass. A Table can return a singleton instance given a String or an integer code.

Hierarchy

```
Object
 StringEnumAbstractBase
```

All Implemented Interfaces

```
Serializable
```

Nested Class Summary

`public static final class`	**`StringEnumAbstractBase.Table`** Used to manage singleton instances of enumerations.

Constructor Summary

StringEnumAbstractBase(`String` *s, int i)*

Singleton instances should only be created by subclasses.

Method Summary

public final int ***hashCode()***	Returns the hash code of the underlying string
public final int ***intValue()***	Returns an int code that can be used for switch statements
public final String ***toString()***	Returns the underlying string value

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, notify, notifyAll, wait, wait, wait`

Constructor Detail

StringEnumAbstractBase

```
protected StringEnumAbstractBase(String s,
                                 int i)
```

Singleton instances should only be created by subclasses.

Method Detail

hashCode() Method

```
public final int hashCode()
```

Returns the hash code of the underlying string

Overrides

`Object.hashCode()`

---

intValue() Method

```
public final int intValue()
```

Returns an int code that can be used for switch statements

---

toString() Method

`public final String toString()`

Returns the underlying string value

Overrides

`Object.toString()`

StringEnumAbstractBase.Table Class

public static final class ***StringEnumAbstractBase.Table***

extends `Object`

Used to manage singleton instances of enumerations. Each subclass of StringEnumAbstractBase has an instance of a table to hold the singleton instances.

Hierarchy

```
Object
 StringEnumAbstractBase.Table
```

Enclosing class

`StringEnumAbstractBase`

Constructor Summary

StringEnumAbstractBase.Table(`StringEnumAbstractBase` *array)*

Method Summary

`public StringEnumAbstractBase` ***forInt***(*int i)*
Returns the singleton for an int code, or null if none.

`public StringEnumAbstractBase` ***forString***(`String` *s)*
Returns the singleton for a `String`, or null if none.

`public int` ***lastInt***()
Returns the last valid int code (the first is 1; zero is not used).

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

StringEnumAbstractBase.Table

public StringEnumAbstractBase.Table(StringEnumAbstractBase[] *array*)

Method Detail

forInt(int) Method

public StringEnumAbstractBase forInt(int *i*)

Returns the singleton for an int code, or null if none.

---

forString(String) Method

public StringEnumAbstractBase forString(String *s*)

Returns the singleton for a `String`, or null if none.

---

lastInt() Method

public int lastInt()

Returns the last valid int code (the first is 1; zero is not used).

com.bea.xml

XmlAnySimpleType Interface

public interface ***XmlAnySimpleType***

extends `XmlObject`

Corresponds to the XML Schema xs:anySimpleType type.

All simple types are convertible to `String`.

All Superinterfaces

`XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlAnyURI`, `XmlBase64Binary`, `XmlBoolean`, `XmlByte`, `XmlDate`, `XmlDateTime`, `XmlDecimal`, `XmlDouble`, `XmlDuration`, `XmlENTITIES`, `XmlENTITY`, `XmlFloat`, `XmlGDay`, `XmlGMonth`, `XmlGMonthDay`, `XmlGYear`, `XmlGYearMonth`, `XmlHexBinary`, `XmlID`, `XmlIDREF`, `XmlIDREFS`, `XmlInt`, `XmlInteger`, `XmlLanguage`, `XmlLong`, `XmlName`, `XmlNCName`, `XmlNegativeInteger`, `XmlNMTOKEN`, `XmlNMTOKENS`, `XmlNonNegativeInteger`, `XmlNonPositiveInteger`, `XmlNormalizedString`, `XmlNOTATION`, `XmlPositiveInteger`, `XmlQName`, `XmlShort`, `XmlString`, `XmlTime`, `XmlToken`, `XmlUnsignedByte`, `XmlUnsignedInt`, `XmlUnsignedLong`, `XmlUnsignedShort`

Nested Class Summary

`public static final class`	**`XmlAnySimpleType.Factory`** A class with methods for creating instances of `XmlAnySimpleType`.

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public String` ***getStringValue()***
Returns the value as a `String`

`public void` ***set****(String s)*
Sets the value as a `String`

`public void` ***setStringValue****(String s)*
Sets the value as a `String`

`public String` ***stringValue()***
Returns the value as a `String`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getStringValue() Method

`public String getStringValue()`

Returns the value as a `String`

set(String) Method

DEPRECATED replaced by `com.bea.xml.XmlAnySimpleType.setStringValue(java.lang.String)`

`public void set(String s)`

Sets the value as a `String`

setStringValue(String) Method

`public void setStringValue(String s)`

Sets the value as a `String`

stringValue() Method

DEPRECATED replaced by `com.bea.xml.XmlAnySimpleType.getStringValue()`

`public String stringValue()`

Returns the value as a `String`

XmlAnySimpleType.Factory Class

public static final class ***XmlAnySimpleType.Factory***

extends `Object`

A class with methods for creating instances of `XmlAnySimpleType`.

Hierarchy

```
Object
 XmlAnySimpleType.Factory
```

Enclosing interface

`XmlAnySimpleType`

Method Summary

`public static XmlAnySimpleType` ***newInstance***()
Creates an empty instance of `XmlAnySimpleType`

`public static XmlAnySimpleType` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlAnySimpleType`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlAnySimpleType` ***newValue***(`Object` *obj*)
Creates an immutable `XmlAnySimpleType` value

`public static XmlAnySimpleType` ***parse***(`String` *s*)
Parses a `XmlAnySimpleType` fragment from a String.

`public static XmlAnySimpleType` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlAnySimpleType` fragment from a String.

`public static XmlAnySimpleType` ***parse***(`File` *f*)
Parses a `XmlAnySimpleType` fragment from a File.

public static XmlAnySimpleType	***parse***(*File f, XmlOptions options*) Parses a XmlAnySimpleType fragment from a File.
public static XmlAnySimpleType	***parse***(*URL u*) Parses a XmlAnySimpleType fragment from a URL.
public static XmlAnySimpleType	***parse***(*URL u, XmlOptions options*) Parses a XmlAnySimpleType fragment from a URL.
public static XmlAnySimpleType	***parse***(*InputStream is*) Parses a XmlAnySimpleType fragment from an InputStream.
public static XmlAnySimpleType	***parse***(*InputStream is, XmlOptions options*) Parses a XmlAnySimpleType fragment from an InputStream.
public static XmlAnySimpleType	***parse***(*Reader r*) Parses a XmlAnySimpleType fragment from a Reader.
public static XmlAnySimpleType	***parse***(*Reader r, XmlOptions options*) Parses a XmlAnySimpleType fragment from a Reader.
public static XmlAnySimpleType	***parse***(*Node node*) Parses a XmlAnySimpleType fragment from a DOM Node.
public static XmlAnySimpleType	***parse***(*Node node, XmlOptions options*) Parses a XmlAnySimpleType fragment from a DOM Node.
public static XmlAnySimpleType	***parse***(*XMLInputStream xis*) Parses a XmlAnySimpleType fragment from an XMLInputStream.
public static XmlAnySimpleType	***parse***(*XMLInputStream xis, XmlOptions options*) Parses a XmlAnySimpleType fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlAnySimpleType newInstance()`

Creates an empty instance of XmlAnySimpleType

---

newInstance(XmlOptions) Method

```
public static XmlAnySimpleType newInstance(XmlOptions options)
```

Creates an empty instance of `XmlAnySimpleType`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlAnySimpleType newValue(Object obj)
```

Creates an immutable `XmlAnySimpleType` value

---

parse(String) Method

```
public static XmlAnySimpleType parse(String s)
     throws XmlException
```

Parses a `XmlAnySimpleType` fragment from a String. For example: "`<xml-fragment>anything</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlAnySimpleType parse(String s,
                                     XmlOptions options)
     throws XmlException
```

Parses a `XmlAnySimpleType` fragment from a String. For example: "`<xml-fragment>anything</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlAnySimpleType parse(File f)
     throws XmlException, IOException
```

Parses a `XmlAnySimpleType` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlAnySimpleType parse(File f,
                                     XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlAnySimpleType` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlAnySimpleType parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlAnySimpleType` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlAnySimpleType parse(URL u,
                                     XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlAnySimpleType` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlAnySimpleType parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlAnySimpleType` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlAnySimpleType parse(InputStream is,
                                     XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlAnySimpleType` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlAnySimpleType parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlAnySimpleType` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlAnySimpleType parse(Reader r,
                                     XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlAnySimpleType` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlAnySimpleType parse(Node node)
     throws XmlException
```

Parses a `XmlAnySimpleType` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlAnySimpleType parse(Node node,
                                     XmlOptions options)
     throws XmlException
```

Parses a `XmlAnySimpleType` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlAnySimpleType parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlAnySimpleType` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlAnySimpleType parse(XMLInputStream xis,
                                     XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlAnySimpleType` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlAnyURI Interface

public interface ***XmlAnyURI***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:anyURI type.

Convertible to `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlAnyURI.Factory`** A class with methods for creating instances of `XmlAnyURI`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlAnyURI.Factory Class

public static final class ***XmlAnyURI.Factory***

extends `Object`

A class with methods for creating instances of `XmlAnyURI`.

Hierarchy

```
Object
 XmlAnyURI.Factory
```

Enclosing interface

`XmlAnyURI`

Method Summary

`public static XmlAnyURI` ***newInstance***()
Creates an empty instance of `XmlAnyURI`

`public static XmlAnyURI` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlAnyURI`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlAnyURI` ***newValue***(`Object` *obj*)
Creates an immutable `XmlAnyURI` value

`public static XmlAnyURI` ***parse***(`String` *s*)
Parses a `XmlAnyURI` fragment from a String.

`public static XmlAnyURI` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlAnyURI` fragment from a String.

`public static XmlAnyURI` ***parse***(`File` *f*)
Parses a `XmlAnyURI` fragment from a File.

`public static XmlAnyURI` ***parse***(*File f, XmlOptions options*)
Parses a `XmlAnyURI` fragment from a File.

`public static XmlAnyURI` ***parse***(*URL u*)
Parses a `XmlAnyURI` fragment from a URL.

`public static XmlAnyURI` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlAnyURI` fragment from a URL.

`public static XmlAnyURI` ***parse***(*InputStream is*)
Parses a `XmlAnyURI` fragment from an InputStream.

`public static XmlAnyURI` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlAnyURI` fragment from an InputStream.

`public static XmlAnyURI` ***parse***(*Reader r*)
Parses a `XmlAnyURI` fragment from a Reader.

`public static XmlAnyURI` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlAnyURI` fragment from a Reader.

`public static XmlAnyURI` ***parse***(*Node node*)
Parses a `XmlAnyURI` fragment from a DOM Node.

`public static XmlAnyURI` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlAnyURI` fragment from a DOM Node.

`public static XmlAnyURI` ***parse***(*XMLInputStream xis*)
Parses a `XmlAnyURI` fragment from an XMLInputStream.

`public static XmlAnyURI` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlAnyURI` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlAnyURI newInstance()`

Creates an empty instance of `XmlAnyURI`

---

newInstance(XmlOptions) Method

```
public static XmlAnyURI newInstance(XmlOptions options)
```

Creates an empty instance of `XmlAnyURI`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlAnyURI newValue(Object obj)
```

Creates an immutable `XmlAnyURI` value

---

parse(String) Method

```
public static XmlAnyURI parse(String s)
     throws XmlException
```

Parses a `XmlAnyURI` fragment from a String. For example: "`<xml-fragment>http://www.w3.org/</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlAnyURI parse(String s,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlAnyURI` fragment from a String. For example: "`<xml-fragment>http://www.w3.org/</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlAnyURI parse(File f)
     throws XmlException, IOException
```

Parses a `XmlAnyURI` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlAnyURI parse(File f,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlAnyURI` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlAnyURI parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlAnyURI` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlAnyURI parse(URL u,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlAnyURI` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlAnyURI parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlAnyURI` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlAnyURI parse(InputStream is,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlAnyURI` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlAnyURI parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlAnyURI` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlAnyURI parse(Reader r,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlAnyURI` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlAnyURI parse(Node node)
     throws XmlException
```

Parses a `XmlAnyURI` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlAnyURI parse(Node node,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlAnyURI` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlAnyURI parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlAnyURI` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlAnyURI parse(XMLInputStream xis,
                              XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlAnyURI` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlBase64Binary Interface

public interface ***XmlBase64Binary***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:base64Binary type.

Convertible to a byte array.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlBase64Binary.Factory`** A class with methods for creating instances of `XmlBase64Binary`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

public byte[] ***byteArrayValue()***
Returns this value as a byte array.

public byte[] ***getByteArrayValue()***
Returns this value as a byte array.

public void ***set****(byte[] ba)*
Sets this value as a byte array.

public void ***setByteArrayValue****(byte[] ba)*
Sets this value as a byte array.

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

byteArrayValue() Method

DEPRECATED replaced by `com.bea.xml.XmlBase64Binary.getByteArrayValue()`

`public byte[] byteArrayValue()`

Returns this value as a byte array.

---

getByteArrayValue() Method

`public byte[] getByteArrayValue()`

Returns this value as a byte array.

---

set(byte[]) Method

DEPRECATED replaced by `com.bea.xml.XmlBase64Binary.setByteArrayValue(byte[])`

`public void set(byte[] `*`ba`*`)`

Sets this value as a byte array.

---

setByteArrayValue(byte[]) Method

`public void setByteArrayValue(byte[] `*`ba`*`)`

Sets this value as a byte array.

XmlBase64Binary.Factory Class

public static final class ***XmlBase64Binary.Factory***

extends `Object`

A class with methods for creating instances of `XmlBase64Binary`.

Hierarchy

```
Object
 XmlBase64Binary.Factory
```

Enclosing interface

`XmlBase64Binary`

Method Summary

public static XmlBase64Binary	***newInstance***() Creates an empty instance of `XmlBase64Binary`
public static XmlBase64Binary	***newInstance***(`XmlOptions` *options*) Creates an empty instance of `XmlBase64Binary`
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*) Returns a validating XMLInputStream.
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*) Returns a validating XMLInputStream.
public static XmlBase64Binary	***newValue***(`Object` *obj*) Creates an immutable `XmlBase64Binary` value
public static XmlBase64Binary	***parse***(`String` *s*) Parses a `XmlBase64Binary` fragment from a String.
public static XmlBase64Binary	***parse***(`String` *s*, `XmlOptions` *options*) Parses a `XmlBase64Binary` fragment from a String.
public static XmlBase64Binary	***parse***(`File` *f*) Parses a `XmlBase64Binary` fragment from a File.

public static XmlBase64Binary **parse**(File f, XmlOptions options)
Parses a XmlBase64Binary fragment from a File.

public static XmlBase64Binary **parse**(URL u)
Parses a XmlBase64Binary fragment from a URL.

public static XmlBase64Binary **parse**(URL u, XmlOptions options)
Parses a XmlBase64Binary fragment from a URL.

public static XmlBase64Binary **parse**(InputStream is)
Parses a XmlBase64Binary fragment from an InputStream.

public static XmlBase64Binary **parse**(InputStream is, XmlOptions options)
Parses a XmlBase64Binary fragment from an InputStream.

public static XmlBase64Binary **parse**(Reader r)
Parses a XmlBase64Binary fragment from a Reader.

public static XmlBase64Binary **parse**(Reader r, XmlOptions options)
Parses a XmlBase64Binary fragment from a Reader.

public static XmlBase64Binary **parse**(Node node)
Parses a XmlBase64Binary fragment from a DOM Node.

public static XmlBase64Binary **parse**(Node node, XmlOptions options)
Parses a XmlBase64Binary fragment from a DOM Node.

public static XmlBase64Binary **parse**(XMLInputStream xis)
Parses a XmlBase64Binary fragment from an XMLInputStream.

public static XmlBase64Binary **parse**(XMLInputStream xis, XmlOptions options)
Parses a XmlBase64Binary fragment from an XMLInputStream.

Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

newInstance() Method

public static XmlBase64Binary newInstance()

Creates an empty instance of XmlBase64Binary

---

newInstance(XmlOptions) Method

```
public static XmlBase64Binary newInstance(XmlOptions options)
```

Creates an empty instance of `XmlBase64Binary`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlBase64Binary newValue(Object obj)
```

Creates an immutable `XmlBase64Binary` value

---

parse(String) Method

```
public static XmlBase64Binary parse(String s)
     throws XmlException
```

Parses a `XmlBase64Binary` fragment from a String. For example: "`<xml-fragment>VGhpcyBzdHJp</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlBase64Binary parse(String s,
                                    XmlOptions options)
     throws XmlException
```

Parses a `XmlBase64Binary` fragment from a String. For example: "`<xml-fragment>VGhpcyBzdHJp</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlBase64Binary parse(File f)
     throws XmlException, IOException
```

Parses a `XmlBase64Binary` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlBase64Binary parse(File f,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlBase64Binary` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlBase64Binary parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlBase64Binary` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlBase64Binary parse(URL u,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlBase64Binary` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlBase64Binary parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlBase64Binary` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlBase64Binary parse(InputStream is,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlBase64Binary` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlBase64Binary parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlBase64Binary` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlBase64Binary parse(Reader r,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlBase64Binary` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlBase64Binary parse(Node node)
     throws XmlException
```

Parses a `XmlBase64Binary` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlBase64Binary parse(Node node,
                                    XmlOptions options)
     throws XmlException
```

Parses a `XmlBase64Binary` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlBase64Binary parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlBase64Binary` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlBase64Binary parse(XMLInputStream xis,
                                    XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlBase64Binary` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlBeans Class

public final class ***XmlBeans***

extends `Object`

Provides an assortment of utilities for managing XML Bean types, type systems, QNames, paths, and queries.

Hierarchy

```
Object
 XmlBeans
```

Field Summary

`public static SchemaType` ***NO_TYPE***

The SchemaType object given to an XmlObject instance when no type can be determined.

Method Summary

`public static String` ***compilePath***(`String` *pathExpr*)

Compiles an XPath, returning a String equal to that which was passed, but whose identity is that of one which has been precompiled and cached.

`public static String` ***compilePath***(`String` *pathExpr*, `XmlOptions` *options*)

Compiles an XPath, returning a String equal to that which was passed, but whose identity is that of one which has been precompiled and cached; takes an option for specifying text that indicates the name of context node.

`public static String` ***compileQuery***(`String` *queryExpr*)

Compiles an XQuery, returning a String equal to that which was passed, but whose identity is that of one which has been precompiled and cached.

`public static String` ***compileQuery***(`String` *queryExpr*, `XmlOptions` *options*)

Compiles an XQuery, returning a String equal to that which was passed, but whose identity is that of one which has been precompiled and cached; takes an option for specifying text that indicates the context node.

`public static SchemaTypeSystem` ***compileXsd***(`XmlObject[]` *schemas*, `SchemaTypeLoader` *typepath*, `XmlOptions` *options*)

Returns the SchemaTypeSystem that results from compiling the XML schema definitions passed.

`public static SchemaTypeSystem` ***getBuiltinTypeSystem***()

Returns the builtin type system.

`public static SchemaTypeLoader` ***getContextTypeLoader***()

Gets the SchemaTypeLoader based on the current thread's context ClassLoader.

`public static QName` ***getQName***(`String` *localPart*)

Obtains a name from the thread local QNameCache

`public static QName` ***getQName***(`String` *namespaceUri*, `String` *localPart*)

Obtains a name from the thread local QNameCache

`public static QNameCache` ***getQNameCache***()

Returns a thread local QNameCache

`public static SchemaTypeLoader` ***loadXsd***(`XmlObject[]` *schemas*)

Returns the SchemaTypeSystem that results from compiling the XML schema definitions passed.

`public static SchemaTypeLoader` ***loadXsd***(`XmlObject[]` *schemas*, `XmlOptions` *options*)

Returns the SchemaTypeSystem that results from compiling the XML schema definitions passed in *schemas*.

`public static SchemaType` ***typeForClass***(`Class` *c*)

Returns the SchemaType from a corresponding XmlObject subclass, or null if none.

`public static SchemaTypeLoader` ***typeLoaderForClassLoader***(`ClassLoader` *loader*)

Returns a SchemaTypeLoader that searches for compiled schema types in the given ClassLoader.

`public static SchemaTypeLoader` ***typeLoaderUnion***(`SchemaTypeLoader[]` *typeLoaders*)

Returns the union of a list of typeLoaders.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

NO_TYPE

```
public static SchemaType NO_TYPE
```

The SchemaType object given to an XmlObject instance when no type can be determined.

The NO_TYPE is the universal derived type. That is, it is derived from all other schema types, and no instances of the NO_TYPE are valid. (It is not to be confused with the anyType, which is the universal base type from which all other types can be derived, and of which all instances are valid.)

Method Detail

compilePath(String) Method

```
public static String compilePath(String pathExpr)
     throws XmlException
```

Compiles an XPath, returning a String equal to that which was passed, but whose identity is that of one which has been precompiled and cached.

Exceptions

XmlException

---

compilePath(String, XmlOptions) Method

```
public static String compilePath(String pathExpr,
                                 XmlOptions options)
     throws XmlException
```

Compiles an XPath, returning a String equal to that which was passed, but whose identity is that of one which has been precompiled and cached; takes an option for specifying text that indicates the name of context node. The default is "this", as in "$this".

Parameters

pathExpr
: Options for the path. For example, you can call the `XmlOptions.setXqueryCurrentNodeVar(String)` method to specify a particular name for the expression variable that indicates the context node.

Exceptions

XmlException

---

compileQuery(String) Method

```
public static String compileQuery(String queryExpr)
     throws XmlException
```

Compiles an XQuery, returning a String equal to that which was passed, but whose identity is that of one which has been precompiled and cached.

Exceptions

XmlException

---

compileQuery(String, XmlOptions) Method

```
public static String compileQuery(String queryExpr,
                                  XmlOptions options)
     throws XmlException
```

Compiles an XQuery, returning a String equal to that which was passed, but whose identity is that of one which has been precompiled and cached; takes an option for specifying text that indicates the context node.

Parameters

queryExpr

Options for the query. For example, you can call the `XmlOptions.setXqueryCurrentNodeVar(String)` method to specify a particular name for the expression variable that indicates the context node and the `XmlOptions.setXqueryVariables(Map)` method to map external variable names to values.

Exceptions

XmlException

---

compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions) Method

```
public static SchemaTypeSystem compileXsd(XmlObject[] schemas,
                                          SchemaTypeLoader typepath,
                                          XmlOptions options)
     throws XmlException
```

Returns the SchemaTypeSystem that results from compiling the XML schema definitions passed.

The XmlObjects passed in should be w3c <schema> elements whose type is org.w3c.x2001.xmlSchema.Schema. (That is, schema elements in the XML namespace http://www.w3c.org/2001/XMLSchema.) Also org.w3c.x2001.xmlSchema.SchemaDocument is permitted.

The optional second argument is a SchemaTypeLoader which will be consulted for already−compiled schema types which may be linked while processing the given schemas.

The SchemaTypeSystem that is returned should be combined (via `XmlBeans.typeLoaderUnion(SchemaTypeLoader[])`) with the typepath typeloader in order to create a typeloader that can be used for creating and validating instances.

Use the *options* parameter to specify the following:

- A collection instance that should be used as an error listener during compilation, as described in `XmlOptions.setErrorListener(Collection)`.
- Whether validation should not be done when building the SchemaTypeSystem, as described in `XmlOptions.setCompileNoValidation()`.

Parameters

schemas
The schema definitions from which to build the schema type system.

typepath
The path to already−compiled schema types for linking while processing.

options
Options specifying an error listener and/or validation behavior.

Exceptions

`XmlException`

---

getBuiltinTypeSystem() Method

`public static SchemaTypeSystem getBuiltinTypeSystem()`

Returns the builtin type system. This SchemaTypeSystem contains only the 46 builtin types defined by the XML Schema specification.

---

getContextTypeLoader() Method

`public static SchemaTypeLoader getContextTypeLoader()`

Gets the SchemaTypeLoader based on the current thread's context ClassLoader. This is the SchemaTypeLoader that is used to assign schema types to XML documents by default. The SchemaTypeLoader is also consulted to resolve wildcards and xsi:type attributes.

The "parse" methods of XmlBeans all delegate to the "parseInstance" methods of the context type loader.

---

getQName(String) Method

`public static QName getQName(String localPart)`

Obtains a name from the thread local QNameCache

---

getQName(String, String) Method

```
public static QName getQName(String namespaceUri,
                             String localPart)
```

Obtains a name from the thread local QNameCache

---

getQNameCache() Method

```
public static QNameCache getQNameCache()
```

Returns a thread local QNameCache

---

loadXsd(XmlObject[]) Method

```
public static SchemaTypeLoader loadXsd(XmlObject[] schemas)
     throws XmlException
```

Returns the SchemaTypeSystem that results from compiling the XML schema definitions passed.

Just like compileTypeSystem, but uses the context type loader for linking, and returns a unioned typeloader that is suitable for creating instances.

Exceptions

XmlException

---

loadXsd(XmlObject[], XmlOptions) Method

```
public static SchemaTypeLoader loadXsd(XmlObject[] schemas,
                                       XmlOptions options)
     throws XmlException
```

Returns the SchemaTypeSystem that results from compiling the XML schema definitions passed in *schemas*.

This is just like compileTypeSystem, but uses the context type loader for linking, and returns a unioned typeloader that is suitable for creating instances.

Use the *options* parameter to specify one or both of the following:

- A collection instance that should be used as an error listener during compilation, as described in `XmlOptions.setErrorListener(Collection)`.
- Whether validation should not be done when building the SchemaTypeSystem, as described in `XmlOptions.setCompileNoValidation()`.

Parameters

schemas
The schema definitions from which to build the schema type system.

options
Options specifying an error listener and/or validation behavior.

Exceptions

XmlException

---

typeForClass(Class) Method

public static SchemaType typeForClass(Class *c*)

Returns the SchemaType from a corresponding XmlObject subclass, or null if none.

---

typeLoaderForClassLoader(ClassLoader) Method

public static SchemaTypeLoader typeLoaderForClassLoader(ClassLoader *loader*)

Returns a SchemaTypeLoader that searches for compiled schema types in the given ClassLoader.

---

typeLoaderUnion(SchemaTypeLoader[]) Method

public static SchemaTypeLoader typeLoaderUnion(SchemaTypeLoader[] *typeLoaders*)

Returns the union of a list of typeLoaders. The returned SchemaTypeLoader searches the given list of SchemaTypeLoaders in order from first to last.

com.bea.xml

XmlBoolean Interface

public interface ***XmlBoolean***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:boolean type.

Naturally, convertible to Java boolean.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlBoolean.Factory`** A class with methods for creating instances of `XmlBoolean`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

public boolean ***booleanValue()***
Returns this value as a boolean

public boolean ***getBooleanValue()***
Returns this value as a boolean

public void ***set****(boolean v)*
Sets this value as a boolean

public void ***setBooleanValue****(boolean v)*
Sets this value as a boolean

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

booleanValue() Method

DEPRECATED replaced by `com.bea.xml.XmlBoolean.getBooleanValue()`

`public boolean booleanValue()`

Returns this value as a boolean

---

getBooleanValue() Method

`public boolean getBooleanValue()`

Returns this value as a boolean

---

set(boolean) Method

DEPRECATED replaced by `com.bea.xml.XmlBoolean.setBooleanValue(boolean)`

`public void set(boolean v)`

Sets this value as a boolean

---

setBooleanValue(boolean) Method

`public void setBooleanValue(boolean v)`

Sets this value as a boolean

XmlBoolean.Factory Class

public static final class ***XmlBoolean.Factory***

extends `Object`

A class with methods for creating instances of `XmlBoolean`.

Hierarchy

```
Object
 XmlBoolean.Factory
```

Enclosing interface

`XmlBoolean`

Method Summary

`public static XmlBoolean` ***newInstance***()
Creates an empty instance of `XmlBoolean`

`public static XmlBoolean` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlBoolean`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlBoolean` ***newValue***(`Object` *obj*)
Creates an immutable `XmlBoolean` value

`public static XmlBoolean` ***parse***(`String` *s*)
Parses a `XmlBoolean` fragment from a String.

`public static XmlBoolean` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlBoolean` fragment from a String.

`public static XmlBoolean` ***parse***(`File` *f*)
Parses a `XmlBoolean` fragment from a File.

`public static XmlBoolean` ***parse***(*File f, XmlOptions options*)
Parses a `XmlBoolean` fragment from a File.

`public static XmlBoolean` ***parse***(*URL u*)
Parses a `XmlBoolean` fragment from a URL.

`public static XmlBoolean` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlBoolean` fragment from a URL.

`public static XmlBoolean` ***parse***(*InputStream is*)
Parses a `XmlBoolean` fragment from an InputStream.

`public static XmlBoolean` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlBoolean` fragment from an InputStream.

`public static XmlBoolean` ***parse***(*Reader r*)
Parses a `XmlBoolean` fragment from a Reader.

`public static XmlBoolean` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlBoolean` fragment from a Reader.

`public static XmlBoolean` ***parse***(*Node node*)
Parses a `XmlBoolean` fragment from a DOM Node.

`public static XmlBoolean` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlBoolean` fragment from a DOM Node.

`public static XmlBoolean` ***parse***(*XMLInputStream xis*)
Parses a `XmlBoolean` fragment from an XMLInputStream.

`public static XmlBoolean` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlBoolean` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlBoolean newInstance()`

Creates an empty instance of `XmlBoolean`

---

newInstance(XmlOptions) Method

```
public static XmlBoolean newInstance(XmlOptions options)
```

Creates an empty instance of `XmlBoolean`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlBoolean newValue(Object obj)
```

Creates an immutable `XmlBoolean` value

---

parse(String) Method

```
public static XmlBoolean parse(String s)
     throws XmlException
```

Parses a `XmlBoolean` fragment from a String. For example: "`<xml-fragment>true</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlBoolean parse(String s,
                               XmlOptions options)
     throws XmlException
```

Parses a `XmlBoolean` fragment from a String. For example: "`<xml-fragment>true</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlBoolean parse(File f)
     throws XmlException, IOException
```

Parses a `XmlBoolean` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlBoolean parse(File f,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlBoolean` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlBoolean parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlBoolean` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlBoolean parse(URL u,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlBoolean` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlBoolean parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlBoolean` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlBoolean parse(InputStream is,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlBoolean` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlBoolean parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlBoolean` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlBoolean parse(Reader r,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlBoolean` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlBoolean parse(Node node)
     throws XmlException
```

Parses a `XmlBoolean` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlBoolean parse(Node node,
                               XmlOptions options)
     throws XmlException
```

Parses a `XmlBoolean` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlBoolean parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlBoolean` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlBoolean parse(XMLInputStream xis,
                               XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlBoolean` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlByte Interface

public interface ***XmlByte***

extends `XmlShort`

Corresponds to the XML Schema xs:byte type.

Naturally, convertible to Java byte.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInt`, `XmlInteger`, `XmlLong`, `XmlObject`, `XmlShort`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlByte.Factory`** A class with methods for creating instances of `XmlByte`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInt`

`XmlInt.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlLong`

`XmlLong.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlShort`

`XmlShort.Factory`

Field Summary

`public static final SchemaType` ***`type`***
The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInt`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlLong`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlShort`

`type`

Method Summary

`public byte` ***`byteValue`****()*
Returns this value as a byte

`public byte` ***`getByteValue`****()*
Returns this value as a byte

`public void` ***set***(*byte s*)
Sets this value as a byte

`public void` ***setByteValue***(*byte s*)
Sets this value as a byte

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInt`

`getIntValue, intValue, set, setIntValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlLong`

`getLongValue, longValue, set, setLongValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlShort`

`getShortValue, set, setShortValue, shortValue`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

byteValue() Method

DEPRECATED replaced by `com.bea.xml.XmlByte.getByteValue()`

`public byte byteValue()`

Returns this value as a byte

---

getByteValue() Method

`public byte getByteValue()`

Returns this value as a byte

---

set(byte) Method

DEPRECATED replaced by `com.bea.xml.XmlByte.setByteValue(byte)`

`public void set(byte` *`s`***`)`**

Sets this value as a byte

---

setByteValue(byte) Method

`public void setByteValue(byte` *`s`***`)`**

Sets this value as a byte

com.bea.xml

XmlByte.Factory Class

public static final class ***XmlByte.Factory***

extends `Object`

A class with methods for creating instances of `XmlByte`.

Hierarchy

```
Object
 XmlByte.Factory
```

Enclosing interface

`XmlByte`

Method Summary

`public static XmlByte` ***newInstance***()
Creates an empty instance of `XmlByte`

`public static XmlByte` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlByte`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlByte` ***newValue***(`Object` *obj*)
Creates an immutable `XmlByte` value

`public static XmlByte` ***parse***(`String` *s*)
Parses a `XmlByte` fragment from a String.

`public static XmlByte` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlByte` fragment from a String.

`public static XmlByte` ***parse***(`File` *f*)
Parses a `XmlByte` fragment from a File.

`public static XmlByte` ***parse***(*File f, XmlOptions options*)
Parses a `XmlByte` fragment from a File.

`public static XmlByte` ***parse***(*URL u*)
Parses a `XmlByte` fragment from a URL.

`public static XmlByte` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlByte` fragment from a URL.

`public static XmlByte` ***parse***(*InputStream is*)
Parses a `XmlByte` fragment from an InputStream.

`public static XmlByte` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlByte` fragment from an InputStream.

`public static XmlByte` ***parse***(*Reader r*)
Parses a `XmlByte` fragment from a Reader.

`public static XmlByte` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlByte` fragment from a Reader.

`public static XmlByte` ***parse***(*Node node*)
Parses a `XmlByte` fragment from a DOM Node.

`public static XmlByte` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlByte` fragment from a DOM Node.

`public static XmlByte` ***parse***(*XMLInputStream xis*)
Parses a `XmlByte` fragment from an XMLInputStream.

`public static XmlByte` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlByte` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlByte newInstance()`

Creates an empty instance of `XmlByte`

---

newInstance(XmlOptions) Method

```
public static XmlByte newInstance(XmlOptions options)
```

Creates an empty instance of `XmlByte`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlByte newValue(Object obj)
```

Creates an immutable `XmlByte` value

---

parse(String) Method

```
public static XmlByte parse(String s)
     throws XmlException
```

Parses a `XmlByte` fragment from a String. For example: "`<xml-fragment>123</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlByte parse(String s,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlByte` fragment from a String. For example: "`<xml-fragment>123</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlByte parse(File f)
     throws XmlException, IOException
```

Parses a `XmlByte` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlByte parse(File f,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlByte` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlByte parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlByte` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlByte parse(URL u,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlByte` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlByte parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlByte` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlByte parse(InputStream is,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlByte` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlByte parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlByte` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlByte parse(Reader r,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlByte` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlByte parse(Node node)
     throws XmlException
```

Parses a `XmlByte` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlByte parse(Node node,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlByte` fragment from a DOM Node.

Exceptions

XmlException

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlByte parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlByte` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlByte parse(XMLInputStream xis,
                            XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlByte` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlCalendar Class

public class ***XmlCalendar***

extends `GregorianCalendar`

An XML Schema compatible subclass of `GregorianCalendar`. XmlCalendar modifies several key details in the behavior of GregorianCalendar to make it more useful when dealing with XML dates.

It is easy to convert between XmlCalendar and `GDate`, or to parse or emit an XmlCalendar using a standard XML Schema lexical representation.

1. To match XML Schema dates, this XmlCalendar is a fully proleptic Gregorian calendar by default, which means that Gregorian calendar rules are applied backwards in time as if they had always been in effect, actual historical circumstances concerning the observance of the 1582 decree of Pope Gregory XIII notwithstanding.
2. In order to better support partially−specified dates for XML Schema, this implementation provides a stable get(field) method that does not modify the instance if you are acessing a field right after it was explicitly set: a set followed by a get will always return the same thing and will not fill in any other fields. However, if you get a field that was not explicitly set, then all the fields are still automatically filled and normalized for you, just like a regular GregorianCalendar. If you wish to force the completion and defaulting of all the fields (without hunting to get one that happens to be unset), you can always do so by calling getTime().
3. When a year is unspecified and needs to be filled in automatically (for example when using a .get or .getTime method as discussed above), the year is defaulted to year 0 (also known as 1 BC). This is different from `GregorianCalendar`, which chooses 1970. The reason 0 is preferable is that it is a leap year and so it permits the date −−2−29 to be specified stably. A different default year can be chosen via the static method `XmlCalendar.setDefaultYear()`, or by setting the system property "user.defaultyear". If you do change this value, you should pick another leap year such as 2000 and avoid non−leap years such as 1900.
4. When constructing an XmlCalendar from an XML Schema formatted date or time string or GDate object, the timezone for the calendar is taken from the string if it is present, or taken to be `TimeZone.getDefault()` if not.

   For example, the XML timezone "Z" is translated to "GMT"; the XML timezone "+05:00" is translated to "GMT+05:00".
5. Finally, this implementation provides a String constructor and a toString() method that comply with the XML Schema conventions for formatting a date. If only a subset of fields have been explicitly set, toString() produces a string with the proper subset of information.

Hierarchy

```
Object
 Calendar
  GregorianCalendar
   XmlCalendar
```

All Implemented Interfaces

`Cloneable`, `Serializable`

Field Summary

Fields from java.util.`GregorianCalendar`

`AD, BC`

Fields from java.util.`Calendar`

`AM, AM_PM, APRIL, areFieldsSet, AUGUST, DATE, DAY_OF_MONTH, DAY_OF_WEEK, DAY_OF_WEEK_IN_MONTH, DAY_OF_YEAR, DECEMBER, DST_OFFSET, ERA, FEBRUARY, FIELD_COUNT, fields, FRIDAY, HOUR, HOUR_OF_DAY, isSet, isTimeSet, JANUARY, JULY, JUNE, MARCH, MAY, MILLISECOND, MINUTE, MONDAY, MONTH, NOVEMBER, OCTOBER, PM, SATURDAY, SECOND, SEPTEMBER, SUNDAY, THURSDAY, time, TUESDAY, UNDECIMBER, WEDNESDAY, WEEK_OF_MONTH, WEEK_OF_YEAR, YEAR, ZONE_OFFSET`

Constructor Summary

XmlCalendar()

Constructs an empty instance with no fields set.

XmlCalendar(`GDateSpecification` date)

Constructs an XmlCalendar from a GDate.

XmlCalendar(int year, int month, int day, int hour, int minute, int second, `BigDecimal` fraction, int tzSign, int tzHour, int tzMinute)

Constructs an XmlCalendar with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, in the specified timezone.

XmlCalendar(int year, int month, int day, int hour, int minute, int second, `BigDecimal` fraction)

Constructs an XmlCalendar with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, in the default timezone.

XmlCalendar(`String` xmlSchemaDateString)

Constructs an XmlCalendar for a standard XML schema formatted date string.

XmlCalendar(`Date` *date)*

Constructs an XmlCalendar from a Date.

Method Summary

`protected void` ***computeTime***()
Overrides GregorianCalendar.computeTime to apply a different default year.

`public int` ***get***(*int field)*
Gets the value for a given time field.

`public static int` ***getDefaultYear***()
Returns the default year that is used when no year is specified.

`public static void` ***setDefaultYear***(*int year)*
Sets the default year to be used when no year is specified.

`public String` ***toString***()
Prints the XmlCalendar using a standard XML Schema format, as described in XmlCalendar(String s).

Methods from `java.util.GregorianCalendar`

`add, computeFields, equals, getActualMaximum, getActualMinimum, getGreatestMinimum, getGregorianChange, getLeastMaximum, getMaximum, getMinimum, hashCode, isLeapYear, roll, roll, setGregorianChange`

Methods from `java.util.Calendar`

`add, after, before, clear, clear, clone, complete, computeFields, computeTime, equals, getActualMaximum, getActualMinimum, getAvailableLocales, getFirstDayOfWeek, getGreatestMinimum, getInstance, getInstance, getInstance, getInstance, getLeastMaximum, getMaximum, getMinimalDaysInFirstWeek, getMinimum, getTime, getTimeInMillis, getTimeZone, hashCode, internalGet, isLenient, isSet, roll, roll, set, set, set, set, setFirstDayOfWeek, setLenient, setMinimalDaysInFirstWeek, setTime, setTimeInMillis, setTimeZone,`

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

XmlCalendar

```
public XmlCalendar()
```

Constructs an empty instance with no fields set.

XmlCalendar

```
public XmlCalendar(GDateSpecification date)
```

Constructs an XmlCalendar from a GDate. If the instance is not completed, you can round−trip to an equivalent GDate by writing "new GDate(new XmlCalendar(gdate))". However, if you access any of the unset fields of the calendar, all the fields will be automatically filled in, so partial dates without timezones or other fields will not round−trip after access.

XmlCalendar

```
public XmlCalendar(int year,
                   int month,
                   int day,
                   int hour,
                   int minute,
                   int second,
                   BigDecimal fraction,
                   int tzSign,
                   int tzHour,
                   int tzMinute)
```

Constructs an XmlCalendar with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, in the specified timezone.

XmlCalendar

```
public XmlCalendar(int year,
                   int month,
                   int day,
                   int hour,
                   int minute,
                   int second,
                   BigDecimal fraction)
```

Constructs an XmlCalendar with the specified year, month, day, hours, minutes, seconds, and optional fractional seconds, in the default timezone.

XmlCalendar

public XmlCalendar(String *xmlSchemaDateString***)**

Constructs an XmlCalendar for a standard XML schema formatted date string. The parser accepts any of the following formats: YYYY−MM−DDThh:mm:ss − dateTime YYYY−MM−DD − date hh:mm:ss − time YYYY − gYear −−MM − gMonth −−−DD − gDay The parser actually accepts all 16 combinations of subsets of fields (Y, M, D, T) using the same scheme, even for combinations that are not defined as types in the schema spec, such as year, day, and time: YYYY−−DDThh:mm:ss − [everything but month specified] In the string, each field must be padded to its full width, for example, January must be −−01, not just −−1. In particular, a year must be padded to at least four digits, so "98" is not a valid year, although "1998" and "0098" are both valid years, unambiguously 19 centuries separated from each other. A year may also be preceded by a minus symbol: 0000 is 1 BC and −0001 is 2 BC. Finally a timezone is always allowed (yet optional) at the end. Timezones must be either "Z" (UTC, which we translate to GMT), or simple offsets from UTC in the range "−14:00" to "+14:00", for example: "14:30:00−05:00" specifies 2:30 PM in the afternoon at UTC−05:00, which is the same as EST. If a timezone is not specified, the default TimeZone is used.

XmlCalendar

public XmlCalendar(Date *date***)**

Constructs an XmlCalendar from a Date. The default TimeZone is used for computing the various fields.

Method Detail

computeTime() Method

protected void computeTime()

Overrides GregorianCalendar.computeTime to apply a different default year. (It must be a leap year.)

Overrides

`GregorianCalendar.computeTime()`

---

get(int) Method

public int get(int *field***)**

Gets the value for a given time field. Unlike the GregorianCalendar implementation, the get() does not force a complete of all fields. If you wish to force a completion of all the fields, call getTime() first.

Overrides

`Calendar.get(int)`

---

getDefaultYear() Method

`public static int getDefaultYear()`

Returns the default year that is used when no year is specified.

---

setDefaultYear(int) Method

`public static void setDefaultYear(int` *`year`***`)`**

Sets the default year to be used when no year is specified.

---

toString() Method

`public String toString()`

Prints the XmlCalendar using a standard XML Schema format, as described in XmlCalendar(String s).

Overrides

`Calendar.toString()`

XmlCursor Interface

public interface ***XmlCursor***

extends `XmlTokenSource`

Represents a position between two logical tokens in an XML document. The tokens themselves are not exposed as objects, but their type and properties are discoverable through methods on the cursor. In particular, the general category of token is represented by a `XmlCursor.TokenType`. You use an XmlCursor instance to navigate through and manipulate an XML instance document. Once you obtain an XML document, you can create a cursor to represent a specific place in the XML. Because you can use a cursor with or without a schema corresponding to the XML, cursors are an ideal way to handle XML without a schema. You can create a new cursor by calling the `newCursor` method exposed by an object representing the XML, whether it was parsed into a strong type compiled from schema or an `XmlObject` (as in the no−schema case). With an XmlCursor, you can also:

- Execute XQuery and XPath expressions against the XML with the execQuery and selectPath methods.
- Edit and reshape the document by inserting, moving, copying, and removing XML.
- Insert bookmarks that "stick" to the XML at the cursor's position even if the cursor or XML moves.
- Get and set values for containers (elements and whole documents), attributes, processing instructions, and comments.

A cursor moves through XML by moving past tokens. A token represents a category of XML markup, such as the start of an element, its end, an attribute, comment, and so on. XmlCursor methods such as toNextToken, toNextSibling, toParent, and so on move the cursor among tokens. Each token's category is of a particular *type*, represented by one of the nine types defined by the `XmlCursor.TokenType` class. When you get a new cursor for a whole instance document, the cursor is intially located before the STARTDOC token. This token, which has no analogy in the XML specification, is present in this logical model of XML so that you may distinguish between the document as a whole and the content of the document. Terminating the document is an ENDDOC token. This token is also not part of the XML specification. A cursor located immediately before this token is at the very end of the document. It is not possible to position the cursor after the ENDDOC token. Thus, the STARTDOC and ENDDOC tokens are effectively "bookends" for the content of the document. For example, for the following XML, if you were the navigate a cursor through the XML document using toNextToken(), the list of token types that follows represents the token sequence you would encounter.

```
<sample x='y'>
    <value>foo</value>
</sample>
```

STARTDOC START (sample) ATTR (x='y') TEXT ("\n ") START (value) TEXT ("foo") END (value) TEXT ("\n") END (sample) ENDDOC When there are no more tokens available, hasNextToken() returns false and toNextToken() returns the special token type NONE and does not move the cursor. The `currentTokenType()` method will return the type of the token that is immediately after the cursor. You can also use a number of convenience methods that test for a particular token type. These include the methods isStart(), isStartdoc(), isText(), isAttr(), and so on. Each returns a boolean value indicating whether the token

that follows the cursor is the type in question. A few other methods determine whether the token is of a kind that may include multiple token types. The isAnyAttr() method, for example, returns true if the token immediately following the cursor is any kind of attribute, including those of the ATTR token type and xmlns attributes. Legitimate sequences of tokens for an XML document are described by the following Backus−Naur Form (BNF):

```
<doc> ::= STARTDOC <attributes> <content> ENDDOC
<element> ::= START <attributes> <content> END
<attributes> ::= ( ATTR | NAMESPACE ) *
<content> ::= ( COMMENT | PROCINST | TEXT | <element> ) *
```

Note that a legitimate sequence is STARTDOC ENDDOC, the result of creating a brand new instance of an empty document. Also note that attributes may only follow container tokens (STARTDOC or START)

All Superinterfaces

`XmlTokenSource`

Nested Class Summary

`public static interface`	**`XmlCursor.ChangeStamp`** Represents the state of a dcoument at a particular point in time.
`public static final class`	**`XmlCursor.TokenType`** An enumeration that identifies the type of an XML token.
`public abstract static class`	**`XmlCursor.XmlBookmark`** Subclasses of XmlBookmark can be used to annotate an XML document.
`public static interface`	**`XmlCursor.XmlMark`** An abstract `XmlCursor` factory.

Method Summary

`public void`
addToSelection()
Appends the current location of the cursor to the selection.

`public void`
beginElement(`QName` *name*)
Inserts a new element around this cursor, giving the element the specified qualified name.

`public void`
beginElement(`String` *localName*)
Inserts a new element around this cursor, giving the element the specified local name.

`public void`
beginElement(`String` *localName*, `String` *uri*)

Inserts a new element around this cursor, giving the element the specified local name and associating it with the specified namespace.

`public void` **`clearBookmark`**`(Object key)`
Clears the bookmark whose key is specified, if the bookmark exists at this cursor's location.

`public void` **`clearSelections`**`()`
Clears this cursor's selection, but does not modify the document.

`public int` **`comparePosition`**`(XmlCursor cursor)`
Returns an integer indicating whether this cursor is before, after, or at the same position as the specified cursor.

`public int` **`copyChars`**`(int maxCharacterCount, XmlCursor toHere)`
Copies characters to the position immediately after the specified cursor.

`public boolean` **`copyXml`**`(XmlCursor toHere)`
Copies the XML immediately after this cursor to the location specified by the *toHere* cursor.

`public boolean` **`copyXmlContents`**`(XmlCursor toHere)`
Copies the contents of the container (STARTDOC OR START) immediately to the right of the cursor to the location specified by the *toHere* cursor.

`public XmlCursor.TokenType` **`currentTokenType`**`()`
Returns the type of the current token.

`public void` **`dispose`**`()`
Deallocates resources needed to manage the cursor, rendering this cursor inoperable.

`public XmlCursor` **`execQuery`**`(String query)`
Executes the specified XQuery expression against the XML this cursor is in.

`public XmlCursor` **`execQuery`**`(String query, XmlOptions options)`
Executes the specified XQuery expression against the XML this cursor is in, and using the specified options.

`public void` **`getAllBookmarkRefs`**`(Collection listToFill)`
Retrieves all the bookmarks at this location, adding them to the specified collection.

`public void` **`getAllNamespaces`**`(Map addToThis)`

Adds to the specified map, all the namespaces in scope at the container where this cursor is positioned.

`public String` **`getAttributeText`**`(QName attrName)`
When at a START or STARTDOC, returns the attribute text for the given attribute.

`public XmlCursor.XmlBookmark` **`getBookmark`**`(Object key)`
Retrieves the bookmark with the specified key at this cursor's location.

`public String` **`getChars`**`()`
Returns characters to the right of the cursor up to the next token.

`public int` **`getChars`**`(char[] returnedChars, int offset, int maxCharacterCount)`
Copies characters up to the specified maximum number, counting right from this cursor's location to the character at *maxCharacterCount*.

`public XmlCursor.ChangeStamp` **`getDocChangeStamp`**`()`
Returns the current change stamp for the document the current cursor is in.

`public QName` **`getName`**`()`
Returns the name of the current token.

`public XmlObject` **`getObject`**`()`
Returns the strongly−typed XmlObject at the current START, STARTDOC, or ATTR.

`public int` **`getSelectionCount`**`()`
Returns the count of the current selection.

`public String` **`getTextValue`**`()`
Gets the text value of the current document, element, attribute, comment, procinst or text token.

`public int` **`getTextValue`**`(char[] returnedChars, int offset, int maxCharacterCount)`
Copies the text value of the current document, element, attribute, comment, processing instruction or text token, counting right from this cursor's location up to *maxCharacterCount*, and copies the returned text into *returnedChars*.

`public boolean` **`hasNextSelection`**`()`
Returns whether or not there is a next selection.

`public boolean` **`hasNextToken`**`()`
True if there is a next token.

`public boolean` ***hasPrevToken***()

True if there is a previous token.

`public void` ***insertAttribute***(`String` *localName*)

Inserts a new attribute immediately before this cursor's location, giving it the specified local name.

`public void` ***insertAttribute***(`String` *localName*, `String` *uri*)

Inserts a new attribute immediately before this cursor's location, giving it the specified local name and associating it with the specified namespace.

`public void` ***insertAttribute***(`QName` *name*)

Inserts a new attribute immediately before this cursor's location, giving it the specified name.

`public void` ***insertAttributeWithValue***(`String` *Name*, `String` *value*)

Inserts a new attribute immediately before this cursor's location, giving it the specified value and name.

`public void` ***insertAttributeWithValue***(`String` *name*, `String` *uri*, `String` *value*)

Inserts an attribute immediately before the cursor's location, giving it the specified name and value, and associating it with the specified namespace.

`public void` ***insertAttributeWithValue***(`QName` *name*, `String` *value*)

Inserts an attribute immediately before the cursor's location, giving it the specified name and value.

`public void` ***insertChars***(`String` *text*)

Inserts the specified text immediately before this cursor's location.

`public void` ***insertComment***(`String` *text*)

Inserts an XML comment immediately before the cursor's location, giving it the specified content.

`public void` ***insertElement***(`QName` *name*)

Inserts an element immediately before this cursor's location, giving the element the specified qualified name.

`public void` ***insertElement***(`String` *localName*)

Inserts an element immediately before this cursor's location, giving the element the specified local name.

`public void` ***insertElement***(`String` *localName*, `String` *uri*)

Inserts a new element immediately before this cursor's location, giving the element the specified local name and associating it with

specified namespace

public void
insertElementWithText(QName *name*, String *text*)
Inserts a new element immediately before this cursor's location, giving the element the specified qualified name and content.

public void
insertElementWithText(String *localName*, String *text*)
Inserts a new element immediately before this cursor's location, giving the element the specified local name and content.

public void
insertElementWithText(String *localName*, String *uri*, String *text*)
Inserts a new element immediately before this cursor's location, giving the element the specified local name, associating it with the specified namespace, and giving it the specified content.

public void
insertNamespace(String *prefix*, String *namespace*)
Inserts a namespace declaration immediately before the cursor's location, giving it the specified prefix and URI.

public void
insertProcInst(String *target*, String *text*)
Inserts an XML processing instruction immediately before the cursor's location, giving it the specified target and text.

public boolean
isAnyAttr()
True if this token is any attribute.

public boolean
isAtSamePositionAs(XmlCursor *cursor*)
Determines if this cursor is at the same position as the specified cursor.

public boolean
isAttr()
True if this token is an ATTR token type, meaning just before an attribute.

public boolean
isComment()
True if this token is a COMMENT token type, meaning just before a comment.

public boolean
isContainer()
True if this token is a container token.

public boolean
isEnd()
True if this token is an END token type, meaning just before an element's end.

public boolean
isEnddoc()

True if this token is an ENDDOC token type, meaning at the very end of the document.

`public boolean`
`isFinish``()`
True if this token is a finish token.

`public boolean`
`isInSameDocument``(XmlCursor cursor)`
Determines if the specified cursor is in the same document as this cursor.

`public boolean`
`isLeftOf``(XmlCursor cursor)`
Determines if this cursor is to the left of (or before) the specified cursor.

`public boolean`
`isNamespace``()`
True if this token is a NAMESPACE token type, meaning just before a namespace declaration.

`public boolean`
`isProcinst``()`
True if this token is a PROCINST token type, meaning just before a processing instruction.

`public boolean`
`isRightOf``(XmlCursor cursor)`
Determines if this cursor is to the right of (or after) the specified cursor.

`public boolean`
`isStart``()`
True if this token is a START token type, meaning just before an element's start.

`public boolean`
`isStartdoc``()`
True if the current token is a STARTDOC token type, meaning at the very root of the document.

`public boolean`
`isText``()`
True if the this token is a TEXT token type, meaning just before or inside text.

`public int`
`moveChars``(int maxCharacterCount, XmlCursor toHere)`
Moves characters immediately after this cursor to the position immediately after the specified cursor.

`public boolean`
`moveXml``(XmlCursor toHere)`
Moves the XML immediately after this cursor to the location specified by the *toHere* cursor, shifting XML at that location to the right to make room.

`public boolean`
`moveXmlContents``(XmlCursor toHere)`

Moves the contents of the container (STARTDOC OR START) immediately after this cursor to the location specified by the *toHere* cursor.

`public String` **`namespaceForPrefix`**`(String` *prefix)*

Returns the namespace URI indicated by the given prefix.

`public boolean` **`pop`**`()`

Restores the cursor location most recently saved with the push() method.

`public String` **`prefixForNamespace`**`(String` *namespaceURI)*

Returns a prefix that can be used to indicate a namespace URI.

`public XmlCursor.TokenType` **`prevTokenType`**`()`

Returns the type of the previous token.

`public void` **`push`**`()`

Saves the current location of this cursor on an internal stack of saved positions (independent of selection).

`public boolean` **`removeAttribute`**`(QName` *attrName)*

When at a START or STARTDOC, removes the attribute with the given name.

`public int` **`removeChars`**`(int` *maxCharacterCount)*

Removes characters up to the specified maximum number, counting right from this cursor's location to the character at *maxCharacterCount*.

`public boolean` **`removeXml`**`()`

Removes the XML that is immediately after this cursor.

`public boolean` **`removeXmlContents`**`()`

Removes the contents of the container (STARTDOC OR START) immediately after this cursor.

`public void` **`selectPath`**`(String` *path)*

Executes the specified XPath expression against the XML that this cursor is in.

`public void` **`selectPath`**`(String` *path,* `XmlOptions` *options)*

Executes the specified XPath expression against the XML that this cursor is in.

`public boolean` **`setAttributeText`**`(QName` *attrName,* `String` *value)*

When at a START or STARTDOC, sets the attribute text for the given attribute.

`public void` ***setBookmark***(`XmlCursor.XmlBookmark` *bookmark*)
Sets a bookmark to the document at this cursor's location.

`public void` ***setName***(`QName` *name*)
Sets the name of the current token.

`public void` ***setTextValue***(`String` *text*)
Sets the text value of the XML at this cursor's location if that XML's token type is START, STARTDOC, ATTR, COMMENT or PROCINST.

`public void` ***setTextValue***(*char[] sourceChars, int offset, int length*)
Sets the text value of the XML at this cursor's location (if that XML's token type is START, STARTDOC, ATTR, COMMENT or PROCINST) to the contents of the specified character array.

`public boolean` ***toBookmark***(`XmlCursor.XmlBookmark` *bookmark*)
Moves this cursor to the same position as the bookmark.

`public boolean` ***toChild***(`String` *name*)
Moves the cursor to the first child element of the specified name in no namespace.

`public boolean` ***toChild***(`String` *namespace*, `String` *name*)
Moves the cursor to the first child element of the specified name in the specified namespace.

`public boolean` ***toChild***(`QName` *name*)
Moves the cursor to the first child element of the specified qualified name.

`public boolean` ***toChild***(*int index*)
Moves the cursor to the child element specified by *index*.

`public boolean` ***toChild***(`QName` *name*, *int index*)
Moves the cursor to the specified *index* child element of the specified name, where that element is the .

`public boolean` ***toCursor***(`XmlCursor` *moveTo*)
Moves this cursor to the same position as the moveTo cursor.

`public void` ***toEndDoc***()
Moves the cursor to the ENDDOC token, which is the end of the document.

`public XmlCursor.TokenType` ***toEndToken***()
Moves the cursor to the END or ENDDOC token corresponding to the current START or STARTDOC, and returns END or ENDDOC.

`public boolean` ***toFirstAttribute()***
Moves the cursor to the first attribute of this element, or returns false and does not move the cursor if there are no attributes.

`public boolean` ***toFirstChild()***
Moves the cursor to the first child element, or returns false and does not move the cursor if there are no element children.

`public XmlCursor.TokenType` ***toFirstContentToken()***
Moves the cursor to the first token in the content of the current START or STARTDOC.

`public boolean` ***toLastAttribute()***
Moves the cursor to the last attribute of this element, or returns false and does not move the cursor if there are no attributes.

`public boolean` ***toLastChild()***
Moves the cursor to the last element child, or returns false and does not move the cursor if there are no element children.

`public boolean` ***toNextAttribute()***
Moves the cursor to the next sibling attribute, or returns false and does not move the cursor if there is no next sibling attribute.

`public XmlCursor.XmlBookmark` ***toNextBookmark(`Object` key)***
Moves this cursor to the location after its current position where a bookmark with the given key exists.

`public int` ***toNextChar(int maxCharacterCount)***
Moves the cursor forward by the specified number of characters, and stops at the next non−TEXT token.

`public boolean` ***toNextSelection()***
Moves this cursor to the next location in the selection, if any.

`public boolean` ***toNextSibling()***
Moves the cursor to the next sibling element, or returns false and does not move the cursor if there is no next sibling element.

`public boolean` ***toNextSibling(`String` name)***
Moves the cursor to the next sibling element of the specified name in no namespace.

`public boolean` ***toNextSibling(`String` namespace, `String` name)***
Moves the cursor to the next sibling element of the specified name in the specified namespace.

`public boolean` ***toNextSibling(`QName` name)***

Moves the cursor to the next sibling element of the specified qualified name.

`public XmlCursor.TokenType` ***toNextToken()***

Moves the cursor to the next token.

`public boolean` ***toParent()***

Moves the cursor to the parent element or STARTDOC, or returns false and does not move the cursor if there is no parent.

`public boolean` ***toPrevAttribute()***

Moves the cursor to the previous sibling attribute, or returns false and does not move the cursor if there is no previous sibling attribute.

`public XmlCursor.XmlBookmark` ***toPrevBookmark***`(Object` *key)*

Moves this cursor to the location before its current position where a bookmark with the given key exists.

`public int` ***toPrevChar****(int maxCharacterCount)*

Moves the cursor backwards by the number of characters given.

`public boolean` ***toPrevSibling()***

Moves the cursor to the previous sibling element, or returns false and does not move the cursor if there is no previous sibling element.

`public XmlCursor.TokenType` ***toPrevToken()***

Moves the cursor to the previous token.

`public boolean` ***toSelection****(int i)*

Moves this cursor to the specified location in the selection.

`public void` ***toStartDoc()***

Moves the cursor to the STARTDOC token, which is the root of the document.

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Method Detail

addToSelection() Method

`public void addToSelection()`

Appends the current location of the cursor to the selection. See also the selectPath() method. You can use this as an alternative to calling the selectPath method when you want to define your own selection.

---

beginElement(QName) Method

```
public void beginElement(QName name)
```

Inserts a new element around this cursor, giving the element the specified qualified name. After the element is inserted, this cursor is between its start and end. This cursor can then be used to insert additional XML into the new element.

Parameters

name
: The qualified name for the new element.

---

beginElement(String) Method

```
public void beginElement(String localName)
```

Inserts a new element around this cursor, giving the element the specified local name. After the element is inserted, this cursor is between its start and end. This cursor can then be used to insert additional XML into the new element.

Parameters

localName
: The local name for the new element.

---

beginElement(String, String) Method

```
public void beginElement(String localName,
                        String uri)
```

Inserts a new element around this cursor, giving the element the specified local name and associating it with the specified namespace. After the element is inserted, this cursor is between its start and end. This cursor can then be used to insert additional XML into the new element.

Parameters

localName
: The local name for the new element.

uri
: The URI for the new element's namespace.

---

clearBookmark(Object) Method

```
public void clearBookmark(Object key)
```

Clears the bookmark whose key is specified, if the bookmark exists at this cursor's location.

Parameters

key
The for the bookmark to clear.

---

clearSelections() Method

```
public void clearSelections()
```

Clears this cursor's selection, but does not modify the document.

---

comparePosition(XmlCursor) Method

```
public int comparePosition(XmlCursor cursor)
```

Returns an integer indicating whether this cursor is before, after, or at the same position as the specified cursor. `a.comparePosition(b) < 0` means a is to the left of b. `a.comparePosition(b) == 0` means a is at the same position as b. `a.comparePosition(b) > 0` means a is to the right of b. The sort order of cursors in the document is the token order. For example, if cursor "a" is at a START token and the cursor "b" is at a token within the contents of the same element, then a.comparePosition(b) will return −1, meaning that the position of a is before b.

Parameters

cursor
The cursor whose position should be compared with this cursor.

Returns

1 if this cursor is after the specified cursor; 0 if this cursor is at the same position as the specified cursor; −1 if this cursor is before the specified cursor.

---

copyChars(int, XmlCursor) Method

```
public int copyChars(int maxCharacterCount,
                     XmlCursor toHere)
```

Copies characters to the position immediately after the specified cursor. Characters are counted to the right up to the specified maximum number. XML after the destination cursor is shifted to the right to make room.

Parameters

maxCharacterCount
The maximum number of characters after this cursor's location to copy.

toHere
The cursor to which the characters should be copied.

Returns

The actual number of characters copied.

---

copyXml(XmlCursor) Method

```
public boolean copyXml(XmlCursor toHere)
```

Copies the XML immediately after this cursor to the location specified by the *toHere* cursor. For the TEXT, ATTR, NAMESPACE, COMMENT and PROCINST tokens, a single token is copied. For a start token, the element and all of its contents are copied. For all other tokens, this is a no−op. The cursors and bookmarks located in the XML that was copied are also copied to the new location.

Parameters

toHere
The cursor at the location to which the XML should be copied.

Returns

true if anything was copied; false if the token supports the operation, but nothing was copied.

---

copyXmlContents(XmlCursor) Method

```
public boolean copyXmlContents(XmlCursor toHere)
```

Copies the contents of the container (STARTDOC OR START) immediately to the right of the cursor to the location specified by the *toHere* cursor. For all other situations, returns false. Does not copy attributes or namespaces.

Parameters

toHere
The cursor at the location to which the XML should be copied.

Returns

true if anything was copied; otherwise, false.

---

currentTokenType() Method

`public XmlCursor.TokenType currentTokenType()`

Returns the type of the current token. By definition, the current token is the token immediately to the right of the cursor. If you're in the middle of text, before a character, you get TEXT. You can't dive into the text of an ATTR, COMMENT or PROCINST. As an alternative, it may be more convenient for you to use one of the methods that test for a particular token type. These include the methods isStart(), isStartdoc(), isText(), isAttr(), and so on. Each returns a boolean value indicating whether the token that follows the cursor is the type in question.

Returns

The TokenType instance for the token at the cursor's current location.

---

dispose() Method

`public void dispose()`

Deallocates resources needed to manage the cursor, rendering this cursor inoperable. Because cursors are managed by a mechanism which stores the XML, simply letting a cursor go out of scope and having the garbage collector attempt to reclaim it may not produce desirable performance. So, explicitly disposing a cursor allows the underlying implementation to release its responsibility of maintaining its position. After a cursor has been disposed, it may not be used again. It can throw IllegalStateException or NullPointerException if used after disposal.

---

execQuery(String) Method

`public XmlCursor execQuery(String` *`query`***`)`**

Executes the specified XQuery expression against the XML this cursor is in. The query may be a String or a compiled query. You can precompile an XQuery expression using the XmlBeans.compileQuery method. The root referred to by the expression should be given as a dot. The following is an example path expression:

```
XmlCursor results = cursor.execQuery("purchase-order/line-item[price <= 20.00]");
```

Parameters

query
: The XQuery expression to execute.

Returns

A cursor containing the results of the query.

---

execQuery(String, XmlOptions) Method

```
public XmlCursor execQuery(String query,
                           XmlOptions options)
```

Executes the specified XQuery expression against the XML this cursor is in, and using the specified options.

Parameters

query
The XQuery expression to execute.

options
Options for the query. For example, you can call the `XmlOptions.setXqueryCurrentNodeVar(String)` method to specify a particular name for the query expression variable that indicates the context node.

---

getAllBookmarkRefs(Collection) Method

```
public void getAllBookmarkRefs(Collection listToFill)
```

Retrieves all the bookmarks at this location, adding them to the specified collection. Bookmarks held by weak references are added to this collection as Weak referenced objects pointing to the bookmark.

Parameters

listToFill
The collection that will contain bookmarks returned by this method.

---

getAllNamespaces(Map) Method

```
public void getAllNamespaces(Map addToThis)
```

Adds to the specified map, all the namespaces in scope at the container where this cursor is positioned. This method is useful for container tokens only.

Parameters

addToThis
The Map to add the namespaces to.

---

getAttributeText(QName) Method

```
public String getAttributeText(QName attrName)
```

When at a START or STARTDOC, returns the attribute text for the given attribute. When not at a START or STARTDOC or the attribute does not exist, returns null.

Parameters

attrName
The name of the attribute whose value is requested.

Returns

The attribute's value if it has one; otherwise, null.

---

getBookmark(Object) Method

```
public XmlCursor.XmlBookmark getBookmark(Object key)
```

Retrieves the bookmark with the specified key at this cursor's location. If there is no bookmark whose key is given by the specified key at the current position, null is returned. If the `getKey` method is not overridden on the bookmark, then the bookmark's class is used as the key.

Parameters

key
The key for the bookmark to retrieve.

Returns

The requested bookmark; null if there is no bookmark corresponding to the specified key.

---

getChars() Method

```
public String getChars()
```

Returns characters to the right of the cursor up to the next token.

---

getChars(char[], int, int) Method

```
public int getChars(char[] returnedChars,
                    int offset,
                    int maxCharacterCount)
```

Copies characters up to the specified maximum number, counting right from this cursor's location to the character at *maxCharacterCount*. The returned characters are added to *returnedChars*, with the first character copied to the *offset* position. The *maxCharacterCount* parameter should be less than or equal to the length of *returnedChars* minus *offset*. Copies a number of characters, which is either *maxCharacterCount* or the number of characters up to the next token, whichever is less.

Parameters

returnedChars
A character array to hold the returned characters.

offset
The position within returnedChars at which the first of the returned characters should be added.

maxCharacterCount
The maximum number of characters after this cursor's location to return.

Returns

The actual number of characters returned; 0 if no characters were returned or if the current token is not TEXT.

---

getDocChangeStamp() Method

```
public XmlCursor.ChangeStamp getDocChangeStamp()
```

Returns the current change stamp for the document the current cursor is in. This change stamp can be queried at a later point in time to find out if the document has changed.

Returns

The change stamp for the document the current cursor is in.

---

getName() Method

```
public QName getName()
```

Returns the name of the current token. Names may be associated with START, ATTR, NAMESPACE or PROCINST. Returns null if there is no name associated with the current token. For START and ATTR, the name returned identifies the name of the element or attribute. For NAMESPACE, the local part of the name is the prefix, while the URI is the namespace defined. For PROCINST, the local part is the target and the uri is "".

Returns

The name of the XML at this cursor's location; null if there is no name.

---

getObject() Method

```
public XmlObject getObject()
```

Returns the strongly−typed XmlObject at the current START, STARTDOC, or ATTR. The strongly−typed object can be cast to the strongly−typed XBean interface corresponding to the XML Schema Type given by result.getSchemaType(). If a more specific type cannot be determined, an XmlObject whose schema type is anyType will be returned.

Returns

The strongly−typed object at the cursor's current location; null if the current location is not a START, STARTDOC, or ATTR.

---

getSelectionCount() Method

```
public int getSelectionCount()
```

Returns the count of the current selection. See also the selectPath() and addToSelection() methods. You may experience better performance if you use the iteration model using the toNextSelection method, rather than the indexing model using the getSelectionCount and toSelection methods.

Returns

A number indicating the size of the current selection.

---

getTextValue() Method

```
public String getTextValue()
```

Gets the text value of the current document, element, attribute, comment, procinst or text token. When getting the text value of an element, non−text content such as comments and processing instructions are ignored and text is concatenated. For elements that have nested element children, this returns the concatenated text of all mixed content and the text of all the element children, recursing in first−to−last depthfirst order. For attributes, this returns the attribute value. For comments and processing instructions, this returns the text contents of the comment or PI, not including the delimiting sequences <!−− −−>, <? ?>. If the current token is not a START, STARTDOC, TEXT, ATTR, COMMENT, or PROCINST, this returns null. The value of an empty tag is the empty string.

Returns

The text value of the current token, if the token's type is START, STARTDOC, TEXT, ATTR, COMMENT, or PROCINST; null if the type is END, ENDDOC, NONE, or NAMESPACE.

---

getTextValue(char[], int, int) Method

```
public int getTextValue(char[] returnedChars,
                        int offset,
                        int maxCharacterCount)
```

Copies the text value of the current document, element, attribute, comment, processing instruction or text token, counting right from this cursor's location up to *maxCharacterCount*, and copies the returned text into *returnedChars*. When getting the text value of an element, non−text content such as comments and processing instructions are ignored and text is concatenated. For elements that have nested element children, this returns the concatenated text of all mixed content and the text of all the element children, recursing in first−to−last depthfirst order. For attributes, this returns the attribute value. For comments and processing instructions, this returns the text contents of the comment or PI, not including the delimiting sequences <!−− −−>, <? ?>. If the

current token is END, ENDDOC, or NAMESPACE, this returns 0. The value of an empty tag is the empty string.

Parameters

returnedChars
A character array to hold the returned characters.

offset
The position within returnedChars to which the first of the returned characters should be copied.

maxCharacterCount
The maximum number of characters after this cursor's location to copy.

Returns

The actual number of characters copied; 0 if no characters were copied.

---

hasNextSelection() Method

```
public boolean hasNextSelection()
```

Returns whether or not there is a next selection.

Returns

true if there is a next selection; otherwise, false.

---

hasNextToken() Method

```
public boolean hasNextToken()
```

True if there is a next token. When this is false, as when the cursor is at the ENDDOC token, the toNextToken() method returns NONE and does not move the cursor.

Returns

true if there is a next token; otherwise, false.

---

hasPrevToken() Method

```
public boolean hasPrevToken()
```

True if there is a previous token. When this is false, toPrevToken returns NONE and does not move the cursor.

Returns

true if there is a previous token; otherwise, false.

---

insertAttribute(String) Method

```
public void insertAttribute(String localName)
```

Inserts a new attribute immediately before this cursor's location, giving it the specified local name.

Parameters

localName
The local name for the new attribute.

---

insertAttribute(String, String) Method

```
public void insertAttribute(String localName,
                            String uri)
```

Inserts a new attribute immediately before this cursor's location, giving it the specified local name and associating it with the specified namespace.

Parameters

localName
The local name for the new attribute.

uri
The URI for the new attribute's namespace.

---

insertAttribute(QName) Method

```
public void insertAttribute(QName name)
```

Inserts a new attribute immediately before this cursor's location, giving it the specified name.

Parameters

name
The local name for the new attribute.

---

insertAttributeWithValue(String, String) Method

```
public void insertAttributeWithValue(String Name,
                                     String value)
```

Inserts a new attribute immediately before this cursor's location, giving it the specified value and name.

Parameters

Name
The local name for the new attribute.

value
The value for the new attribute.

---

insertAttributeWithValue(String, String, String) Method

```
public void insertAttributeWithValue(String name,
                                     String uri,
                                     String value)
```

Inserts an attribute immediately before the cursor's location, giving it the specified name and value, and associating it with the specified namespace.

Parameters

name
The name for the new attribute.

uri
The URI for the new attribute's namespace.

value
The value for the new attribute.

---

insertAttributeWithValue(QName, String) Method

```
public void insertAttributeWithValue(QName name,
                                     String value)
```

Inserts an attribute immediately before the cursor's location, giving it the specified name and value.

Parameters

name
The name for the new attribute.

value
The value for the new attribute.

---

insertChars(String) Method

`public void insertChars(String text)`

Inserts the specified text immediately before this cursor's location.

Parameters

text
The text to insert.

---

insertComment(String) Method

`public void insertComment(String text)`

Inserts an XML comment immediately before the cursor's location, giving it the specified content.

Parameters

text
The new comment's content.

---

insertElement(QName) Method

`public void insertElement(QName name)`

Inserts an element immediately before this cursor's location, giving the element the specified qualified name.

Parameters

name
The qualified name for the element.

---

insertElement(String) Method

`public void insertElement(String localName)`

Inserts an element immediately before this cursor's location, giving the element the specified local name.

Parameters

localName
The local name for the new element.

---

insertElement(String, String) Method

```
public void insertElement(String localName,
                          String uri)
```

Inserts a new element immediately before this cursor's location, giving the element the specified local name and associating it with specified namespace

Parameters

localName
The local name for the new element.

uri
The URI for the new element's namespace.

---

insertElementWithText(QName, String) Method

```
public void insertElementWithText(QName name,
                                  String text)
```

Inserts a new element immediately before this cursor's location, giving the element the specified qualified name and content.

Parameters

name
The qualified name for the new element.

text
The content for the new element.

---

insertElementWithText(String, String) Method

```
public void insertElementWithText(String localName,
                                  String text)
```

Inserts a new element immediately before this cursor's location, giving the element the specified local name and content.

Parameters

localName
The local name for the new element.

text
The content for the new element.

---

insertElementWithText(String, String, String) Method

```
public void insertElementWithText(String localName,
                                  String uri,
                                  String text)
```

Inserts a new element immediately before this cursor's location, giving the element the specified local name, associating it with the specified namespace, and giving it the specified content.

Parameters

localName
The local name for the new element.

uri
The URI for the new element's namespace.

text
The content for the new element.

---

insertNamespace(String, String) Method

```
public void insertNamespace(String prefix,
                            String namespace)
```

Inserts a namespace declaration immediately before the cursor's location, giving it the specified prefix and URI.

Parameters

prefix
The prefix for the namespace.

namespace
The URI for the namespace.

---

insertProcInst(String, String) Method

```
public void insertProcInst(String target,
                           String text)
```

Inserts an XML processing instruction immediately before the cursor's location, giving it the specified target and text.

Parameters

target
The target for the processing instruction.

text
The new processing instruction's text.

---

isAnyAttr() Method

`public boolean isAnyAttr()`

True if this token is any attribute. This includes an ATTR token type and the NAMESPACE token type attribute.

Returns

true if the current cursor is at any attribute; otherwise, false.

---

isAtSamePositionAs(XmlCursor) Method

`public boolean isAtSamePositionAs(XmlCursor cursor)`

Determines if this cursor is at the same position as the specified cursor. Note that this is the same as `a.comparePosition(b) == 0`

Parameters

cursor
The cursor whose position should be compared with this cursor.

Returns

true if this cursor is at the same position as the specified cursor; otherwise, false.

---

isAttr() Method

`public boolean isAttr()`

True if this token is an ATTR token type, meaning just before an attribute.

Returns

true if this token is an ATTR token type; otherwise, false.

---

isComment() Method

`public boolean isComment()`

True if this token is a COMMENT token type, meaning just before a comment.

Returns

true if this token is a COMMENT token type; otherwise, false.

---

isContainer() Method

```
public boolean isContainer()
```

True if this token is a container token. The STARTDOC and START token types are containers. Containers, including documents and elements, have the same content model. In other words, a document and an element may have the same contents. For example, a document may contain attributes or text, without any child elements.

Returns

true if this token is a container token; otherwise, false.

---

isEnd() Method

```
public boolean isEnd()
```

True if this token is an END token type, meaning just before an element's end.

Returns

true if this token is an END token type; otherwise, false.

---

isEnddoc() Method

```
public boolean isEnddoc()
```

True if this token is an ENDDOC token type, meaning at the very end of the document.

Returns

true if this token is an ENDDOC token type; otherwise, false.

---

isFinish() Method

```
public boolean isFinish()
```

True if this token is a finish token. A finish token can be an ENDDOC or END token type.

Returns

true if this token is a finish token; otherwise, false.

---

isInSameDocument(XmlCursor) Method

`public boolean isInSameDocument(XmlCursor cursor)`

Determines if the specified cursor is in the same document as this cursor.

Parameters

cursor
The cursor that may be in the same document as this cursor.

Returns

true if the specified cursor is in the same document; otherwise, false.

---

isLeftOf(XmlCursor) Method

`public boolean isLeftOf(XmlCursor cursor)`

Determines if this cursor is to the left of (or before) the specified cursor. Note that this is the same as `a.comparePosition(b) < 0`

Parameters

cursor
The cursor whose position should be compared with this cursor.

Returns

true if this cursor is to the left of the specified cursor; otherwise, false.

---

isNamespace() Method

`public boolean isNamespace()`

True if this token is a NAMESPACE token type, meaning just before a namespace declaration.

Returns

true if this token is a NAMESPACE token type; otherwise, false.

---

isProcinst() Method

`public boolean isProcinst()`

True if this token is a PROCINST token type, meaning just before a processing instruction.

Returns

true if this token is a PROCINST token type; otherwise, false.

---

isRightOf(XmlCursor) Method

`public boolean isRightOf(XmlCursor` *`cursor`***`)`**

Determines if this cursor is to the right of (or after) the specified cursor. Note that this is the same as `a.comparePosition(b) > 0`

Parameters

cursor
The cursor whose position should be compared with this cursor.

Returns

true if this cursor is to the right of the specified cursor; otherwise, false.

---

isStart() Method

`public boolean isStart()`

True if this token is a START token type, meaning just before an element's start.

Returns

true if this token is a START token type; otherwise, false.

---

isStartdoc() Method

`public boolean isStartdoc()`

True if the current token is a STARTDOC token type, meaning at the very root of the document.

Returns

true if this token is a STARTDOC token type; otherwise, false.

---

isText() Method

```
public boolean isText()
```

True if the this token is a TEXT token type, meaning just before or inside text.

Returns

true if this token is a TEXT token type; otherwise, false.

---

moveChars(int, XmlCursor) Method

```
public int moveChars(int maxCharacterCount,
                     XmlCursor toHere)
```

Moves characters immediately after this cursor to the position immediately after the specified cursor. Characters are counted to the right up to the specified maximum number. XML after the destination cursor is shifted to the right to make room. The space remaining from moving the characters collapses up to this cursor.

Parameters

maxCharacterCount
: The maximum number of characters after this cursor's location to move.

toHere
: The cursor to which the characters should be moved.

Returns

The actual number of characters moved.

---

moveXml(XmlCursor) Method

```
public boolean moveXml(XmlCursor toHere)
```

Moves the XML immediately after this cursor to the location specified by the *toHere* cursor, shifting XML at that location to the right to make room. For the TEXT, ATTR, NAMESPACE, COMMENT and PROCINST tokens, a single token is moved. For a start token, the element and all of its contents are moved. For all other tokens, this is a no−op. The bookmarks located in the XML that was moved also move to the new location; the cursors don't move with the content.

Parameters

toHere
: The cursor at the location to which the XML should be moved.

Returns

true if anything was moved; false only if the cursor is just before END or ENDDOC token.

---

moveXmlContents(XmlCursor) Method

`public boolean moveXmlContents(XmlCursor toHere)`

Moves the contents of the container (STARTDOC OR START) immediately after this cursor to the location specified by the *toHere* cursor. For all other situations, returns false. Does not move attributes or namespaces.

Parameters

toHere
: The cursor at the location to which the XML should be moved.

Returns

true if anything was moved; otherwise, false.

---

namespaceForPrefix(String) Method

`public String namespaceForPrefix(String prefix)`

Returns the namespace URI indicated by the given prefix. The current context must be at a START or STARTDOC. Namespace prefix mappings are queried for the mappings defined at the current container first, then parents are queried. The prefix can be "" or null to indicate a search for the default namespace. To conform with the XML spec, the default namespace will return the no−namespace ("") if it is not mapped. Note that this queries the current state of the document. When the document is persisted, the saving mechanism may synthesize namespaces (ns1, ns2, and so on) for the purposes of persistence. These namepaces are only present in the serialized form, and are not reflected back into the document being saved.

Parameters

prefix
: The namespace prefix for the requested namespace.

Returns

The URI for corresponding to the specified prefix if it exists; otherwise, null.

---

pop() Method

`public boolean pop()`

Restores the cursor location most recently saved with the push() method.

Returns

true if there was a location to restore; otherwise, false.

---

prefixForNamespace(String) Method

`public String prefixForNamespace(String namespaceURI)`

Returns a prefix that can be used to indicate a namespace URI. The current context must be at a START or STARTDOC. If there is an existing prefix that indicates the URI in the current context, that prefix may be returned. Otherwise, a new prefix for the URI will be defined by adding an xmlns attribute to the current container or a parent container. Note that this queries the current state of the document. When the document is persisted, the saving mechanism may synthesize namespaces (ns1, ns2, and so on) for the purposes of persistence. These namepaces are only present in the serialized form, and are not reflected back into the document being saved.

Parameters

namespaceURI
: The namespace URI corresponding to the requested prefix.

Returns

The prefix corresponding to the specified URI if it exists; otherwise, a newly generated prefix.

---

prevTokenType() Method

`public XmlCursor.TokenType prevTokenType()`

Returns the type of the previous token. By definition, the previous token is the token immediately to the left of the cursor. If you're in the middle of text, after a character, you get TEXT.

Returns

The TokenType instance for the token immediately before the token at the cursor's current location.

---

push() Method

`public void push()`

Saves the current location of this cursor on an internal stack of saved positions (independent of selection). This location may be restored later by calling the pop() method.

---

removeAttribute(QName) Method

`public boolean removeAttribute(QName attrName)`

When at a START or STARTDOC, removes the attribute with the given name.

Parameters

attrName
The name of the attribute that should be removed.

Returns

true if the attribute was removed; otherwise, false.

---

removeChars(int) Method

`public int removeChars(int maxCharacterCount)`

Removes characters up to the specified maximum number, counting right from this cursor's location to the character at *maxCharacterCount*. The space remaining from removing the characters collapses up to this cursor.

Parameters

maxCharacterCount
The maximum number of characters after this cursor's location to remove.

Returns

The actual number of characters removed.

---

removeXml() Method

`public boolean removeXml()`

Removes the XML that is immediately after this cursor. For the TEXT, ATTR, NAMESPACE, COMMENT and PROCINST tokens, a single token is removed. For a START token, the corresponding element and all of its contents are removed. For all other tokens, this is a no−op. You cannot remove a STARTDOC. The cursors located in the XML that was removed all collapse to the same location. All bookmarks in this XML will be orphaned.

Returns

true if anything was removed; false only if the cursor is just before END or ENDDOC token.

---

removeXmlContents() Method

`public boolean removeXmlContents()`

Removes the contents of the container (STARTDOC OR START) immediately after this cursor. For all other situations, returns false. Does not remove attributes or namspaces.

Returns

true if anything was copied; otherwise, false.

---

selectPath(String) Method

`public void selectPath(String` *`path`***`)`**

Executes the specified XPath expression against the XML that this cursor is in. The cursors position does not change. To navigate to the selections, use hasNextSelection and toNextSelection (similar to java.util.Iterator). The root referred to by the expression should be given as a dot. The following is an example path expression:

```
cursor.selectPath("./purchase-order/line-item[price <= 20.00]");
```

Parameters

path
: The path expression to execute.

---

selectPath(String, XmlOptions) Method

`public void selectPath(String` *`path`***`,`**
`XmlOptions` *`options`***`)`**

Executes the specified XPath expression against the XML that this cursor is in. The cursor's position does not change. To navigate to the selections, use hasNextSelection and toNextSelection (similar to java.util.Iterator). The root referred to by the expression should be given as a dot. The following is an example path expression:

```
cursor.selectPath("./purchase-order/line-item[price <= 20.00]");
```

Parameters

path
: The path expression to execute.

options
: Options for the query. For example, you can call the `XmlOptions.setXqueryCurrentNodeVar(String)` method to specify a particular name for the query expression variable that indicates the context node.

---

setAttributeText(QName, String) Method

```
public boolean setAttributeText(QName attrName,
                                String value)
```

When at a START or STARTDOC, sets the attribute text for the given attribute. When not at a START or STARTDOC returns false. If the attribute does not exist, one is created.

Parameters

attrName
The name of the attribute whose value is being set.

value
The new value for the attribute.

Returns

true if the new value was set; otherwise, false.

---

setBookmark(XmlCursor.XmlBookmark) Method

```
public void setBookmark(XmlCursor.XmlBookmark bookmark)
```

Sets a bookmark to the document at this cursor's location. The bookmark is attached to the token in the tree immediately after the cursor. If the tree is manipulated to move that object to a different place, the bookmark moves with it. If the tree is manipulated to delete that token from the tree, the bookmark is orphaned. Copy operations do not copy bookmarks.

Parameters

bookmark
The bookmark to set.

---

setName(QName) Method

```
public void setName(QName name)
```

Sets the name of the current token. This token can be START, NAMESPACE, ATTR or PROCINST.

Parameters

name
The new name for the current token.

---

setTextValue(String) Method

```
public void setTextValue(String text)
```

Sets the text value of the XML at this cursor's location if that XML's token type is START, STARTDOC, ATTR, COMMENT or PROCINST. For elements that have nested children this first removes all the content of the element and replaces it with the given text.

Parameters

text
The text to use as a new value.

---

setTextValue(char[], int, int) Method

```
public void setTextValue(char[] sourceChars,
                         int offset,
                         int length)
```

Sets the text value of the XML at this cursor's location (if that XML's token type is START, STARTDOC, ATTR, COMMENT or PROCINST) to the contents of the specified character array. For elements that have nested children this first removes all the content of the element and replaces it with the given text.

Parameters

sourceChars
A character array containing the XML's new value.

offset
The position within sourceChars from which the first of the source characters should be copied.

length
The maximum number of characters to set as the XML's new value.

---

toBookmark(XmlCursor.XmlBookmark) Method

```
public boolean toBookmark(XmlCursor.XmlBookmark bookmark)
```

Moves this cursor to the same position as the bookmark. If the bookmark is in a different document from this cursor or if the bookmark is orphaned, this cursor will not be moved, and false will be returned.

Parameters

bookmark
The bookmark at the location to which this cursor should be moved.

Returns

true if the cursor moved; otherwise, false.

---

toChild(String) Method

```
public boolean toChild(String name)
```

Moves the cursor to the first child element of the specified name in no namespace.

Parameters

name
: The name of the element to move the cursor to.

Returns

true if the cursor was moved; otherwise, false.

---

toChild(String, String) Method

```
public boolean toChild(String namespace,
                       String name)
```

Moves the cursor to the first child element of the specified name in the specified namespace.

Parameters

namespace
: The namespace URI for the element to move the cursor to.

name
: The name of the element to move to.

Returns

true if the cursor was moved; otherwise, false.

---

toChild(QName) Method

```
public boolean toChild(QName name)
```

Moves the cursor to the first child element of the specified qualified name.

Parameters

name
: The name of the element to move the cursor to.

toChild(int) Method

```
public boolean toChild(int index)
```

Moves the cursor to the child element specified by *index*.

Parameters

index
: The position of the element in the sequence of child elements.

Returns

true if the cursor was moved; otherwise, false.

toChild(QName, int) Method

```
public boolean toChild(QName name,
                       int index)
```

Moves the cursor to the specified *index* child element of the specified name, where that element is the .

Parameters

name
: The name of the child element to move the cursor to.

index
: The position of the element in the sequence of child elements.

Returns

true if the cursor was moved; otherwise, false.

toCursor(XmlCursor) Method

```
public boolean toCursor(XmlCursor moveTo)
```

Moves this cursor to the same position as the moveTo cursor. if the moveTo cursor is in a different document from this cursor, this cursor will not be moved, and false returned.

Parameters

moveTo
: The cursor at the location to which this cursor should be moved.

Returns

true if the cursor moved; otherwise, false.

---

toEndDoc() Method

```
public void toEndDoc()
```

Moves the cursor to the ENDDOC token, which is the end of the document.

---

toEndToken() Method

```
public XmlCursor.TokenType toEndToken()
```

Moves the cursor to the END or ENDDOC token corresponding to the current START or STARTDOC, and returns END or ENDDOC. If the current token is not a START or STARTDOC, the cursor is not moved and NONE is returned.

Returns

The new current token type.

---

toFirstAttribute() Method

```
public boolean toFirstAttribute()
```

Moves the cursor to the first attribute of this element, or returns false and does not move the cursor if there are no attributes. The order of attributes is arbitrary, but stable. xmlns attributes (namespace declarations) are not considered attributes by this function. The cursor must be on a START or STARTDOC for this method to succeed.

Returns

true if the cursor was moved; otherwise, false.

---

toFirstChild() Method

```
public boolean toFirstChild()
```

Moves the cursor to the first child element, or returns false and does not move the cursor if there are no element children. If the cursor is not currently in an element, it moves into the first child element of the next element.

Returns

true if the cursor was moved; otherwise, false.

---

toFirstContentToken() Method

```
public XmlCursor.TokenType toFirstContentToken()
```

Moves the cursor to the first token in the content of the current START or STARTDOC. That is, the first token after all ATTR and NAMESPACE tokens associated with this START. If the current token is not a START or STARTDOC, the cursor is not moved and NONE is returned. If the current START or STARTDOC has no content, the cursor is moved to the END or ENDDOC token.

Returns

The new current token type.

---

toLastAttribute() Method

```
public boolean toLastAttribute()
```

Moves the cursor to the last attribute of this element, or returns false and does not move the cursor if there are no attributes. The order of attributes is arbitrary, but stable. xmlns attributes (namespace declarations) are not considered attributes by this function. The cursor must be on a START or STARTDOC for this method to succeed.

Returns

true if the cursor was moved; otherwise, false.

---

toLastChild() Method

```
public boolean toLastChild()
```

Moves the cursor to the last element child, or returns false and does not move the cursor if there are no element children.

Returns

true if the cursor was moved; otherwise, false.

---

toNextAttribute() Method

```
public boolean toNextAttribute()
```

Moves the cursor to the next sibling attribute, or returns false and does not move the cursor if there is no next

sibling attribute. The order of attributes is arbitrary, but stable. xmlns attributes (namespace declarations) are not considered attributes by this function. The cursor must be on an attribute for this method to succeed.

Returns

true if the cursor was moved; otherwise, false.

---

toNextBookmark(Object) Method

`public XmlCursor.XmlBookmark toNextBookmark(Object key)`

Moves this cursor to the location after its current position where a bookmark with the given key exists. Returns false if no such bookmark exists.

Parameters

key
The key held by the next bookmark at the location to which this cursor should be moved.

Returns

The next corresponding bookmark, if it exists; null if there is no next bookmark with the specified key.

---

toNextChar(int) Method

`public int toNextChar(int maxCharacterCount)`

Moves the cursor forward by the specified number of characters, and stops at the next non−TEXT token. Returns the number of characters actually moved across, which is guaranteed to be less than or equal to *maxCharacterCount*. If there is no further text, or if there is no text at all, returns zero. Note this does not dive into attribute values, comment contents, processing instruction contents, etc., but only content text. You can pass maxCharacterCount < 0 to move over all the text to the right. This has the same effect as toNextToken, but returns the amount of text moved over.

Parameters

maxCharacterCount
The maximum number of characters by which the cursor should be moved.

Returns

The actual number of characters by which the cursor was moved; 0 if the cursor was not moved.

---

toNextSelection() Method

`public boolean toNextSelection()`

Moves this cursor to the next location in the selection, if any. See the selectPath() and addToSelection() methods.

Returns

true if the cursor moved; otherwise, false.

---

toNextSibling() Method

```
public boolean toNextSibling()
```

Moves the cursor to the next sibling element, or returns false and does not move the cursor if there is no next sibling element. (By definition the position of an element is the same as the position of its START token.) If the current token is not s START, the cursor will be moved to the next START without moving out of the scope of the current element.

Returns

true if the cursor was moved; otherwise, false.

---

toNextSibling(String) Method

```
public boolean toNextSibling(String name)
```

Moves the cursor to the next sibling element of the specified name in no namespace.

Parameters

name
: The name of the element to move the cursor to.

Returns

true if the cursor was moved; otherwise, false.

---

toNextSibling(String, String) Method

```
public boolean toNextSibling(String namespace,
                             String name)
```

Moves the cursor to the next sibling element of the specified name in the specified namespace.

Parameters

namespace
: The namespace URI for the element to move the cursor to.

name

The name of the element to move the cursor to.

Returns

true if the cursor was moved; otherwise, false.

---

toNextSibling(QName) Method

`public boolean toNextSibling(QName name)`

Moves the cursor to the next sibling element of the specified qualified name.

Parameters

name
The name of the element to move the cursor to.

Returns

true if the cursor was moved; otherwise, false.

---

toNextToken() Method

`public XmlCursor.TokenType toNextToken()`

Moves the cursor to the next token. When there are no more tokens available, hasNextToken returns false and toNextToken() returns NONE and does not move the cursor. Returns the token type of the token to the right of the cursor upon a successful move.

Returns

The token type for the next token if the cursor was moved; otherwise, NONE.

---

toParent() Method

`public boolean toParent()`

Moves the cursor to the parent element or STARTDOC, or returns false and does not move the cursor if there is no parent. Works if you're in attributes or content. Returns false only if at STARTDOC. Note that the parent of an END token is the corresponding START token.

Returns

true if the cursor was moved; false if the cursor is at the STARTDOC token.

---

toPrevAttribute() Method

`public boolean toPrevAttribute()`

Moves the cursor to the previous sibling attribute, or returns false and does not move the cursor if there is no previous sibling attribute. The order of attributes is arbitrary, but stable. xmlns attributes (namespace declarations) are not considered attributes by this function. The cursor must be on an attribute for this method to succeed.

Returns

true if the cursor was moved; otherwise, false.

---

toPrevBookmark(Object) Method

`public XmlCursor.XmlBookmark toPrevBookmark(Object key)`

Moves this cursor to the location before its current position where a bookmark with the given key exists. Returns false if no such bookmark exists.

Parameters

key
: The key held by the previous bookmark at the location to which this cursor should be moved.

Returns

The previous corresponding bookmark, if it exists; null if there is no previous bookmark with the specified key.

---

toPrevChar(int) Method

`public int toPrevChar(int maxCharacterCount)`

Moves the cursor backwards by the number of characters given. Has similar characteristics to the `toNextChar` method.

Parameters

maxCharacterCount
: The maximum number of characters by which the cursor should be moved.

Returns

The actual number of characters by which the cursor was moved; 0 if the cursor was not moved.

---

toPrevSibling() Method

`public boolean toPrevSibling()`

Moves the cursor to the previous sibling element, or returns false and does not move the cursor if there is no previous sibling element. (By definition the position of an element is the same as the position of its START token.)

Returns

true if the cursor was moved; otherwise, false.

---

toPrevToken() Method

`public XmlCursor.TokenType toPrevToken()`

Moves the cursor to the previous token. When there is no previous token, returns NONE, otherwise returns the token to the left of the new position of the cursor.

Returns

The token type for the previous token if the cursor was moved; otherwise, NONE.

---

toSelection(int) Method

`public boolean toSelection(int i)`

Moves this cursor to the specified location in the selection. If i is less than zero or greater than or equal to the selection count, this method returns false. See also the selectPath() and addToSelection() methods.

Parameters

i
: The index of the desired location.

Returns

true if the cursor was moved; otherwise, false.

---

toStartDoc() Method

`public void toStartDoc()`

Moves the cursor to the STARTDOC token, which is the root of the document.

com.bea.xml

XmlCursor.ChangeStamp Interface

public static interface ***XmlCursor.ChangeStamp***

Represents the state of a dcoument at a particular point in time. It is used to determine if a document has been changed since that point in time.

Enclosing interface

`XmlCursor`

Method Summary

`public boolean` ***hasChanged()***
Returns whether or not the document assoiated with this ChangeStamp has been altered since the ChangeStamp had been created.

Method Detail

hasChanged() Method

`public boolean hasChanged()`

Returns whether or not the document assoiated with this ChangeStamp has been altered since the ChangeStamp had been created.

com.bea.xml

XmlCursor.TokenType Class

public static final class ***XmlCursor.TokenType***

extends `Object`

An enumeration that identifies the type of an XML token.

Hierarchy

```
Object
 XmlCursor.TokenType
```

Enclosing interface

`XmlCursor`

Field Summary

`public static final XmlCursor.TokenType` ***ATTR***
The singleton attribute token type

`public static final XmlCursor.TokenType` ***COMMENT***
The singleton comment token type

`public static final XmlCursor.TokenType` ***END***
The singleton end−element token type

`public static final XmlCursor.TokenType` ***ENDDOC***
The singleton start−document token type

`public static final int` ***INT_ATTR***
The attribute token.

`public static final int` ***INT_COMMENT***
The comment token.

`public static final int` ***INT_END***
The end−element token.

`public static final int` ***INT_ENDDOC***
The end−document token.

`public static final int` ***INT_NAMESPACE***
The namespace declaration token.

`public static final int` ***INT_NONE***
No token.

`public static final int` ***INT_PROCINST***
The processing instruction token.

`public static final int` ***INT_START***
The start−element token.

`public static final int` ***INT_STARTDOC***
The start−document token.

`public static final int` ***INT_TEXT***
The text token.

`public static final XmlCursor.TokenType` ***NAMESPACE***
The singleton namespace declaration token type

`public static final XmlCursor.TokenType` ***NONE***
The singleton no−token type

`public static final XmlCursor.TokenType` ***PROCINST***
The singleton processing instruction token type

`public static final XmlCursor.TokenType` ***START***
The singleton start−element token type

`public static final XmlCursor.TokenType` ***STARTDOC***
The singleton start−document token type

`public static final XmlCursor.TokenType` ***TEXT***
The singleton text token type

Method Summary

`public int` ***intValue()***
Returns one of the INT_ values defined in this class.

`public boolean` ***isAnyAttr()***
True if is attribute or namespace declaration token

public boolean ***isAttr()***
True if is attribute token.

public boolean ***isComment()***
True if is comment token.

public boolean ***isContainer()***
True if is start−document or start−element token

public boolean ***isEnd()***
True if is end−element token.

public boolean ***isEnddoc()***
True if is end−document token.

public boolean ***isFinish()***
True if is end−document or end−element token

public boolean ***isNamespace()***
True if is namespace declaration token.

public boolean ***isNone()***
True if no token.

public boolean ***isProcinst()***
True if is processing instruction token.

public boolean ***isStart()***
True if is start−element token.

public boolean ***isStartdoc()***
True if is start−document token.

public boolean ***isText()***
True if is text token.

public String ***toString()***

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

ATTR

`public static final XmlCursor.TokenType ATTR`

The singleton attribute token type

COMMENT

`public static final XmlCursor.TokenType COMMENT`

The singleton comment token type

END

`public static final XmlCursor.TokenType END`

The singleton end−element token type

ENDDOC

`public static final XmlCursor.TokenType ENDDOC`

The singleton start−document token type

INT_ATTR

`public static final int INT_ATTR`

The attribute token. See `XmlCursor.TokenType.intValue()`.

INT_COMMENT

`public static final int INT_COMMENT`

The comment token. See `XmlCursor.TokenType.intValue()`.

INT_END

`public static final int INT_END`

The end−element token. See `XmlCursor.TokenType.intValue()`.

INT_ENDDOC

public static final int INT_ENDDOC

The end−document token. See `XmlCursor.TokenType.intValue()`.

INT_NAMESPACE

public static final int INT_NAMESPACE

The namespace declaration token. See `XmlCursor.TokenType.intValue()`.

INT_NONE

public static final int INT_NONE

No token. See `XmlCursor.TokenType.intValue()`.

INT_PROCINST

public static final int INT_PROCINST

The processing instruction token. See `XmlCursor.TokenType.intValue()`.

INT_START

public static final int INT_START

The start−element token. See `XmlCursor.TokenType.intValue()`.

INT_STARTDOC

public static final int INT_STARTDOC

The start−document token. See `XmlCursor.TokenType.intValue()`.

INT_TEXT

public static final int INT_TEXT

The text token. See `XmlCursor.TokenType.intValue()`.

NAMESPACE

`public static final XmlCursor.TokenType NAMESPACE`

The singleton namespace declaration token type

NONE

`public static final XmlCursor.TokenType NONE`

The singleton no−token type

PROCINST

`public static final XmlCursor.TokenType PROCINST`

The singleton processing instruction token type

START

`public static final XmlCursor.TokenType START`

The singleton start−element token type

STARTDOC

`public static final XmlCursor.TokenType STARTDOC`

The singleton start−document token type

TEXT

`public static final XmlCursor.TokenType TEXT`

The singleton text token type

Method Detail

intValue() Method

`public int intValue()`

Returns one of the INT_ values defined in this class.

---

isAnyAttr() Method

```
public boolean isAnyAttr()
```

True if is attribute or namespace declaration token

---

isAttr() Method

```
public boolean isAttr()
```

True if is attribute token.

---

isComment() Method

```
public boolean isComment()
```

True if is comment token.

---

isContainer() Method

```
public boolean isContainer()
```

True if is start−document or start−element token

---

isEnd() Method

```
public boolean isEnd()
```

True if is end−element token.

---

isEnddoc() Method

```
public boolean isEnddoc()
```

True if is end−document token.

---

isFinish() Method

```
public boolean isFinish()
```

True if is end−document or end−element token

isNamespace() Method

`public boolean isNamespace()`

True if is namespace declaration token.

isNone() Method

`public boolean isNone()`

True if no token.

isProcinst() Method

`public boolean isProcinst()`

True if is processing instruction token.

isStart() Method

`public boolean isStart()`

True if is start−element token.

isStartdoc() Method

`public boolean isStartdoc()`

True if is start−document token.

isText() Method

`public boolean isText()`

True if is text token.

toString() Method

`public String toString()`

Overrides

`Object.toString()`

com.bea.xml

XmlCursor.XmlBookmark Class

public abstract static class ***XmlCursor.XmlBookmark***

extends `Object`

Subclasses of XmlBookmark can be used to annotate an XML document. This class is abstract to prevent parties from inadvertently interfering with each others' bookmarks without explicitly sharing a bookmark class.

Hierarchy

```
Object
 XmlCursor.XmlBookmark
```

Direct Known Subclasses

`XmlLineNumber`

Enclosing interface

`XmlCursor`

Field Summary

`public XmlCursor.XmlMark` ***`_currentMark`***

The mark is set by the host document; it is capable of returning an XmlCursor implementation at the location of the bookmark.

`public final Reference` ***`_ref`***

If non−null, the ref is used by the host document to maintain a reference to the bookmark.

Constructor Summary

XmlCursor.XmlBookmark*()*

Constructs a strongly−referenced bookmark.

XmlCursor.XmlBookmark*(boolean weak)*

Constructs a bookmark.

Method Summary

public final XmlCursor	***createCursor()*** Call the createCursor method to create a new cursor which is positioned at the same splace as the bookmark.
public Object	***getKey()*** The default key for bookmarks is the class which implements them.
public final XmlCursor	***toBookmark***(XmlCursor c) Moves the given cursor to this bookmark, and returns it.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

_currentMark

`public XmlCursor.XmlMark _currentMark`

The mark is set by the host document; it is capable of returning an XmlCursor implementation at the location of the bookmark.

_ref

`public final Reference _ref`

If non−null, the ref is used by the host document to maintain a reference to the bookmark. If it is a weak reference, the host document will not prevent the Bookmark from being garbage collected.

Constructor Detail

XmlCursor.XmlBookmark

`public XmlCursor.XmlBookmark()`

Constructs a strongly−referenced bookmark.

XmlCursor.XmlBookmark

public XmlCursor.XmlBookmark(boolean *weak*)

Constructs a bookmark.

Method Detail

createCursor() Method

public final XmlCursor createCursor()

Call the createCursor method to create a new cursor which is positioned at the same splace as the bookmark. It is much more efficient to call toBookmark on an existing cursor than it is to create a new cursor. However, toBookmark may fail if the bookmark is in a different document than the cursor. It is under these circumstances where createCursor needs to be called on the bookmark. Subsequent navigations to bookmark positions should attempt to reuse the last cursor to improve performace.

---

getKey() Method

public Object getKey()

The default key for bookmarks is the class which implements them. This way, multiple parties using bookmarks in the same instance document will not interfere with eachother. One can, however, override getKey() to use a key other than the class.

---

toBookmark(XmlCursor) Method

public final XmlCursor toBookmark(XmlCursor *c*)

Moves the given cursor to this bookmark, and returns it.

com.bea.xml

XmlCursor.XmlMark Interface

public static interface ***XmlCursor.XmlMark***

An abstract `XmlCursor` factory. Implementations of XmlCursor implement XmlMark to be able to reconstitute a cursor from a bookmark. When content moves between implementations, the XmlMark is set to the implmentation's which recieves the new content.

Enclosing interface

`XmlCursor`

Method Summary

`public XmlCursor` ***`createCursor`****()*

Method Detail

createCursor() Method

`public XmlCursor createCursor()`

com.bea.xml

XmlDate Interface

public interface ***XmlDate***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:date type.

Convertible to `Calendar`, `Date`, and `GDate`.

Related Topics

`XmlCalendar`
`GDate`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlDate.Factory`** A class with methods for creating instances of `XmlDate`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public Calendar` ***calendarValue()***
Returns this value as a `Calendar`

`public Date` ***dateValue()***
Returns this value as a `Date`

`public GDate` ***gDateValue()***
Returns this value as a `GDate`

`public Calendar` ***getCalendarValue()***
Returns this value as a `Calendar`

`public Date` ***getDateValue()***
Returns this value as a `Date`

`public GDate` ***getGDateValue()***
Returns this value as a `GDate`

`public void` ***set***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***set***`(GDateSpecification gd)`
Sets this value as a `GDateSpecification`

`public void` ***set***`(Date d)`
Sets this value as a `Date`

`public void` ***setCalendarValue***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***setDateValue***`(Date d)`
Sets this value as a `Date`

`public void` ***setGDateValue***`(GDate gd)`
Sets this value as a `GDateSpecification`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDate.getCalendarValue()`

`public Calendar calendarValue()`

Returns this value as a `Calendar`

---

dateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDate.getDateValue()`

`public Date dateValue()`

Returns this value as a `Date`

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDate.getGDateValue()`

public GDate gDateValue()

Returns this value as a `GDate`

---

getCalendarValue() Method

public Calendar getCalendarValue()

Returns this value as a `Calendar`

---

getDateValue() Method

public Date getDateValue()

Returns this value as a `Date`

---

getGDateValue() Method

public GDate getGDateValue()

Returns this value as a `GDate`

---

set(Calendar) Method

DEPRECATED replaced with `com.bea.xml.XmlDate.setCalendarValue(java.util.Calendar)`

public void set(Calendar *c*)

Sets this value as a `Calendar`

---

set(GDateSpecification) Method

DEPRECATED replaced with `com.bea.xml.XmlDate.setGDateValue(com.bea.xml.GDate)`

public void set(GDateSpecification *gd*)

Sets this value as a `GDateSpecification`

---

set(Date) Method

DEPRECATED replaced with `com.bea.xml.XmlDate.setDateValue(java.util.Date)`

public void set(Date *d*)

Sets this value as a `Date`

---

setCalendarValue(Calendar) Method

`public void setCalendarValue(Calendar c)`

Sets this value as a `Calendar`

---

setDateValue(Date) Method

`public void setDateValue(Date d)`

Sets this value as a `Date`

---

setGDateValue(GDate) Method

`public void setGDateValue(GDate gd)`

Sets this value as a `GDateSpecification`

XmlDate.Factory Class

public static final class ***XmlDate.Factory***

extends `Object`

A class with methods for creating instances of `XmlDate`.

Hierarchy

```
Object
 XmlDate.Factory
```

Enclosing interface

`XmlDate`

Method Summary

`public static XmlDate` ***newInstance***()
Creates an empty instance of `XmlDate`

`public static XmlDate` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlDate`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlDate` ***newValue***(`Object` *obj*)
Creates an immutable `XmlDate` value

`public static XmlDate` ***parse***(`String` *s*)
Parses a `XmlDate` fragment from a String.

`public static XmlDate` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlDate` fragment from a String.

`public static XmlDate` ***parse***(`File` *f*)
Parses a `XmlDate` fragment from a File.

`public static XmlDate` ***parse***(*File f, XmlOptions options*)
Parses a `XmlDate` fragment from a File.

`public static XmlDate` ***parse***(*URL u*)
Parses a `XmlDate` fragment from a URL.

`public static XmlDate` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlDate` fragment from a URL.

`public static XmlDate` ***parse***(*InputStream is*)
Parses a `XmlDate` fragment from an InputStream.

`public static XmlDate` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlDate` fragment from an InputStream.

`public static XmlDate` ***parse***(*Reader r*)
Parses a `XmlDate` fragment from a Reader.

`public static XmlDate` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlDate` fragment from a Reader.

`public static XmlDate` ***parse***(*Node node*)
Parses a `XmlDate` fragment from a DOM Node.

`public static XmlDate` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlDate` fragment from a DOM Node.

`public static XmlDate` ***parse***(*XMLInputStream xis*)
Parses a `XmlDate` fragment from an XMLInputStream.

`public static XmlDate` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlDate` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlDate newInstance()`

Creates an empty instance of `XmlDate`

---

newInstance(XmlOptions) Method

```
public static XmlDate newInstance(XmlOptions options)
```

Creates an empty instance of `XmlDate`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlDate newValue(Object obj)
```

Creates an immutable `XmlDate` value

---

parse(String) Method

```
public static XmlDate parse(String s)
     throws XmlException
```

Parses a `XmlDate` fragment from a String. For example: "`<xml-fragment>2003-06-14</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlDate parse(String s,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlDate` fragment from a String. For example: "`<xml-fragment>2003-06-14</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlDate parse(File f)
     throws XmlException, IOException
```

Parses a `XmlDate` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlDate parse(File f,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDate` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlDate parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlDate` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlDate parse(URL u,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDate` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlDate parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlDate` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlDate parse(InputStream is,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDate` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlDate parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlDate` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlDate parse(Reader r,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDate` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlDate parse(Node node)
     throws XmlException
```

Parses a `XmlDate` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlDate parse(Node node,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlDate` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDate parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlDate` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDate parse(XMLInputStream xis,
                            XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlDate` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlDateTime Interface

public interface ***XmlDateTime***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:dateTime type.

Convertible to `Calendar`, `Date`, and `GDate`.

Related Topics

`XmlCalendar`
`GDate`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlDateTime.Factory`** A class with methods for creating instances of `XmlDateTime`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public Calendar` ***calendarValue()***
Returns this value as a `Calendar`

`public Date` ***dateValue()***
Returns this value as a `Date`

`public GDate` ***gDateValue()***
Returns this value as a `GDate`

`public Calendar` ***getCalendarValue()***
Returns this value as a `Calendar`

`public Date` ***getDateValue()***
Returns this value as a `Date`

`public GDate` ***getGDateValue()***
Returns this value as a `GDate`

`public void` ***set***`(Calendar` *c)*
Sets this value as a `Calendar`

`public void` ***set***`(GDateSpecification` *gd)*
Sets this value as a `GDateSpecification`

`public void` ***set***`(Date` *d)*
Sets this value as a `Date`

`public void` ***setCalendarValue***`(Calendar` *c)*
Sets this value as a `Calendar`

`public void` ***setDateValue***`(Date` *d)*
Sets this value as a `Date`

`public void` ***setGDateValue***`(GDate` *gd)*
Sets this value as a `GDateSpecification`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDateTime.getCalendarValue()`

`public Calendar calendarValue()`

Returns this value as a `Calendar`

---

dateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDateTime.getDateValue()`

`public Date dateValue()`

Returns this value as a `Date`

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDateTime.getGDateValue()`

public GDate gDateValue()

Returns this value as a `GDate`

---

getCalendarValue() Method

public Calendar getCalendarValue()

Returns this value as a `Calendar`

---

getDateValue() Method

public Date getDateValue()

Returns this value as a `Date`

---

getGDateValue() Method

public GDate getGDateValue()

Returns this value as a `GDate`

---

set(Calendar) Method

DEPRECATED replaced with `com.bea.xml.XmlDateTime.setCalendarValue(java.util.Calendar)`

public void set(Calendar *c*)

Sets this value as a `Calendar`

---

set(GDateSpecification) Method

DEPRECATED replaced with `com.bea.xml.XmlDateTime.setGDateValue(com.bea.xml.GDate)`

public void set(GDateSpecification *gd*)

Sets this value as a `GDateSpecification`

---

set(Date) Method

DEPRECATED replaced with `com.bea.xml.XmlDateTime.setDateValue(java.util.Date)`

public void set(Date *d*)

Sets this value as a `Date`

---

setCalendarValue(Calendar) Method

public void setCalendarValue(Calendar *c*)

Sets this value as a `Calendar`

---

setDateValue(Date) Method

public void setDateValue(Date *d*)

Sets this value as a `Date`

---

setGDateValue(GDate) Method

public void setGDateValue(GDate *gd*)

Sets this value as a `GDateSpecification`

com.bea.xml

XmlDateTime.Factory Class

public static final class ***XmlDateTime.Factory***

extends `Object`

A class with methods for creating instances of `XmlDateTime`.

Hierarchy

```
Object
 XmlDateTime.Factory
```

Enclosing interface

`XmlDateTime`

Method Summary

`public static XmlDateTime` ***newInstance***()
Creates an empty instance of `XmlDateTime`

`public static XmlDateTime` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlDateTime`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlDateTime` ***newValue***(`Object` *obj*)
Creates an immutable `XmlDateTime` value

`public static XmlDateTime` ***parse***(`String` *s*)
Parses a `XmlDateTime` fragment from a String.

`public static XmlDateTime` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlDateTime` fragment from a String.

`public static XmlDateTime` ***parse***(`File` *f*)
Parses a `XmlDateTime` fragment from a File.

`public static XmlDateTime` ***parse***(*File f, XmlOptions options*)
Parses a `XmlDateTime` fragment from a File.

`public static XmlDateTime` ***parse***(*URL u*)
Parses a `XmlDateTime` fragment from a URL.

`public static XmlDateTime` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlDateTime` fragment from a URL.

`public static XmlDateTime` ***parse***(*InputStream is*)
Parses a `XmlDateTime` fragment from an InputStream.

`public static XmlDateTime` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlDateTime` fragment from an InputStream.

`public static XmlDateTime` ***parse***(*Reader r*)
Parses a `XmlDateTime` fragment from a Reader.

`public static XmlDateTime` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlDateTime` fragment from a Reader.

`public static XmlDateTime` ***parse***(*Node node*)
Parses a `XmlDateTime` fragment from a DOM Node.

`public static XmlDateTime` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlDateTime` fragment from a DOM Node.

`public static XmlDateTime` ***parse***(*XMLInputStream xis*)
Parses a `XmlDateTime` fragment from an XMLInputStream.

`public static XmlDateTime` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlDateTime` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlDateTime newInstance()`

Creates an empty instance of `XmlDateTime`

---

newInstance(XmlOptions) Method

```
public static XmlDateTime newInstance(XmlOptions options)
```

Creates an empty instance of `XmlDateTime`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlDateTime newValue(Object obj)
```

Creates an immutable `XmlDateTime` value

---

parse(String) Method

```
public static XmlDateTime parse(String s)
     throws XmlException
```

Parses a `XmlDateTime` fragment from a String. For example: "`<xml-fragment>2003-06-14T12:00:00</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlDateTime parse(String s,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlDateTime` fragment from a String. For example: "`<xml-fragment>2003-06-14T12:00:00</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlDateTime parse(File f)
     throws XmlException, IOException
```

Parses a `XmlDateTime` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlDateTime parse(File f,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDateTime` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlDateTime parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlDateTime` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlDateTime parse(URL u,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDateTime` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlDateTime parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlDateTime` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlDateTime parse(InputStream is,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDateTime` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlDateTime parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlDateTime` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlDateTime parse(Reader r,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDateTime` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlDateTime parse(Node node)
     throws XmlException
```

Parses a `XmlDateTime` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlDateTime parse(Node node,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlDateTime` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDateTime parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlDateTime` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDateTime parse(XMLInputStream xis,
                                XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlDateTime` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlDecimal Interface

public interface ***XmlDecimal***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:decimal type.

Convertible to `BigDecimal`.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlByte`, `XmlInt`, `XmlInteger`, `XmlLong`, `XmlNegativeInteger`, `XmlNonNegativeInteger`, `XmlNonPositiveInteger`, `XmlPositiveInteger`, `XmlShort`, `XmlUnsignedByte`, `XmlUnsignedInt`, `XmlUnsignedLong`, `XmlUnsignedShort`

Nested Class Summary

`public static final class`	**`XmlDecimal.Factory`** A class with methods for creating instances of `XmlDecimal`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public BigDecimal` ***`bigDecimalValue`***`()`
Returns this value as a `BigDecimal`

`public BigDecimal` ***`getBigDecimalValue`***`()`
Returns this value as a `BigDecimal`

`public void` ***`set`***`(BigDecimal bd)`
Sets this value as a `BigDecimal`

`public void` ***`setBigDecimalValue`***`(BigDecimal bd)`
Sets this value as a `BigDecimal`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

bigDecimalValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDecimal.getBigDecimalValue()`

`public BigDecimal bigDecimalValue()`

Returns this value as a `BigDecimal`

---

getBigDecimalValue() Method

`public BigDecimal getBigDecimalValue()`

Returns this value as a `BigDecimal`

---

set(BigDecimal) Method

DEPRECATED replaced with `com.bea.xml.XmlDecimal.setBigDecimalValue(java.math.BigDecimal)`

`public void set(BigDecimal` *`bd`***`)`**

Sets this value as a `BigDecimal`

---

setBigDecimalValue(BigDecimal) Method

`public void setBigDecimalValue(BigDecimal` *`bd`***`)`**

Sets this value as a `BigDecimal`

XmlDecimal.Factory Class

public static final class ***XmlDecimal.Factory***

extends `Object`

A class with methods for creating instances of `XmlDecimal`.

Hierarchy

```
Object
 XmlDecimal.Factory
```

Enclosing interface

`XmlDecimal`

Method Summary

`public static XmlDecimal` ***newInstance***()
Creates an empty instance of `XmlDecimal`

`public static XmlDecimal` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlDecimal`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlDecimal` ***newValue***(`Object` *obj*)
Creates an immutable `XmlDecimal` value

`public static XmlDecimal` ***parse***(`String` *s*)
Parses a `XmlDecimal` fragment from a String.

`public static XmlDecimal` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlDecimal` fragment from a String.

`public static XmlDecimal` ***parse***(`File` *f*)
Parses a `XmlDecimal` fragment from a File.

`public static XmlDecimal` ***parse***(*File f, XmlOptions options*)
Parses a `XmlDecimal` fragment from a File.

`public static XmlDecimal` ***parse***(*URL u*)
Parses a `XmlDecimal` fragment from a URL.

`public static XmlDecimal` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlDecimal` fragment from a URL.

`public static XmlDecimal` ***parse***(*InputStream is*)
Parses a `XmlDecimal` fragment from an InputStream.

`public static XmlDecimal` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlDecimal` fragment from an InputStream.

`public static XmlDecimal` ***parse***(*Reader r*)
Parses a `XmlDecimal` fragment from a Reader.

`public static XmlDecimal` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlDecimal` fragment from a Reader.

`public static XmlDecimal` ***parse***(*Node node*)
Parses a `XmlDecimal` fragment from a DOM Node.

`public static XmlDecimal` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlDecimal` fragment from a DOM Node.

`public static XmlDecimal` ***parse***(*XMLInputStream xis*)
Parses a `XmlDecimal` fragment from an XMLInputStream.

`public static XmlDecimal` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlDecimal` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlDecimal newInstance()`

Creates an empty instance of `XmlDecimal`

---

newInstance(XmlOptions) Method

```
public static XmlDecimal newInstance(XmlOptions options)
```

Creates an empty instance of `XmlDecimal`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlDecimal newValue(Object obj)
```

Creates an immutable `XmlDecimal` value

---

parse(String) Method

```
public static XmlDecimal parse(String s)
     throws XmlException
```

Parses a `XmlDecimal` fragment from a String. For example: "`<xml-fragment>1234.56789</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlDecimal parse(String s,
                               XmlOptions options)
     throws XmlException
```

Parses a `XmlDecimal` fragment from a String. For example: "`<xml-fragment>1234.56789</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlDecimal parse(File f)
     throws XmlException, IOException
```

Parses a `XmlDecimal` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlDecimal parse(File f,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDecimal` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlDecimal parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlDecimal` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlDecimal parse(URL u,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDecimal` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlDecimal parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlDecimal` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlDecimal parse(InputStream is,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDecimal` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlDecimal parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlDecimal` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlDecimal parse(Reader r,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDecimal` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlDecimal parse(Node node)
     throws XmlException
```

Parses a `XmlDecimal` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlDecimal parse(Node node,
                               XmlOptions options)
     throws XmlException
```

Parses a `XmlDecimal` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDecimal parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlDecimal` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDecimal parse(XMLInputStream xis,
                               XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlDecimal` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlDocumentProperties Class

public abstract class ***XmlDocumentProperties***

extends `Object`

This class is used to attach arbitrary information to an XML document. It also defines several well−known types of information that can be attached or found on a document.

For example, suppose you wanted to associate a filename with the document containing an xml object "xobj". This could be done via the following code:

tokenSource.documentProperties().set(XmlDocumentProperties.NAME, "MyFilename.xml");

To fetch the filename later, given an xobj2 anywhere in the same document, you can write:

filename = (String)tokenSource.documentProperties().get(XmlDocumentProperties.NAME);

Hierarchy

```
Object
 XmlDocumentProperties
```

Field Summary

public static final Object	***DOCTYPE_NAME*** Doc type name
public static final Object	***DOCTYPE_PUBLIC_ID*** Doc type public id
public static final Object	***DOCTYPE_SYSTEM_ID*** Doc type system id
public static final Object	***ENCODING*** Document encoding
public static final Object	***MESSAGE_DIGEST*** SHA message digest
public static final Object	***SOURCE_NAME*** Used to store the original name (a String) for the source from which the XML document was loaded.

`public static final Object` ***VERSION***
Document version

Constructor Summary

XmlDocumentProperties()

Method Summary

`public abstract Object` ***get****(Object key)*
Returns a value previously attached to a document using set.

`public String` ***getDoctypeName()***
Returns the DOCTYPE name used in the DOCTYPE> declaration.

`public String` ***getDoctypePublicId()***
Returns the DOCTYPE public ID used in the DOCTYPE> declaration.

`public String` ***getDoctypeSystemId()***
Returns the DOCTYPE system ID used in the DOCTYPE> declaration.

`public String` ***getEncoding()***
Returns the encoding used for the XML document, as an ISO encoding name.

`public byte[]` ***getMessageDigest()***
Returns the message digest used to summarize the document.

`public String` ***getSourceName()***
Returns the name of the XML document file.

`public String` ***getVersion()***
Returns the XML version string used in the xml?> declaration.

`public abstract Object` ***put****(Object key, Object value)*
Attaches a value to the root of the document containing the given token source.

`public abstract Object` ***remove****(Object key)*
Removes a value previously attached to a document using set.

`public void` ***setDoctypeName****(String doctypename)*
Sets the DOCTYPE name use in the DOCTYPE> declaration.

`public void` ***setDoctypePublicId***(`String` *publicid*)
Sets the DOCTYPE public ID to use in the DOCTYPE> declaration.

`public void` ***setDoctypeSystemId***(`String` *systemid*)
Sets the DOCTYPE system ID to use in the DOCTYPE> declaration.

`public void` ***setEncoding***(`String` *encoding*)
Sets the encoding to use for the XML document.

`public void` ***setMessageDigest***(*byte[] digest*)
Sets the message digest used to summarize the document.

`public void` ***setSourceName***(`String` *sourceName*)
Sets the name of the XML document file.

`public void` ***setVersion***(`String` *version*)
Sets the XML version string to use in the xml?> declaration.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

DOCTYPE_NAME

`public static final Object DOCTYPE_NAME`

Doc type name

DOCTYPE_PUBLIC_ID

`public static final Object DOCTYPE_PUBLIC_ID`

Doc type public id

DOCTYPE_SYSTEM_ID

`public static final Object DOCTYPE_SYSTEM_ID`

Doc type system id

ENCODING

`public static final Object ENCODING`

Document encoding

MESSAGE_DIGEST

`public static final Object MESSAGE_DIGEST`

SHA message digest

SOURCE_NAME

`public static final Object SOURCE_NAME`

Used to store the original name (a String) for the source from which the XML document was loaded. This name, if present, is used to identify the document when reporting validation or comilation errors. XmlObject.Factory.parse(File) and SchemaTypeLoader.loadInstance(File) both automatically set this value to the filename.

VERSION

`public static final Object VERSION`

Document version

Constructor Detail

XmlDocumentProperties

`public XmlDocumentProperties()`

Method Detail

get(Object) Method

`public abstract Object get(Object key)`

Returns a value previously attached to a document using set.

Parameters

key
The key: this is the key that was previously passed to set to store the value.

Returns

The saved value, or null if none is found.

---

getDoctypeName() Method

`public String getDoctypeName()`

Returns the DOCTYPE name used in the DOCTYPE> declaration.

---

getDoctypePublicId() Method

`public String getDoctypePublicId()`

Returns the DOCTYPE public ID used in the DOCTYPE> declaration.

---

getDoctypeSystemId() Method

`public String getDoctypeSystemId()`

Returns the DOCTYPE system ID used in the DOCTYPE> declaration.

---

getEncoding() Method

`public String getEncoding()`

Returns the encoding used for the XML document, as an ISO encoding name.

Related Topics

`XmlOptions.setCharacterEncoding(String)`

---

getMessageDigest() Method

`public byte[] getMessageDigest()`

Returns the message digest used to summarize the document.

Related Topics

`XmlOptions.setLoadMessageDigest()`

---

getSourceName() Method

```
public String getSourceName()
```

Returns the name of the XML document file. Typically a URL, but may be any String.

Related Topics

`XmlOptions.setDocumentSourceName(String)`

---

getVersion() Method

```
public String getVersion()
```

Returns the XML version string used in the xml?> declaration.

---

put(Object, Object) Method

```
public abstract Object put(Object key,
                           Object value)
```

Attaches a value to the root of the document containing the given token source.

Parameters

key
: The key: there can be one value for each key.

value
: The value to attach to the document.

---

remove(Object) Method

```
public abstract Object remove(Object key)
```

Removes a value previously attached to a document using set.

Parameters

key
: The key: this is the key that was previously passed to set to store the value.

---

setDoctypeName(String) Method

`public void setDoctypeName(String` *`doctypename`***`)`**

Sets the DOCTYPE name use in the DOCTYPE> declaration.

Parameters

doctypename
the doctypename

---

setDoctypePublicId(String) Method

`public void setDoctypePublicId(String` *`publicid`***`)`**

Sets the DOCTYPE public ID to use in the DOCTYPE> declaration.

Parameters

publicid
the public ID

---

setDoctypeSystemId(String) Method

`public void setDoctypeSystemId(String` *`systemid`***`)`**

Sets the DOCTYPE system ID to use in the DOCTYPE> declaration.

Parameters

systemid
the system ID

---

setEncoding(String) Method

`public void setEncoding(String` *`encoding`***`)`**

Sets the encoding to use for the XML document. Should be a valid XML encoding string.

Parameters

encoding
the ISO encoding name

Related Topics

`XmlOptions.setCharacterEncoding(String)`

setMessageDigest(byte[]) Method

public void setMessageDigest(byte[] *digest***)**

Sets the message digest used to summarize the document.

Parameters

digest
the bytes of the digest

Related Topics

`XmlOptions.setLoadMessageDigest()`

setSourceName(String) Method

public void setSourceName(String *sourceName***)**

Sets the name of the XML document file. Typically a URL, but may be any String.

Parameters

sourceName
the name to set

Related Topics

`XmlOptions.setDocumentSourceName(String)`

setVersion(String) Method

public void setVersion(String *version***)**

Sets the XML version string to use in the xml?> declaration. (The XML specification is quite stable at "1.0".)

Parameters

version
the XML version string

com.bea.xml

XmlDouble Interface

public interface ***XmlDouble***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:double type.

Naturally, convertible to a Java double.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlDouble.Factory`** A class with methods for creating instances of `XmlDouble`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public double` ***doubleValue***()
Returns this value as a double

`public double` ***getDoubleValue***()
Returns this value as a double

`public void` ***set***(*double v*)
Sets this value as a double

`public void` ***setDoubleValue***(*double v*)
Sets this value as a double

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

doubleValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDouble.getDoubleValue()`

`public double doubleValue()`

Returns this value as a double

---

getDoubleValue() Method

`public double getDoubleValue()`

Returns this value as a double

---

set(double) Method

DEPRECATED replaced with `com.bea.xml.XmlDouble.setDoubleValue(double)`

`public void set(double v)`

Sets this value as a double

---

setDoubleValue(double) Method

`public void setDoubleValue(double v)`

Sets this value as a double

com.bea.xml

XmlDouble.Factory Class

public static final class ***XmlDouble.Factory***

extends `Object`

A class with methods for creating instances of `XmlDouble`.

Hierarchy

```
Object
 XmlDouble.Factory
```

Enclosing interface

`XmlDouble`

Method Summary

`public static XmlDouble` ***newInstance***()
Creates an empty instance of `XmlDouble`

`public static XmlDouble` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlDouble`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlDouble` ***newValue***(`Object` *obj*)
Creates an immutable `XmlDouble` value

`public static XmlDouble` ***parse***(`String` *s*)
Parses a `XmlDouble` fragment from a String.

`public static XmlDouble` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlDouble` fragment from a String.

`public static XmlDouble` ***parse***(`File` *f*)
Parses a `XmlDouble` fragment from a File.

`public static XmlDouble` ***parse***(*File f, XmlOptions options*)
Parses a `XmlDouble` fragment from a File.

`public static XmlDouble` ***parse***(*URL u*)
Parses a `XmlDouble` fragment from a URL.

`public static XmlDouble` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlDouble` fragment from a URL.

`public static XmlDouble` ***parse***(*InputStream is*)
Parses a `XmlDouble` fragment from an InputStream.

`public static XmlDouble` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlDouble` fragment from an InputStream.

`public static XmlDouble` ***parse***(*Reader r*)
Parses a `XmlDouble` fragment from a Reader.

`public static XmlDouble` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlDouble` fragment from a Reader.

`public static XmlDouble` ***parse***(*Node node*)
Parses a `XmlDouble` fragment from a DOM Node.

`public static XmlDouble` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlDouble` fragment from a DOM Node.

`public static XmlDouble` ***parse***(*XMLInputStream xis*)
Parses a `XmlDouble` fragment from an XMLInputStream.

`public static XmlDouble` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlDouble` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlDouble newInstance()`

Creates an empty instance of `XmlDouble`

---

newInstance(XmlOptions) Method

```
public static XmlDouble newInstance(XmlOptions options)
```

Creates an empty instance of `XmlDouble`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlDouble newValue(Object obj)
```

Creates an immutable `XmlDouble` value

---

parse(String) Method

```
public static XmlDouble parse(String s)
     throws XmlException
```

Parses a `XmlDouble` fragment from a String. For example: "`<xml-fragment>123.34e+57</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlDouble parse(String s,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlDouble` fragment from a String. For example: "`<xml-fragment>123.34e+57</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlDouble parse(File f)
     throws XmlException, IOException
```

Parses a `XmlDouble` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlDouble parse(File f,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDouble` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlDouble parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlDouble` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlDouble parse(URL u,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDouble` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlDouble parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlDouble` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlDouble parse(InputStream is,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDouble` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlDouble parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlDouble` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlDouble parse(Reader r,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDouble` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlDouble parse(Node node)
     throws XmlException
```

Parses a `XmlDouble` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlDouble parse(Node node,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlDouble` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDouble parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlDouble` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDouble parse(XMLInputStream xis,
                              XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlDouble` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlDuration Interface

public interface ***XmlDuration***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:duration type.

Convertible to a `GDuration`.

Related Topics

`GDuration`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlDuration.Factory`** A class with methods for creating instances of `XmlDuration`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public GDuration` ***gDurationValue()***
Returns this value as a `GDuration`

`public GDuration` ***getGDurationValue()***
Returns this value as a `GDuration`

`public void` ***set***`(GDurationSpecification gd)`
Sets this value as a `GDuration`

`public void` ***setGDurationValue***`(GDuration gd)`
Sets this value as a `GDuration`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

gDurationValue() Method

DEPRECATED replaced with `com.bea.xml.XmlDuration.getGDurationValue()`

`public GDuration gDurationValue()`

Returns this value as a `GDuration`

---

getGDurationValue() Method

`public GDuration getGDurationValue()`

Returns this value as a `GDuration`

---

set(GDurationSpecification) Method

DEPRECATED replaced with `com.bea.xml.XmlDuration.setGDurationValue(com.bea.xml.GDuration)`

`public void set(GDurationSpecification` *`gd`***`)`**

Sets this value as a `GDuration`

---

setGDurationValue(GDuration) Method

`public void setGDurationValue(GDuration` *`gd`***`)`**

Sets this value as a `GDuration`

XmlDuration.Factory Class

public static final class ***XmlDuration.Factory***

extends `Object`

A class with methods for creating instances of `XmlDuration`.

Hierarchy

```
Object
 XmlDuration.Factory
```

Enclosing interface

`XmlDuration`

Method Summary

`public static XmlDuration` ***newInstance***()
Creates an empty instance of `XmlDuration`

`public static XmlDuration` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlDuration`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlDuration` ***newValue***(`Object` *obj*)
Creates an immutable `XmlDuration` value

`public static XmlDuration` ***parse***(`String` *s*)
Parses a `XmlDuration` fragment from a String.

`public static XmlDuration` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlDuration` fragment from a String.

`public static XmlDuration` ***parse***(`File` *f*)
Parses a `XmlDuration` fragment from a File.

`public static XmlDuration` ***parse***(*File f, XmlOptions options*)
Parses a `XmlDuration` fragment from a File.

`public static XmlDuration` ***parse***(*URL u*)
Parses a `XmlDuration` fragment from a URL.

`public static XmlDuration` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlDuration` fragment from a URL.

`public static XmlDuration` ***parse***(*InputStream is*)
Parses a `XmlDuration` fragment from an InputStream.

`public static XmlDuration` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlDuration` fragment from an InputStream.

`public static XmlDuration` ***parse***(*Reader r*)
Parses a `XmlDuration` fragment from a Reader.

`public static XmlDuration` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlDuration` fragment from a Reader.

`public static XmlDuration` ***parse***(*Node node*)
Parses a `XmlDuration` fragment from a DOM Node.

`public static XmlDuration` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlDuration` fragment from a DOM Node.

`public static XmlDuration` ***parse***(*XMLInputStream xis*)
Parses a `XmlDuration` fragment from an XMLInputStream.

`public static XmlDuration` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlDuration` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlDuration newInstance()`

Creates an empty instance of `XmlDuration`

---

newInstance(XmlOptions) Method

```
public static XmlDuration newInstance(XmlOptions options)
```

Creates an empty instance of `XmlDuration`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlDuration newValue(Object obj)
```

Creates an immutable `XmlDuration` value

---

parse(String) Method

```
public static XmlDuration parse(String s)
     throws XmlException
```

Parses a `XmlDuration` fragment from a String. For example: "`<xml-fragment>P1Y2MT2H</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlDuration parse(String s,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlDuration` fragment from a String. For example: "`<xml-fragment>P1Y2MT2H</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlDuration parse(File f)
     throws XmlException, IOException
```

Parses a `XmlDuration` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlDuration parse(File f,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDuration` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlDuration parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlDuration` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlDuration parse(URL u,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDuration` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlDuration parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlDuration` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlDuration parse(InputStream is,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDuration` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlDuration parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlDuration` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlDuration parse(Reader r,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlDuration` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlDuration parse(Node node)
     throws XmlException
```

Parses a `XmlDuration` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlDuration parse(Node node,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlDuration` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDuration parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlDuration` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlDuration parse(XMLInputStream xis,
                                XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlDuration` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlENTITIES Interface

public interface ***XmlENTITIES***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:ENTITIES type, a list type.

A list type.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlENTITIES.Factory`** A class with methods for creating instances of `XmlENTITIES`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

public
List ***getListValue()***
Returns the value as a `List` of `String` values

public
List ***listValue()***
Returns the value as a `List` of `String` values

public
void ***set(List l)***
Sets the value as a `List`

public
void ***setListValue(List l)***
Sets the value as a `List`

public
List ***xgetListValue()***
Returns the value as a `List` of `XmlENTITY` values

public
List ***xlistValue()***
Returns the value as a `List` of `XmlENTITY` values

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getListValue() Method

`public List getListValue()`

Returns the value as a `List` of `String` values

---

listValue() Method

DEPRECATED replaced by `com.bea.xml.XmlENTITIES.getListValue()`

`public List listValue()`

Returns the value as a `List` of `String` values

---

set(List) Method

DEPRECATED replaced by `com.bea.xml.XmlENTITIES.getListValue()`

`public void set(List l)`

Sets the value as a `List`

---

setListValue(List) Method

`public void setListValue(List l)`

Sets the value as a `List`

---

xgetListValue() Method

`public List xgetListValue()`

Returns the value as a `List` of `XmlENTITY` values

---

xlistValue() Method

DEPRECATED replaced by `com.bea.xml.XmlENTITIES.getListValue()`

`public List xlistValue()`

Returns the value as a `List` of `XmlENTITY` values

XmlENTITIES.Factory Class

public static final class ***XmlENTITIES.Factory***

extends `Object`

A class with methods for creating instances of `XmlENTITIES`.

Hierarchy

```
Object
 XmlENTITIES.Factory
```

Enclosing interface

`XmlENTITIES`

Method Summary

`public static XmlENTITIES` ***newInstance***()
Creates an empty instance of `XmlENTITIES`

`public static XmlENTITIES` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlENTITIES`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlENTITIES` ***newValue***(`Object` *obj*)
Creates an immutable `XmlENTITIES` value

`public static XmlENTITIES` ***parse***(`String` *s*)
Parses a `XmlENTITIES` fragment from a String.

`public static XmlENTITIES` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlENTITIES` fragment from a String.

`public static XmlENTITIES` ***parse***(`File` *f*)
Parses a `XmlENTITIES` fragment from a File.

`public static XmlENTITIES` ***parse***(*File f, XmlOptions options*)
Parses a `XmlENTITIES` fragment from a File.

`public static XmlENTITIES` ***parse***(*URL u*)
Parses a `XmlENTITIES` fragment from a URL.

`public static XmlENTITIES` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlENTITIES` fragment from a URL.

`public static XmlENTITIES` ***parse***(*InputStream is*)
Parses a `XmlENTITIES` fragment from an InputStream.

`public static XmlENTITIES` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlENTITIES` fragment from an InputStream.

`public static XmlENTITIES` ***parse***(*Reader r*)
Parses a `XmlENTITIES` fragment from a Reader.

`public static XmlENTITIES` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlENTITIES` fragment from a Reader.

`public static XmlENTITIES` ***parse***(*Node node*)
Parses a `XmlENTITIES` fragment from a DOM Node.

`public static XmlENTITIES` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlENTITIES` fragment from a DOM Node.

`public static XmlENTITIES` ***parse***(*XMLInputStream xis*)
Parses a `XmlENTITIES` fragment from an XMLInputStream.

`public static XmlENTITIES` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlENTITIES` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlENTITIES newInstance()`

Creates an empty instance of `XmlENTITIES`

---

newInstance(XmlOptions) Method

```
public static XmlENTITIES newInstance(XmlOptions options)
```

Creates an empty instance of `XmlENTITIES`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlENTITIES newValue(Object obj)
```

Creates an immutable `XmlENTITIES` value

---

parse(String) Method

```
public static XmlENTITIES parse(String s)
     throws XmlException
```

Parses a `XmlENTITIES` fragment from a String.

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlENTITIES parse(String s,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlENTITIES` fragment from a String.

Exceptions

XmlException

---

parse(File) Method

```
public static XmlENTITIES parse(File f)
     throws XmlException, IOException
```

Parses a `XmlENTITIES` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlENTITIES parse(File f,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlENTITIES` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlENTITIES parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlENTITIES` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlENTITIES parse(URL u,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlENTITIES` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlENTITIES parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlENTITIES` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlENTITIES parse(InputStream is,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlENTITIES` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlENTITIES parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlENTITIES` fragment from a Reader.

Exceptions

`XmlException`
`IOException`

---

parse(Reader, XmlOptions) Method

```
public static XmlENTITIES parse(Reader r,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlENTITIES` fragment from a Reader.

Exceptions

`XmlException`
`IOException`

---

parse(Node) Method

```
public static XmlENTITIES parse(Node node)
     throws XmlException
```

Parses a `XmlENTITIES` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(Node, XmlOptions) Method

```
public static XmlENTITIES parse(Node node,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlENTITIES` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlENTITIES parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlENTITIES` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlENTITIES parse(XMLInputStream xis,
                                XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlENTITIES` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlENTITY Interface

public interface ***XmlENTITY***

extends `XmlNCName`

Corresponds to the XML Schema xs:ENTITY type.

Convertible to `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlName`, `XmlNCName`, `XmlNormalizedString`, `XmlObject`, `XmlString`, `XmlToken`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlENTITY.Factory`** A class with methods for creating instances of `XmlENTITY`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlName`

`XmlName.Factory`

Nested classes from interface com.bea.xml.`XmlNCName`

`XmlNCName.Factory`

Nested classes from interface com.bea.xml.`XmlNormalizedString`

`XmlNormalizedString.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Nested classes from interface com.bea.xml.`XmlToken`

`XmlToken.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlName`

`type`

Fields from interface com.bea.xml.`XmlNCName`

`type`

Fields from interface com.bea.xml.`XmlNormalizedString`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Fields from interface com.bea.xml.`XmlToken`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil,`

`toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlENTITY.Factory Class

public static final class ***XmlENTITY.Factory***

extends `Object`

A class with methods for creating instances of `XmlENTITY`.

Hierarchy

```
Object
 XmlENTITY.Factory
```

Enclosing interface

`XmlENTITY`

Method Summary

`public static XmlENTITY` ***newInstance***()
Creates an empty instance of `XmlENTITY`

`public static XmlENTITY` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlENTITY`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlENTITY` ***newValue***(`Object` *obj*)
Creates an immutable `XmlENTITY` value

`public static XmlENTITY` ***parse***(`String` *s*)
Parses a `XmlENTITY` fragment from a String.

`public static XmlENTITY` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlENTITY` fragment from a String.

`public static XmlENTITY` ***parse***(`File` *f*)
Parses a `XmlENTITY` fragment from a File.

`public static XmlENTITY` ***parse***(*File f, XmlOptions options*)
Parses a `XmlENTITY` fragment from a File.

`public static XmlENTITY` ***parse***(*URL u*)
Parses a `XmlENTITY` fragment from a URL.

`public static XmlENTITY` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlENTITY` fragment from a URL.

`public static XmlENTITY` ***parse***(*InputStream is*)
Parses a `XmlENTITY` fragment from an InputStream.

`public static XmlENTITY` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlENTITY` fragment from an InputStream.

`public static XmlENTITY` ***parse***(*Reader r*)
Parses a `XmlENTITY` fragment from a Reader.

`public static XmlENTITY` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlENTITY` fragment from a Reader.

`public static XmlENTITY` ***parse***(*Node node*)
Parses a `XmlENTITY` fragment from a DOM Node.

`public static XmlENTITY` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlENTITY` fragment from a DOM Node.

`public static XmlENTITY` ***parse***(*XMLInputStream xis*)
Parses a `XmlENTITY` fragment from an XMLInputStream.

`public static XmlENTITY` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlENTITY` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlENTITY newInstance()`

Creates an empty instance of `XmlENTITY`

---

newInstance(XmlOptions) Method

```
public static XmlENTITY newInstance(XmlOptions options)
```

Creates an empty instance of `XmlENTITY`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlENTITY newValue(Object obj)
```

Creates an immutable `XmlENTITY` value

---

parse(String) Method

```
public static XmlENTITY parse(String s)
     throws XmlException
```

Parses a `XmlENTITY` fragment from a String.

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlENTITY parse(String s,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlENTITY` fragment from a String.

Exceptions

XmlException

---

parse(File) Method

```
public static XmlENTITY parse(File f)
     throws XmlException, IOException
```

Parses a `XmlENTITY` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlENTITY parse(File f,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlENTITY` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlENTITY parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlENTITY` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlENTITY parse(URL u,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlENTITY` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlENTITY parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlENTITY` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlENTITY parse(InputStream is,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlENTITY` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlENTITY parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlENTITY` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlENTITY parse(Reader r,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlENTITY` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlENTITY parse(Node node)
     throws XmlException
```

Parses a `XmlENTITY` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlENTITY parse(Node node,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlENTITY` fragment from a DOM Node.

Exceptions

XmlException

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlENTITY parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlENTITY` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlENTITY parse(XMLInputStream xis,
                              XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlENTITY` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlError Class

public class ***XmlError***

extends `Object`
implements `Serializable`

Represents a message at a specific XML location.

The message can be an error, warning, or simple information, and it may optionally be associated with a specific location in an XML document. The class includes methods for extracting the location as a line number, XmlCursor, or XmlObject, as well as for obtaining and message and severity of the error.

Related Topics

`XmlOptions.setErrorListener(Collection)`
`XmlException`

Hierarchy

```
Object
 XmlError
```

All Implemented Interfaces

```
Serializable
```

Field Summary

`public static final int`	***SEVERITY_ERROR*** An error.
`public static final int`	***SEVERITY_INFO*** An informational message.
`public static final int`	***SEVERITY_WARNING*** A warning.

Constructor Summary

XmlError*(`XmlError` src)*

Copy constructor.

Method Summary

public static XmlError	**forCursor**(String message, XmlCursor cursor) Returns an XmlError for the given message, located at the XmlCursor, with XmlError.SEVERITY_ERROR.
public static XmlError	**forCursor**(String message, int severity, XmlCursor cursor) Returns an XmlError for the given message, with the given severity, located at the XmlCursor.
public static XmlError	**forLocation**(String message, String sourceName, int line, int column, int offset) Returns an XmlError for the given message, located at a specific point in the given file and XmlError.SEVERITY_ERROR.
public static XmlError	**forLocation**(String message, int severity, String sourceName, int line, int column, int offset) Returns an XmlError for the given message, with the given severity, located at a specific point in the given file.
public static XmlError	**forLocationAndCursor**(String message, int severity, String sourceName, int line, int column, int offset, XmlCursor cursor) Returns an XmlError for the given message, with the given severity, located at the given physcial location and XmlCursor.
protected static String	**formattedFileName**(String rawString, URI base) Tries to produce a nicely formatted filename from the given string.
public static XmlError	**forMessage**(String message) Returns an XmlError for the given message, with no location and XmlError.SEVERITY_ERROR.
public static XmlError	**forMessage**(String message, int severity) Returns an XmlError for the given message, with no location and the given severity.
public static XmlError	**forObject**(String message, XmlObject xobj) Returns an XmlError for the given message, located at the XmlObject, with XmlError.SEVERITY_ERROR.
public static XmlError	**forObject**(String message, int severity, XmlObject xobj) Returns an XmlError for the given message, with the given severity, located at the XmlObject.
public static	**forSource**(String message, String sourceName)

`XmlError` Returns an XmlError for the given message, located in the given file and `XmlError.SEVERITY_ERROR`.

`public static` ***forSource***(`String` *message, int severity,* `String` *sourceName)*
`XmlError` Returns an XmlError for the given message, with the given severity, located in the given file.

`public int` ***getColumn***()
Returns the column number of the error, if available, −1 if not.

`public XmlCursor` ***getCursorLocation***()
Returns a location of the error as an `XmlCursor`, null if not available.

`public int` ***getLine***()
Returns the line number of the error, if available, −1 if not.

`public Object` ***getLocation***(`Object` *type)*
Returns a location object of the given type.

`public String` ***getMessage***()
Returns the error message without location information.

`public XmlObject` ***getObjectLocation***()
Returns a location of the error as an `XmlObject`, null if not available.

`public int` ***getOffset***()
Returns the file character offset of the error, if available, −1 if not.

`public int` ***getSeverity***()
Returns the severity.

`public String` ***getSourceName***()
Returns the URL (or other name) of the file with the error, if available.

`public String` ***toString***()
Produces a standard string for the error message, complete with filename and location offsets if available.

`public String` ***toString***(`URI` *base)*
Produces a standard string with the error message.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

SEVERITY_ERROR

public static final int SEVERITY_ERROR

An error. See `XmlError.getSeverity()`.

SEVERITY_INFO

public static final int SEVERITY_INFO

An informational message. See `XmlError.getSeverity()`.

SEVERITY_WARNING

public static final int SEVERITY_WARNING

A warning. See `XmlError.getSeverity()`.

Constructor Detail

XmlError

public XmlError(XmlError *src*)

Copy constructor.

Method Detail

forCursor(String, XmlCursor) Method

public static XmlError forCursor(String *message*,
XmlCursor *cursor*)

Returns an XmlError for the given message, located at the XmlCursor, with `XmlError.SEVERITY_ERROR`.

Parameters

message
the error message

cursor
the XmlCursor representing the location of the error

---

forCursor(String, int, XmlCursor) Method

```
public static XmlError forCursor(String message,
                                 int severity,
                                 XmlCursor cursor)
```

Returns an XmlError for the given message, with the given severity, located at the XmlCursor.

Parameters

message
the error message

severity
the severity (`XmlError.SEVERITY_ERROR`, `XmlError.SEVERITY_WARNING`, or `XmlError.SEVERITY_INFO`)

cursor
the XmlCursor representing the location of the error

---

forLocation(String, String, int, int, int) Method

```
public static XmlError forLocation(String message,
                                   String sourceName,
                                   int line,
                                   int column,
                                   int offset)
```

Returns an XmlError for the given message, located at a specific point in the given file and `XmlError.SEVERITY_ERROR`.

Parameters

message
the error message

sourceName
the URL or other name for the file

line
the 1−based line number, or −1 if not known

column
the 1−based column number, or −1 if not known

offset
the 0−base file character offset, or −1 if not known

forLocation(String, int, String, int, int, int) Method

```
public static XmlError forLocation(String message,
                                   int severity,
                                   String sourceName,
                                   int line,
                                   int column,
                                   int offset)
```

Returns an XmlError for the given message, with the given severity, located at a specific point in the given file.

Parameters

message
: the error message

severity
: the severity (`XmlError.SEVERITY_ERROR`, `XmlError.SEVERITY_WARNING`, or `XmlError.SEVERITY_INFO`)

sourceName
: the URL or other name for the file

line
: the 1−based line number, or −1 if not known

column
: the 1−based column number, or −1 if not known

offset
: the 0−base file character offset, or −1 if not known

forLocationAndCursor(String, int, String, int, int, int, XmlCursor) Method

```
public static XmlError forLocationAndCursor(String message,
                                            int severity,
                                            String sourceName,
                                            int line,
                                            int column,
                                            int offset,
                                            XmlCursor cursor)
```

Returns an XmlError for the given message, with the given severity, located at the given physcial location and XmlCursor.

Parameters

message
the error message

severity
the severity (`XmlError.SEVERITY_ERROR`, `XmlError.SEVERITY_WARNING`, or `XmlError.SEVERITY_INFO`)

sourceName
the URL or other name for the file

line
the 1−based line number, or −1 if not known

column
the 1−based column number, or −1 if not known

offset
the 0−base file character offset, or −1 if not known

cursor
the XmlCursor representing the location of the error

---

formattedFileName(String, URI) Method

```
protected static String formattedFileName(String rawString,
                                          URI base)
```

Tries to produce a nicely formatted filename from the given string.

---

forMessage(String) Method

```
public static XmlError forMessage(String message)
```

Returns an XmlError for the given message, with no location and `XmlError.SEVERITY_ERROR`.

Parameters

message
the error message

---

forMessage(String, int) Method

```
public static XmlError forMessage(String message,
                                  int severity)
```

Returns an XmlError for the given message, with no location and the given severity.

Parameters

message
the error message

severity
the severity (`XmlError.SEVERITY_ERROR`, `XmlError.SEVERITY_WARNING`, or `XmlError.SEVERITY_INFO`)

---

forObject(String, XmlObject) Method

```
public static XmlError forObject(String message,
                                 XmlObject xobj)
```

Returns an XmlError for the given message, located at the XmlObject, with `XmlError.SEVERITY_ERROR`.

Parameters

message
the error message

xobj
the XmlObject representing the location of the error

---

forObject(String, int, XmlObject) Method

```
public static XmlError forObject(String message,
                                 int severity,
                                 XmlObject xobj)
```

Returns an XmlError for the given message, with the given severity, located at the XmlObject.

Parameters

message
the error message

severity
the severity (`XmlError.SEVERITY_ERROR`, `XmlError.SEVERITY_WARNING`, or `XmlError.SEVERITY_INFO`)

xobj
the XmlObject representing the location of the error

---

forSource(String, String) Method

```
public static XmlError forSource(String message,
                                 String sourceName)
```

Returns an XmlError for the given message, located in the given file and `XmlError.SEVERITY_ERROR`.

Parameters

message
the error message

sourceName
the URL or other name for the file

---

forSource(String, int, String) Method

```
public static XmlError forSource(String message,
                                 int severity,
                                 String sourceName)
```

Returns an XmlError for the given message, with the given severity, located in the given file.

Parameters

message
the error message

severity
the severity (`XmlError.SEVERITY_ERROR`, `XmlError.SEVERITY_WARNING`, or `XmlError.SEVERITY_INFO`)

sourceName
the URL or other name for the file

---

getColumn() Method

```
public int getColumn()
```

Returns the column number of the error, if available, −1 if not.

---

getCursorLocation() Method

```
public XmlCursor getCursorLocation()
```

Returns a location of the error as an `XmlCursor`, null if not available.

---

getLine() Method

`public int getLine()`

Returns the line number of the error, if available, −1 if not.

---

getLocation(Object) Method

`public Object getLocation(Object type)`

Returns a location object of the given type. XmlCursor.class and XmlObject.class can be passed, for example. Null if not available.

---

getMessage() Method

`public String getMessage()`

Returns the error message without location information.

---

getObjectLocation() Method

`public XmlObject getObjectLocation()`

Returns a location of the error as an `XmlObject`, null if not available.

---

getOffset() Method

`public int getOffset()`

Returns the file character offset of the error, if available, −1 if not.

---

getSeverity() Method

`public int getSeverity()`

Returns the severity. Either `XmlError.SEVERITY_ERROR`, `XmlError.SEVERITY_WARNING`, or `XmlError.SEVERITY_INFO`.

---

getSourceName() Method

`public String getSourceName()`

Returns the URL (or other name) of the file with the error, if available.

---

toString() Method

`public String toString()`

Produces a standard string for the error message, complete with filename and location offsets if available.

Overrides

`Object.toString()`

---

toString(URI) Method

`public String toString(URI` *`base`***`)`**

Produces a standard string with the error message. If a non−null URI is supplied, source names are relativized against the given URI.

com.bea.xml

XmlException Class

public class ***XmlException***

extends `Exception`

A checked exception that can be thrown while processing, parsing, or compiling XML. May contain any number of `XmlError` objects.

Related Topics

`XmlError`
`XmlRuntimeException`

Hierarchy

```
Object
 Throwable
  Exception
   XmlException
```

All Implemented Interfaces

`Serializable`

Constructor Summary

XmlException*(`XmlError` error)*

Constructs an XmlException from an `XmlError`.

XmlException*(`XmlRuntimeException` xmlRuntimeException)*

Constructs an XmlException from an `XmlRuntimeException`.

XmlException*(`String` m, `Throwable` t, `XmlError` error)*

Constructs an XmlException from a message, a cause, and an `XmlError`.

XmlException*(`String` m, `Throwable` t, `Collection` errors)*

Constructs an XmlException from a message, a cause, and a collection of `XmlError`.

XmlException(`String` *m*, `Throwable` *t*)

Constructs an XmlException from a message and a cause.

XmlException(`String` *m*)

Constructs an XmlException from a message.

XmlException(`Throwable` *t*)

Constructs an XmlException from a cause.

Method Summary

`public XmlError` ***getError***()
Returns the first `XmlError` that caused the exception, if any.

`public Collection` ***getErrors***()
Returns the collection of `XmlError` that caused the exception, if any.

Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

XmlException

`public XmlException(XmlError` *`error`*`)`

Constructs an XmlException from an `XmlError`.

XmlException

`public XmlException(XmlRuntimeException` *`xmlRuntimeException`*`)`

Constructs an XmlException from an `XmlRuntimeException`.

XmlException

```
public XmlException(String m,
                    Throwable t,
                    XmlError error)
```

Constructs an XmlException from a message, a cause, and an `XmlError`.

XmlException

```
public XmlException(String m,
                    Throwable t,
                    Collection errors)
```

Constructs an XmlException from a message, a cause, and a collection of `XmlError`.

XmlException

```
public XmlException(String m,
                    Throwable t)
```

Constructs an XmlException from a message and a cause.

XmlException

```
public XmlException(String m)
```

Constructs an XmlException from a message.

XmlException

```
public XmlException(Throwable t)
```

Constructs an XmlException from a cause.

Method Detail

getError() Method

```
public XmlError getError()
```

Returns the first `XmlError` that caused the exception, if any.

---

getErrors() Method

`public Collection getErrors()`

Returns the collection of `XmlError` that caused the exception, if any.

XmlFactoryHook.ThreadContext Class

public static final class ***XmlFactoryHook.ThreadContext***

extends `Object`

Used to manage the XmlFactoryHook for the current thread.

Hierarchy

```
Object
 XmlFactoryHook.ThreadContext
```

Enclosing interface

`XmlFactoryHook`

Method Summary

`public static XmlFactoryHook` ***`getHook`****()*
Returns the current thread's hook, or null if none.

`public static void` ***`setHook`****(`XmlFactoryHook` hook)*
Sets the hook for the current thread.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

getHook() Method

`public static XmlFactoryHook getHook()`

Returns the current thread's hook, or null if none.

---

setHook(XmlFactoryHook) Method

`public static void setHook(XmlFactoryHook` *`hook`*`)`

Sets the hook for the current thread.

XmlFactoryHook Interface

public interface ***XmlFactoryHook***

A hook for the XML Bean Factory mechanism. Provided for advanced users who wish to provide their own implementation of the Factory.parse methods. This is used, for example, to defer reading XML streams until needed.

To use the hook, call XmlFactoryHook.ThreadContext.setHook(), passing your own XmlFactoryHook implementation. Then every call to a Factory method will be delgated to your hook.

```
MyHook hook = new MyHook();
XmlFactoryHook.ThreadContext.setHook(hook);
// this results in a call to hook.parse(...)
XmlObject.Factory.parse(new File("test.xml"));
```

If the hook needs to turn around and invoke the built−in parsers, then it should do so by calling the appropriate method on the passed SchemaTypeLoader. Since SchemaTypeLoader.parse() methods delegate to the registered hook, a hook that wishes to actually invoke the default parser without having itself called back again should unregister itself before calling loader.parse(), and then re−register itself again after the call.

```
void parse(SchemaTypeLoader loader, ...)
{
    XmlFactoryHook remember = XmlFactoryHook.ThreadContext.getHook();
    XmlFactoryHook.ThreadContext.setHook(null);
    loader.parse(...); // isn't hooked.
    XmlFactoryHook.ThreadContext.setHook(remember);
}
```

Nested Class Summary

`public static final class`	**`XmlFactoryHook.ThreadContext`** Used to manage the XmlFactoryHook for the current thread.

Method Summary

`public XmlObject`	***`newInstance`***`(`*`SchemaTypeLoader loader, SchemaType type, XmlOptions options)`* Hooks Factory.newInstance calls
`public XmlSaxHandler`	***`newXmlSaxHandler`***`(`*`SchemaTypeLoader loader, SchemaType type, XmlOptions options)`*

Hooks Factory.newXmlSaxHandler calls

public XmlObject ***parse***(*SchemaTypeLoader loader, String xmlText, SchemaType type, XmlOptions options*)

Hooks Factory.parse calls

public XmlObject ***parse***(*SchemaTypeLoader loader, InputStream jiois, SchemaType type, XmlOptions options*)

Hooks Factory.parse calls

public XmlObject ***parse***(*SchemaTypeLoader loader, Reader jior, SchemaType type, XmlOptions options*)

Hooks Factory.parse calls

public XmlObject ***parse***(*SchemaTypeLoader loader, Node node, SchemaType type, XmlOptions options*)

Hooks Factory.parse calls

public XmlObject ***parse***(*SchemaTypeLoader loader, XMLInputStream xis, SchemaType type, XmlOptions options*)

Hooks Factory.parse calls

Method Detail

newInstance(SchemaTypeLoader, SchemaType, XmlOptions) Method

```
public XmlObject newInstance(SchemaTypeLoader loader,
                             SchemaType type,
                             XmlOptions options)
```

Hooks Factory.newInstance calls

---

newXmlSaxHandler(SchemaTypeLoader, SchemaType, XmlOptions) Method

```
public XmlSaxHandler newXmlSaxHandler(SchemaTypeLoader loader,
                                      SchemaType type,
                                      XmlOptions options)
```

Hooks Factory.newXmlSaxHandler calls

---

parse(SchemaTypeLoader, String, SchemaType, XmlOptions) Method

```
public XmlObject parse(SchemaTypeLoader loader,
                       String xmlText,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException
```

Hooks Factory.parse calls

Exceptions

XmlException

---

parse(SchemaTypeLoader, InputStream, SchemaType, XmlOptions) Method

```
public XmlObject parse(SchemaTypeLoader loader,
                       InputStream jiois,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException, IOException
```

Hooks Factory.parse calls

Exceptions

XmlException
IOException

---

parse(SchemaTypeLoader, Reader, SchemaType, XmlOptions) Method

```
public XmlObject parse(SchemaTypeLoader loader,
                       Reader jior,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException, IOException
```

Hooks Factory.parse calls

Exceptions

XmlException
IOException

---

parse(SchemaTypeLoader, Node, SchemaType, XmlOptions) Method

```
public XmlObject parse(SchemaTypeLoader loader,
                       Node node,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException
```

Hooks Factory.parse calls

Exceptions

XmlException

---

parse(SchemaTypeLoader, XMLInputStream, SchemaType, XmlOptions) Method

```
public XmlObject parse(SchemaTypeLoader loader,
                       XMLInputStream xis,
                       SchemaType type,
                       XmlOptions options)
     throws XmlException, XMLStreamException
```

Hooks Factory.parse calls

Exceptions

XmlException
XMLStreamException

com.bea.xml

XmlFloat Interface

public interface ***XmlFloat***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:float type.

Naturally, convertible to a Java float.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlFloat.Factory`** A class with methods for creating instances of `XmlFloat`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public float` ***`floatValue`***`()`
Returns this value as a float

`public float` ***`getFloatValue`***`()`
Returns this value as a float

`public void` ***`set`***`(float v)`
Sets this value as a float

`public void` ***`setFloatValue`***`(float v)`
Sets this value as a float

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

floatValue() Method

DEPRECATED replaced by `com.bea.xml.XmlFloat.getFloatValue()`

`public float floatValue()`

Returns this value as a float

---

getFloatValue() Method

`public float getFloatValue()`

Returns this value as a float

---

set(float) Method

DEPRECATED replaced by `com.bea.xml.XmlFloat.setFloatValue(float)`

`public void set(float v)`

Sets this value as a float

---

setFloatValue(float) Method

`public void setFloatValue(float v)`

Sets this value as a float

XmlFloat.Factory Class

public static final class ***XmlFloat.Factory***

extends `Object`

A class with methods for creating instances of `XmlFloat`.

Hierarchy

```
Object
 XmlFloat.Factory
```

Enclosing interface

`XmlFloat`

Method Summary

`public static XmlFloat` ***newInstance***()
Creates an empty instance of `XmlFloat`

`public static XmlFloat` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlFloat`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlFloat` ***newValue***(`Object` *obj*)
Creates an immutable `XmlFloat` value

`public static XmlFloat` ***parse***(`String` *s*)
Parses a `XmlFloat` fragment from a String.

`public static XmlFloat` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlFloat` fragment from a String.

`public static XmlFloat` ***parse***(`File` *f*)
Parses a `XmlFloat` fragment from a File.

`public static XmlFloat` ***parse***(*File f, XmlOptions options*)
Parses a `XmlFloat` fragment from a File.

`public static XmlFloat` ***parse***(*URL u*)
Parses a `XmlFloat` fragment from a URL.

`public static XmlFloat` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlFloat` fragment from a URL.

`public static XmlFloat` ***parse***(*InputStream is*)
Parses a `XmlFloat` fragment from an InputStream.

`public static XmlFloat` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlFloat` fragment from an InputStream.

`public static XmlFloat` ***parse***(*Reader r*)
Parses a `XmlFloat` fragment from a Reader.

`public static XmlFloat` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlFloat` fragment from a Reader.

`public static XmlFloat` ***parse***(*Node node*)
Parses a `XmlFloat` fragment from a DOM Node.

`public static XmlFloat` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlFloat` fragment from a DOM Node.

`public static XmlFloat` ***parse***(*XMLInputStream xis*)
Parses a `XmlFloat` fragment from an XMLInputStream.

`public static XmlFloat` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlFloat` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlFloat newInstance()`

Creates an empty instance of `XmlFloat`

---

newInstance(XmlOptions) Method

```
public static XmlFloat newInstance(XmlOptions options)
```

Creates an empty instance of `XmlFloat`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlFloat newValue(Object obj)
```

Creates an immutable `XmlFloat` value

---

parse(String) Method

```
public static XmlFloat parse(String s)
     throws XmlException
```

Parses a `XmlFloat` fragment from a String. For example: "`<xml-fragment>12.34e+5</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlFloat parse(String s,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlFloat` fragment from a String. For example: "`<xml-fragment>12.34e+5</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlFloat parse(File f)
     throws XmlException, IOException
```

Parses a `XmlFloat` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlFloat parse(File f,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlFloat` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlFloat parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlFloat` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlFloat parse(URL u,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlFloat` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlFloat parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlFloat` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlFloat parse(InputStream is,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlFloat` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlFloat parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlFloat` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlFloat parse(Reader r,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlFloat` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlFloat parse(Node node)
     throws XmlException
```

Parses a `XmlFloat` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlFloat parse(Node node,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlFloat` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlFloat parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlFloat` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlFloat parse(XMLInputStream xis,
                             XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlFloat` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlGDay Interface

public interface ***XmlGDay***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:gDay type. A gDay specifies only a day−of−month.

Convertible to `Calendar`, `GDate`, or an int.

Related Topics

`XmlCalendar`
`GDate`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlGDay.Factory`** A class with methods for creating instances of `XmlGDay`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public Calendar` ***calendarValue()***
Returns this value as a `Calendar`

`public GDate` ***gDateValue()***
Returns this value as a `GDate`

`public Calendar` ***getCalendarValue()***
Returns this value as a `Calendar`

`public GDate` ***getGDateValue()***
Returns this value as a `GDate`

`public int` ***getIntValue()***
Returns this value as an int from 1−31

`public int` ***intValue()***
Returns this value as an int from 1−31

`public void` ***set***`(Calendar` *c)*
Sets this value as a `Calendar`

`public void` ***set***`(GDateSpecification` *gd)*
Sets this value as a `GDateSpecification`

`public void` ***set***`(int` *v)*
Sets this value as an int from 1−31

`public void` ***setCalendarValue***`(Calendar` *c)*
Sets this value as a `Calendar`

`public void` ***setGDateValue***`(GDate` *gd)*
Sets this value as a `GDateSpecification`

`public void` ***setIntValue***`(int` *v)*
Sets this value as an int from 1−31

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGDay.getCalendarValue()`

`public Calendar calendarValue()`

Returns this value as a `Calendar`

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGDay.getGDateValue()`

`public GDate gDateValue()`

Returns this value as a `GDate`

---

getCalendarValue() Method

`public Calendar getCalendarValue()`

Returns this value as a `Calendar`

---

getGDateValue() Method

`public GDate getGDateValue()`

Returns this value as a `GDate`

---

getIntValue() Method

`public int getIntValue()`

Returns this value as an int from 1−31

---

intValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGDay.getIntValue()`

`public int intValue()`

Returns this value as an int from 1−31

---

set(Calendar) Method

DEPRECATED replaced with `com.bea.xml.XmlGDay.setCalendarValue(java.util.Calendar)`

`public void set(Calendar` *`c`***`)`**

Sets this value as a `Calendar`

---

set(GDateSpecification) Method

DEPRECATED replaced with `com.bea.xml.XmlGDay.setGDateValue(com.bea.xml.GDate)`

`public void set(GDateSpecification` *`gd`***`)`**

Sets this value as a `GDateSpecification`

---

set(int) Method

DEPRECATED replaced with `com.bea.xml.XmlGDay.setIntValue(int)`

`public void set(int` *`v`***`)`**

Sets this value as an int from 1−31

---

setCalendarValue(Calendar) Method

`public void setCalendarValue(Calendar c)`

Sets this value as a `Calendar`

---

setGDateValue(GDate) Method

`public void setGDateValue(GDate gd)`

Sets this value as a `GDateSpecification`

---

setIntValue(int) Method

`public void setIntValue(int v)`

Sets this value as an int from 1−31

com.bea.xml

XmlGDay.Factory Class

public static final class ***XmlGDay.Factory***

extends `Object`

A class with methods for creating instances of `XmlGDay`.

Hierarchy

```
Object
 XmlGDay.Factory
```

Enclosing interface

`XmlGDay`

Method Summary

public static `XmlGDay` ***newInstance***()
Creates an empty instance of `XmlGDay`

public static `XmlGDay` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlGDay`

public static `XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

public static `XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

public static `XmlGDay` ***newValue***(`Object` *obj*)
Creates an immutable `XmlGDay` value

public static `XmlGDay` ***parse***(`String` *s*)
Parses a `XmlGDay` fragment from a String.

public static `XmlGDay` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlGDay` fragment from a String.

public static `XmlGDay` ***parse***(`File` *f*)
Parses a `XmlGDay` fragment from a File.

`public static` ***parse***(*File f, XmlOptions options*)
`XmlGDay` Parses a `XmlGDay` fragment from a File.

`public static`
`XmlGDay` ***parse***(*URL u*)
Parses a `XmlGDay` fragment from a URL.

`public static`
`XmlGDay` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlGDay` fragment from a URL.

`public static`
`XmlGDay` ***parse***(*InputStream is*)
Parses a `XmlGDay` fragment from an InputStream.

`public static`
`XmlGDay` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlGDay` fragment from an InputStream.

`public static`
`XmlGDay` ***parse***(*Reader r*)
Parses a `XmlGDay` fragment from a Reader.

`public static`
`XmlGDay` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlGDay` fragment from a Reader.

`public static`
`XmlGDay` ***parse***(*Node node*)
Parses a `XmlGDay` fragment from a DOM Node.

`public static`
`XmlGDay` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlGDay` fragment from a DOM Node.

`public static`
`XmlGDay` ***parse***(*XMLInputStream xis*)
Parses a `XmlGDay` fragment from an XMLInputStream.

`public static`
`XmlGDay` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlGDay` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlGDay newInstance()`

Creates an empty instance of `XmlGDay`

---

newInstance(XmlOptions) Method

```
public static XmlGDay newInstance(XmlOptions options)
```

Creates an empty instance of `XmlGDay`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlGDay newValue(Object obj)
```

Creates an immutable `XmlGDay` value

---

parse(String) Method

```
public static XmlGDay parse(String s)
     throws XmlException
```

Parses a `XmlGDay` fragment from a String. For example: "`<xml-fragment>---14</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlGDay parse(String s,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlGDay` fragment from a String. For example: "`<xml-fragment>---14</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlGDay parse(File f)
     throws XmlException, IOException
```

Parses a `XmlGDay` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlGDay parse(File f,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGDay` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlGDay parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlGDay` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlGDay parse(URL u,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGDay` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlGDay parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlGDay` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlGDay parse(InputStream is,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGDay` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlGDay parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlGDay` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlGDay parse(Reader r,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGDay` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlGDay parse(Node node)
     throws XmlException
```

Parses a `XmlGDay` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlGDay parse(Node node,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlGDay` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGDay parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlGDay` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGDay parse(XMLInputStream xis,
                            XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlGDay` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlGMonth Interface

public interface ***XmlGMonth***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:gMonth type. A gMonth specifies only a month−of−year.

Convertible to `Calendar`, `GDate`, or an int.

Related Topics

`XmlCalendar`
`GDate`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlGMonth.Factory`** A class with methods for creating instances of `XmlGMonth`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public Calendar` ***calendarValue()***
Returns this value as a `Calendar`

`public GDate` ***gDateValue()***
Returns this value as a `GDate`

`public Calendar` ***getCalendarValue()***
Returns this value as a `Calendar`

`public GDate` ***getGDateValue()***
Returns this value as a `GDate`

`public int` ***getIntValue()***
Returns this value as an int from 1−12

`public int` ***intValue()***
Returns this value as an int from 1−12

`public void` ***set***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***set***`(GDateSpecification gd)`
Sets this value as a `GDateSpecification`

`public void` ***set***`(int v)`
Sets this value as an int from 1−12

`public void` ***setCalendarValue***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***setGDateValue***`(GDate gd)`
Sets this value as a `GDateSpecification`

`public void` ***setIntValue***`(int v)`
Sets this value as an int from 1−12

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGMonth.getCalendarValue()`

`public Calendar calendarValue()`

Returns this value as a `Calendar`

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGMonth.getGDateValue()`

`public GDate gDateValue()`

Returns this value as a `GDate`

---

getCalendarValue() Method

`public Calendar getCalendarValue()`

Returns this value as a `Calendar`

---

getGDateValue() Method

`public GDate getGDateValue()`

Returns this value as a `GDate`

---

getIntValue() Method

`public int getIntValue()`

Returns this value as an int from 1−12

---

intValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGMonth.getIntValue()`

`public int intValue()`

Returns this value as an int from 1−12

---

set(Calendar) Method

DEPRECATED replaced with `com.bea.xml.XmlGMonth.setCalendarValue(java.util.Calendar)`

`public void set(Calendar` *`c`***`)`**

Sets this value as a `Calendar`

---

set(GDateSpecification) Method

DEPRECATED replaced with `com.bea.xml.XmlGMonth.setGDateValue(com.bea.xml.GDate)`

`public void set(GDateSpecification` *`gd`***`)`**

Sets this value as a `GDateSpecification`

---

set(int) Method

DEPRECATED replaced with `com.bea.xml.XmlGMonth.setIntValue(int)`

public void set(int *v*)

Sets this value as an int from 1−12

---

setCalendarValue(Calendar) Method

public void setCalendarValue(Calendar *c*)

Sets this value as a `Calendar`

---

setGDateValue(GDate) Method

public void setGDateValue(GDate *gd*)

Sets this value as a `GDateSpecification`

---

setIntValue(int) Method

public void setIntValue(int *v*)

Sets this value as an int from 1−12

XmlGMonth.Factory Class

public static final class ***XmlGMonth.Factory***

extends `Object`

A class with methods for creating instances of `XmlGMonth`.

Hierarchy

```
Object
 XmlGMonth.Factory
```

Enclosing interface

`XmlGMonth`

Method Summary

`public static XmlGMonth` ***newInstance***()
Creates an empty instance of `XmlGMonth`

`public static XmlGMonth` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlGMonth`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlGMonth` ***newValue***(`Object` *obj*)
Creates an immutable `XmlGMonth` value

`public static XmlGMonth` ***parse***(`String` *s*)
Parses a `XmlGMonth` fragment from a String.

`public static XmlGMonth` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlGMonth` fragment from a String.

`public static XmlGMonth` ***parse***(`File` *f*)
Parses a `XmlGMonth` fragment from a File.

`public static XmlGMonth` ***parse***(*File f, XmlOptions options*)
Parses a `XmlGMonth` fragment from a File.

`public static XmlGMonth` ***parse***(*URL u*)
Parses a `XmlGMonth` fragment from a URL.

`public static XmlGMonth` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlGMonth` fragment from a URL.

`public static XmlGMonth` ***parse***(*InputStream is*)
Parses a `XmlGMonth` fragment from an InputStream.

`public static XmlGMonth` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlGMonth` fragment from an InputStream.

`public static XmlGMonth` ***parse***(*Reader r*)
Parses a `XmlGMonth` fragment from a Reader.

`public static XmlGMonth` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlGMonth` fragment from a Reader.

`public static XmlGMonth` ***parse***(*Node node*)
Parses a `XmlGMonth` fragment from a DOM Node.

`public static XmlGMonth` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlGMonth` fragment from a DOM Node.

`public static XmlGMonth` ***parse***(*XMLInputStream xis*)
Parses a `XmlGMonth` fragment from an XMLInputStream.

`public static XmlGMonth` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlGMonth` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlGMonth newInstance()`

Creates an empty instance of `XmlGMonth`

---

newInstance(XmlOptions) Method

```
public static XmlGMonth newInstance(XmlOptions options)
```

Creates an empty instance of `XmlGMonth`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlGMonth newValue(Object obj)
```

Creates an immutable `XmlGMonth` value

---

parse(String) Method

```
public static XmlGMonth parse(String s)
     throws XmlException
```

Parses a `XmlGMonth` fragment from a String. For example: "`<xml-fragment>--06</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlGMonth parse(String s,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlGMonth` fragment from a String. For example: "`<xml-fragment>--06</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlGMonth parse(File f)
     throws XmlException, IOException
```

Parses a `XmlGMonth` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlGMonth parse(File f,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGMonth` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlGMonth parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlGMonth` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlGMonth parse(URL u,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGMonth` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlGMonth parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlGMonth` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlGMonth parse(InputStream is,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGMonth` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlGMonth parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlGMonth` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlGMonth parse(Reader r,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGMonth` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlGMonth parse(Node node)
     throws XmlException
```

Parses a `XmlGMonth` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlGMonth parse(Node node,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlGMonth` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGMonth parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlGMonth` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGMonth parse(XMLInputStream xis,
                              XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlGMonth` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlGMonthDay Interface

public interface ***XmlGMonthDay***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:gMonthDay type. A gMonthDay specifies a day of a specific month without specifying the year.

Convertible to `Calendar` or `GDate`.

Related Topics

`XmlCalendar`
`GDate`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlGMonthDay.Factory`** A class with methods for creating instances of `XmlGMonthDay`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public Calendar` ***`calendarValue`***`()`
Returns this value as a `Calendar`

`public GDate` ***`gDateValue`***`()`
Returns this value as a `GDate`

`public Calendar` ***`getCalendarValue`***`()`
Returns this value as a `Calendar`

`public GDate` ***`getGDateValue`***`()`
Returns this value as a `GDate`

`public void` ***`set`***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***`set`***`(GDateSpecification gd)`
Sets this value as a `GDateSpecification`

`public void` ***`setCalendarValue`***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***`setGDateValue`***`(GDate gd)`
Sets this value as a `GDateSpecification`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save,`

`xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGMonthDay.getCalendarValue()`

`public Calendar calendarValue()`

Returns this value as a `Calendar`

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGMonthDay.getGDateValue()`

`public GDate gDateValue()`

Returns this value as a `GDate`

---

getCalendarValue() Method

`public Calendar getCalendarValue()`

Returns this value as a `Calendar`

---

getGDateValue() Method

`public GDate getGDateValue()`

Returns this value as a `GDate`

---

set(Calendar) Method

DEPRECATED replaced with
`com.bea.xml.XmlGMonthDay.setCalendarValue(java.util.Calendar)`

`public void set(Calendar c)`

Sets this value as a `Calendar`

---

set(GDateSpecification) Method

DEPRECATED replaced with
`com.bea.xml.XmlGMonthDay.setGDateValue(com.bea.xml.GDate)`

`public void set(GDateSpecification gd)`

Sets this value as a `GDateSpecification`

---

setCalendarValue(Calendar) Method

`public void setCalendarValue(Calendar c)`

Sets this value as a `Calendar`

---

setGDateValue(GDate) Method

`public void setGDateValue(GDate gd)`

Sets this value as a `GDateSpecification`

XmlGMonthDay.Factory Class

public static final class ***XmlGMonthDay.Factory***

extends `Object`

A class with methods for creating instances of `XmlGMonthDay`.

Hierarchy

```
Object
 XmlGMonthDay.Factory
```

Enclosing interface

`XmlGMonthDay`

Method Summary

`public static XmlGMonthDay` ***newInstance***()
Creates an empty instance of `XmlGMonthDay`

`public static XmlGMonthDay` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlGMonthDay`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlGMonthDay` ***newValue***(`Object` *obj*)
Creates an immutable `XmlGMonthDay` value

`public static XmlGMonthDay` ***parse***(`String` *s*)
Parses a `XmlGMonthDay` fragment from a String.

`public static XmlGMonthDay` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlGMonthDay` fragment from a String.

`public static XmlGMonthDay` ***parse***(`File` *f*)
Parses a `XmlGMonthDay` fragment from a File.

`public static XmlGMonthDay` ***parse***(*File f, XmlOptions options*)
Parses a `XmlGMonthDay` fragment from a File.

`public static XmlGMonthDay` ***parse***(*URL u*)
Parses a `XmlGMonthDay` fragment from a URL.

`public static XmlGMonthDay` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlGMonthDay` fragment from a URL.

`public static XmlGMonthDay` ***parse***(*InputStream is*)
Parses a `XmlGMonthDay` fragment from an InputStream.

`public static XmlGMonthDay` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlGMonthDay` fragment from an InputStream.

`public static XmlGMonthDay` ***parse***(*Reader r*)
Parses a `XmlGMonthDay` fragment from a Reader.

`public static XmlGMonthDay` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlGMonthDay` fragment from a Reader.

`public static XmlGMonthDay` ***parse***(*Node node*)
Parses a `XmlGMonthDay` fragment from a DOM Node.

`public static XmlGMonthDay` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlGMonthDay` fragment from a DOM Node.

`public static XmlGMonthDay` ***parse***(*XMLInputStream xis*)
Parses a `XmlGMonthDay` fragment from an XMLInputStream.

`public static XmlGMonthDay` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlGMonthDay` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlGMonthDay newInstance()`

Creates an empty instance of `XmlGMonthDay`

---

newInstance(XmlOptions) Method

```
public static XmlGMonthDay newInstance(XmlOptions options)
```

Creates an empty instance of `XmlGMonthDay`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlGMonthDay newValue(Object obj)
```

Creates an immutable `XmlGMonthDay` value

---

parse(String) Method

```
public static XmlGMonthDay parse(String s)
     throws XmlException
```

Parses a `XmlGMonthDay` fragment from a String. For example: "`<xml-fragment>--06-14</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlGMonthDay parse(String s,
                                 XmlOptions options)
     throws XmlException
```

Parses a `XmlGMonthDay` fragment from a String. For example: "`<xml-fragment>--06-14</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlGMonthDay parse(File f)
     throws XmlException, IOException
```

Parses a `XmlGMonthDay` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlGMonthDay parse(File f,
                                 XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGMonthDay` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlGMonthDay parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlGMonthDay` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlGMonthDay parse(URL u,
                                 XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGMonthDay` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlGMonthDay parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlGMonthDay` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlGMonthDay parse(InputStream is,
                                 XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGMonthDay` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlGMonthDay parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlGMonthDay` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlGMonthDay parse(Reader r,
                                 XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGMonthDay` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlGMonthDay parse(Node node)
     throws XmlException
```

Parses a `XmlGMonthDay` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlGMonthDay parse(Node node,
                                 XmlOptions options)
     throws XmlException
```

Parses a `XmlGMonthDay` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGMonthDay parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlGMonthDay` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGMonthDay parse(XMLInputStream xis,
                                 XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlGMonthDay` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlGYear Interface

public interface ***XmlGYear***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:gYear type. A gYear specifies a Gregorian year (AD).

Convertible to `Calendar`, `GDate`, or an int.

Related Topics

`XmlCalendar`
`GDate`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlGYear.Factory`** A class with methods for creating instances of `XmlGYear`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public Calendar` ***calendarValue()***
Returns this value as a `Calendar`

`public GDate` ***gDateValue()***
Returns this value as a `GDate`

`public Calendar` ***getCalendarValue()***
Returns this value as a `Calendar`

`public GDate` ***getGDateValue()***
Returns this value as a `GDate`

`public int` ***getIntValue()***
Returns this value as an int from 1−31

`public int` ***intValue()***
Returns this value as an int from 1−31

`public void` ***set***`(Calendar` *c)*
Sets this value as a `Calendar`

`public void` ***set***`(GDateSpecification` *gd)*
Sets this value as a `GDateSpecification`

`public void` ***set***`(int` *v)*
Sets this value as an int from 1−31

`public void` ***setCalendarValue***`(Calendar` *c)*
Sets this value as a `Calendar`

`public void` ***setGDateValue***`(GDate` *gd)*
Sets this value as a `GDateSpecification`

`public void` ***setIntValue***`(int` *v)*
Sets this value as an int from 1−31

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGYear.getCalendarValue()`

`public Calendar calendarValue()`

Returns this value as a `Calendar`

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGYear.getGDateValue()`

`public GDate gDateValue()`

Returns this value as a `GDate`

---

getCalendarValue() Method

`public Calendar getCalendarValue()`

Returns this value as a `Calendar`

---

getGDateValue() Method

`public GDate getGDateValue()`

Returns this value as a `GDate`

---

getIntValue() Method

`public int getIntValue()`

Returns this value as an int from 1−31

---

intValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGYear.getIntValue()`

`public int intValue()`

Returns this value as an int from 1−31

---

set(Calendar) Method

DEPRECATED replaced with `com.bea.xml.XmlGYear.setCalendarValue(java.util.Calendar)`

`public void set(Calendar` *`c`***`)`**

Sets this value as a `Calendar`

---

set(GDateSpecification) Method

DEPRECATED replaced with `com.bea.xml.XmlGYear.setGDateValue(com.bea.xml.GDate)`

`public void set(GDateSpecification` *`gd`***`)`**

Sets this value as a `GDateSpecification`

---

set(int) Method

DEPRECATED replaced with `com.bea.xml.XmlGYear.setIntValue(int)`

public void set(int *v*)

Sets this value as an int from 1−31

---

setCalendarValue(Calendar) Method

public void setCalendarValue(Calendar *c*)

Sets this value as a `Calendar`

---

setGDateValue(GDate) Method

public void setGDateValue(GDate *gd*)

Sets this value as a `GDateSpecification`

---

setIntValue(int) Method

public void setIntValue(int *v*)

Sets this value as an int from 1−31

XmlGYear.Factory Class

public static final class ***XmlGYear.Factory***

extends `Object`

A class with methods for creating instances of `XmlGYear`.

Hierarchy

```
Object
 XmlGYear.Factory
```

Enclosing interface

`XmlGYear`

Method Summary

`public static XmlGYear` ***newInstance***()
Creates an empty instance of `XmlGYear`

`public static XmlGYear` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlGYear`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlGYear` ***newValue***(`Object` *obj*)
Creates an immutable `XmlGYear` value

`public static XmlGYear` ***parse***(`String` *s*)
Parses a `XmlGYear` fragment from a String.

`public static XmlGYear` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlGYear` fragment from a String.

`public static XmlGYear` ***parse***(`File` *f*)
Parses a `XmlGYear` fragment from a File.

`public static XmlGYear` ***parse***(*File f, XmlOptions options*)
Parses a `XmlGYear` fragment from a File.

`public static XmlGYear` ***parse***(*URL u*)
Parses a `XmlGYear` fragment from a URL.

`public static XmlGYear` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlGYear` fragment from a URL.

`public static XmlGYear` ***parse***(*InputStream is*)
Parses a `XmlGYear` fragment from an InputStream.

`public static XmlGYear` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlGYear` fragment from an InputStream.

`public static XmlGYear` ***parse***(*Reader r*)
Parses a `XmlGYear` fragment from a Reader.

`public static XmlGYear` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlGYear` fragment from a Reader.

`public static XmlGYear` ***parse***(*Node node*)
Parses a `XmlGYear` fragment from a DOM Node.

`public static XmlGYear` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlGYear` fragment from a DOM Node.

`public static XmlGYear` ***parse***(*XMLInputStream xis*)
Parses a `XmlGYear` fragment from an XMLInputStream.

`public static XmlGYear` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlGYear` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlGYear newInstance()`

Creates an empty instance of `XmlGYear`

---

newInstance(XmlOptions) Method

```
public static XmlGYear newInstance(XmlOptions options)
```

Creates an empty instance of `XmlGYear`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlGYear newValue(Object obj)
```

Creates an immutable `XmlGYear` value

---

parse(String) Method

```
public static XmlGYear parse(String s)
     throws XmlException
```

Parses a `XmlGYear` fragment from a String. For example: "`<xml-fragment>2003</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlGYear parse(String s,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlGYear` fragment from a String. For example: "`<xml-fragment>2003</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlGYear parse(File f)
     throws XmlException, IOException
```

Parses a `XmlGYear` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlGYear parse(File f,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGYear` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlGYear parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlGYear` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlGYear parse(URL u,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGYear` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlGYear parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlGYear` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlGYear parse(InputStream is,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGYear` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlGYear parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlGYear` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlGYear parse(Reader r,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGYear` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlGYear parse(Node node)
     throws XmlException
```

Parses a `XmlGYear` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlGYear parse(Node node,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlGYear` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGYear parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlGYear` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGYear parse(XMLInputStream xis,
                             XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlGYear` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlGYearMonth Interface

public interface ***XmlGYearMonth***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:gYearMonth type. A gYearMonth specifies a month in a specific year.

Convertible to `Calendar`, `GDate`.

Related Topics

`XmlCalendar`
`GDate`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlGYearMonth.Factory`** A class with methods for creating instances of `XmlGYearMonth`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public Calendar` ***`calendarValue`***`()`
Returns this value as a `Calendar`

`public GDate` ***`gDateValue`***`()`
Returns this value as a `GDate`

`public Calendar` ***`getCalendarValue`***`()`
Returns this value as a `Calendar`

`public GDate` ***`getGDateValue`***`()`
Returns this value as a `GDate`

`public void` ***`set`***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***`set`***`(GDateSpecification gd)`
Sets this value as a `GDateSpecification`

`public void` ***`setCalendarValue`***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***`setGDateValue`***`(GDate gd)`
Sets this value as a `GDateSpecification`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save,`

`xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGYearMonth.getCalendarValue()`

`public Calendar calendarValue()`

Returns this value as a `Calendar`

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlGYearMonth.getGDateValue()`

`public GDate gDateValue()`

Returns this value as a `GDate`

---

getCalendarValue() Method

`public Calendar getCalendarValue()`

Returns this value as a `Calendar`

---

getGDateValue() Method

`public GDate getGDateValue()`

Returns this value as a `GDate`

---

set(Calendar) Method

DEPRECATED replaced with
`com.bea.xml.XmlGYearMonth.setCalendarValue(java.util.Calendar)`

`public void set(Calendar c)`

Sets this value as a `Calendar`

---

set(GDateSpecification) Method

DEPRECATED replaced with
`com.bea.xml.XmlGYearMonth.setGDateValue(com.bea.xml.GDate)`

`public void set(GDateSpecification gd)`

Sets this value as a `GDateSpecification`

---

setCalendarValue(Calendar) Method

`public void setCalendarValue(Calendar c)`

Sets this value as a `Calendar`

---

setGDateValue(GDate) Method

`public void setGDateValue(GDate gd)`

Sets this value as a `GDateSpecification`

XmlGYearMonth.Factory Class

public static final class ***XmlGYearMonth.Factory***

extends `Object`

A class with methods for creating instances of `XmlGYearMonth`.

Hierarchy

```
Object
 XmlGYearMonth.Factory
```

Enclosing interface

`XmlGYearMonth`

Method Summary

`public static XmlGYearMonth` ***newInstance***()
Creates an empty instance of `XmlGYearMonth`

`public static XmlGYearMonth` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlGYearMonth`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlGYearMonth` ***newValue***(`Object` *obj*)
Creates an immutable `XmlGYearMonth` value

`public static XmlGYearMonth` ***parse***(`String` *s*)
Parses a `XmlGYearMonth` fragment from a String.

`public static XmlGYearMonth` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlGYearMonth` fragment from a String.

`public static XmlGYearMonth` ***parse***(`File` *f*)
Parses a `XmlGYearMonth` fragment from a File.

`public static XmlGYearMonth` ***parse***(*File f, XmlOptions options*)
Parses a `XmlGYearMonth` fragment from a File.

`public static XmlGYearMonth` ***parse***(*URL u*)
Parses a `XmlGYearMonth` fragment from a URL.

`public static XmlGYearMonth` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlGYearMonth` fragment from a URL.

`public static XmlGYearMonth` ***parse***(*InputStream is*)
Parses a `XmlGYearMonth` fragment from an InputStream.

`public static XmlGYearMonth` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlGYearMonth` fragment from an InputStream.

`public static XmlGYearMonth` ***parse***(*Reader r*)
Parses a `XmlGYearMonth` fragment from a Reader.

`public static XmlGYearMonth` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlGYearMonth` fragment from a Reader.

`public static XmlGYearMonth` ***parse***(*Node node*)
Parses a `XmlGYearMonth` fragment from a DOM Node.

`public static XmlGYearMonth` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlGYearMonth` fragment from a DOM Node.

`public static XmlGYearMonth` ***parse***(*XMLInputStream xis*)
Parses a `XmlGYearMonth` fragment from an XMLInputStream.

`public static XmlGYearMonth` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlGYearMonth` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlGYearMonth newInstance()`

Creates an empty instance of `XmlGYearMonth`

---

newInstance(XmlOptions) Method

```
public static XmlGYearMonth newInstance(XmlOptions options)
```

Creates an empty instance of `XmlGYearMonth`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlGYearMonth newValue(Object obj)
```

Creates an immutable `XmlGYearMonth` value

---

parse(String) Method

```
public static XmlGYearMonth parse(String s)
     throws XmlException
```

Parses a `XmlGYearMonth` fragment from a String. For example: "`<xml-fragment>2003-06</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlGYearMonth parse(String s,
                                  XmlOptions options)
     throws XmlException
```

Parses a `XmlGYearMonth` fragment from a String. For example: "`<xml-fragment>2003-06</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlGYearMonth parse(File f)
     throws XmlException, IOException
```

Parses a `XmlGYearMonth` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlGYearMonth parse(File f,
                                  XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGYearMonth` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlGYearMonth parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlGYearMonth` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlGYearMonth parse(URL u,
                                  XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGYearMonth` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlGYearMonth parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlGYearMonth` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlGYearMonth parse(InputStream is,
                                  XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGYearMonth` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlGYearMonth parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlGYearMonth` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlGYearMonth parse(Reader r,
                                  XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlGYearMonth` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlGYearMonth parse(Node node)
     throws XmlException
```

Parses a `XmlGYearMonth` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlGYearMonth parse(Node node,
                                  XmlOptions options)
     throws XmlException
```

Parses a `XmlGYearMonth` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGYearMonth parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlGYearMonth` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlGYearMonth parse(XMLInputStream xis,
                                  XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlGYearMonth` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlHexBinary Interface

public interface ***XmlHexBinary***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:hexBinary type.

Convertible to a byte array.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlHexBinary.Factory`** A class with methods for creating instances of `XmlHexBinary`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

public
byte[] ***byteArrayValue***()
Returns this value as a byte array.

public
byte[] ***getByteArrayValue***()
Returns this value as a byte array.

public
void ***set***(*byte[] ba*)
Sets this value as a byte array.

public
void ***setByteArrayValue***(*byte[] ba*)
Sets this value as a byte array.

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

byteArrayValue() Method

DEPRECATED replaced by `com.bea.xml.XmlHexBinary.getByteArrayValue()`

`public byte[] byteArrayValue()`

Returns this value as a byte array.

---

getByteArrayValue() Method

`public byte[] getByteArrayValue()`

Returns this value as a byte array.

---

set(byte[]) Method

DEPRECATED replaced by `com.bea.xml.XmlHexBinary.setByteArrayValue(byte[])`

`public void set(byte[] `*`ba`*`)`

Sets this value as a byte array.

---

setByteArrayValue(byte[]) Method

`public void setByteArrayValue(byte[] `*`ba`*`)`

Sets this value as a byte array.

XmlHexBinary.Factory Class

public static final class ***XmlHexBinary.Factory***

extends `Object`

A class with methods for creating instances of `XmlHexBinary`.

Hierarchy

```
Object
 XmlHexBinary.Factory
```

Enclosing interface

`XmlHexBinary`

Method Summary

`public static XmlHexBinary` ***newInstance***()
Creates an empty instance of `XmlHexBinary`

`public static XmlHexBinary` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlHexBinary`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlHexBinary` ***newValue***(`Object` *obj*)
Creates an immutable `XmlHexBinary` value

`public static XmlHexBinary` ***parse***(`String` *s*)
Parses a `XmlHexBinary` fragment from a String.

`public static XmlHexBinary` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlHexBinary` fragment from a String.

`public static XmlHexBinary` ***parse***(`File` *f*)
Parses a `XmlHexBinary` fragment from a File.

`public static XmlHexBinary` ***parse***(*File f, XmlOptions options*)
Parses a `XmlHexBinary` fragment from a File.

`public static XmlHexBinary` ***parse***(*URL u*)
Parses a `XmlHexBinary` fragment from a URL.

`public static XmlHexBinary` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlHexBinary` fragment from a URL.

`public static XmlHexBinary` ***parse***(*InputStream is*)
Parses a `XmlHexBinary` fragment from an InputStream.

`public static XmlHexBinary` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlHexBinary` fragment from an InputStream.

`public static XmlHexBinary` ***parse***(*Reader r*)
Parses a `XmlHexBinary` fragment from a Reader.

`public static XmlHexBinary` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlHexBinary` fragment from a Reader.

`public static XmlHexBinary` ***parse***(*Node node*)
Parses a `XmlHexBinary` fragment from a DOM Node.

`public static XmlHexBinary` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlHexBinary` fragment from a DOM Node.

`public static XmlHexBinary` ***parse***(*XMLInputStream xis*)
Parses a `XmlHexBinary` fragment from an XMLInputStream.

`public static XmlHexBinary` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlHexBinary` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlHexBinary newInstance()`

Creates an empty instance of `XmlHexBinary`

---

newInstance(XmlOptions) Method

```
public static XmlHexBinary newInstance(XmlOptions options)
```

Creates an empty instance of `XmlHexBinary`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlHexBinary newValue(Object obj)
```

Creates an immutable `XmlHexBinary` value

---

parse(String) Method

```
public static XmlHexBinary parse(String s)
     throws XmlException
```

Parses a `XmlHexBinary` fragment from a String. For example: "`<xml-fragment>68656c6c6f</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlHexBinary parse(String s,
                                 XmlOptions options)
     throws XmlException
```

Parses a `XmlHexBinary` fragment from a String. For example: "`<xml-fragment>68656c6c6f</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlHexBinary parse(File f)
     throws XmlException, IOException
```

Parses a `XmlHexBinary` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlHexBinary parse(File f,
                                 XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlHexBinary` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlHexBinary parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlHexBinary` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlHexBinary parse(URL u,
                                 XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlHexBinary` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlHexBinary parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlHexBinary` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlHexBinary parse(InputStream is,
                                 XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlHexBinary` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlHexBinary parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlHexBinary` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlHexBinary parse(Reader r,
                                 XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlHexBinary` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlHexBinary parse(Node node)
     throws XmlException
```

Parses a `XmlHexBinary` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlHexBinary parse(Node node,
                                 XmlOptions options)
     throws XmlException
```

Parses a `XmlHexBinary` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlHexBinary parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlHexBinary` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlHexBinary parse(XMLInputStream xis,
                                 XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlHexBinary` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlID Interface

public interface ***XmlID***

extends `XmlNCName`

Corresponds to the XML Schema xs:ID type.

When validated, IDs must be unique within a document. An element may not have more than one attribute whose type is ID.

Convertible to a `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlName`, `XmlNCName`, `XmlNormalizedString`, `XmlObject`, `XmlString`, `XmlToken`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlID.Factory`** A class with methods for creating instances of `XmlID`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlName`

`XmlName.Factory`

Nested classes from interface com.bea.xml.`XmlNCName`

`XmlNCName.Factory`

Nested classes from interface com.bea.xml.`XmlNormalizedString`

`XmlNormalizedString.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Nested classes from interface com.bea.xml.`XmlToken`

`XmlToken.Factory`

Field Summary

`public static final SchemaType` ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlName`

`type`

Fields from interface com.bea.xml.`XmlNCName`

`type`

Fields from interface com.bea.xml.`XmlNormalizedString`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Fields from interface com.bea.xml.`XmlToken`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlID.Factory Class

public static final class ***XmlID.Factory***

extends `Object`

A class with methods for creating instances of `XmlID`.

Hierarchy

```
Object
 XmlID.Factory
```

Enclosing interface

`XmlID`

Method Summary

`public static XmlID` ***newInstance***()
Creates an empty instance of `XmlID`

`public static XmlID` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlID`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlID` ***newValue***(`Object` *obj*)
Creates an immutable `XmlID` value

`public static XmlID` ***parse***(`String` *s*)
Parses a `XmlID` fragment from a String.

`public static XmlID` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlID` fragment from a String.

`public static XmlID` ***parse***(`File` *f*)
Parses a `XmlID` fragment from a File.

`public static XmlID` ***parse***(*File f, XmlOptions options*)
Parses a `XmlID` fragment from a File.

`public static XmlID` ***parse***(*URL u*)
Parses a `XmlID` fragment from a URL.

`public static XmlID` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlID` fragment from a URL.

`public static XmlID` ***parse***(*InputStream is*)
Parses a `XmlID` fragment from an InputStream.

`public static XmlID` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlID` fragment from an InputStream.

`public static XmlID` ***parse***(*Reader r*)
Parses a `XmlID` fragment from a Reader.

`public static XmlID` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlID` fragment from a Reader.

`public static XmlID` ***parse***(*Node node*)
Parses a `XmlID` fragment from a DOM Node.

`public static XmlID` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlID` fragment from a DOM Node.

`public static XmlID` ***parse***(*XMLInputStream xis*)
Parses a `XmlID` fragment from an XMLInputStream.

`public static XmlID` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlID` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlID newInstance()`

Creates an empty instance of `XmlID`

---

newInstance(XmlOptions) Method

```
public static XmlID newInstance(XmlOptions options)
```

Creates an empty instance of `XmlID`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlID newValue(Object obj)
```

Creates an immutable `XmlID` value

---

parse(String) Method

```
public static XmlID parse(String s)
     throws XmlException
```

Parses a `XmlID` fragment from a String. For example: "`<xml-fragment>n1</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlID parse(String s,
                          XmlOptions options)
     throws XmlException
```

Parses a `XmlID` fragment from a String. For example: "`<xml-fragment>n1</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlID parse(File f)
     throws XmlException, IOException
```

Parses a `XmlID` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlID parse(File f,
                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlID` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlID parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlID` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlID parse(URL u,
                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlID` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlID parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlID` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlID parse(InputStream is,
                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlID` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlID parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlID` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlID parse(Reader r,
                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlID` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlID parse(Node node)
     throws XmlException
```

Parses a `XmlID` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlID parse(Node node,
                          XmlOptions options)
     throws XmlException
```

Parses a `XmlID` fragment from a DOM Node.

Exceptions

XmlException

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlID parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlID` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlID parse(XMLInputStream xis,
                          XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlID` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlIDREF Interface

public interface ***XmlIDREF***

extends `XmlNCName`

Corresponds to the XML Schema xs:IDREF type.

When validated, IDREF values must match an ID value that is present within the document. This rule is only verified when a whole document is validated at once.

Convertible to a `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlName`, `XmlNCName`, `XmlNormalizedString`, `XmlObject`, `XmlString`, `XmlToken`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlIDREF.Factory`** A class with methods for creating instances of `XmlIDREF`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlName`

`XmlName.Factory`

Nested classes from interface com.bea.xml.`XmlNCName`

`XmlNCName.Factory`

Nested classes from interface com.bea.xml.`XmlNormalizedString`

`XmlNormalizedString.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Nested classes from interface com.bea.xml.`XmlToken`

`XmlToken.Factory`

Field Summary

`public static final SchemaType` ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlName`

`type`

Fields from interface com.bea.xml.`XmlNCName`

`type`

Fields from interface com.bea.xml.`XmlNormalizedString`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Fields from interface com.bea.xml.`XmlToken`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

com.bea.xml

XmlIDREF.Factory Class

public static final class ***XmlIDREF.Factory***

extends `Object`

A class with methods for creating instances of `XmlIDREF`.

Hierarchy

```
Object
 XmlIDREF.Factory
```

Enclosing interface

`XmlIDREF`

Method Summary

`public static XmlIDREF` ***newInstance***()
Creates an empty instance of `XmlIDREF`

`public static XmlIDREF` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlIDREF`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlIDREF` ***newValue***(`Object` *obj*)
Creates an immutable `XmlIDREF` value

`public static XmlIDREF` ***parse***(`String` *s*)
Parses a `XmlIDREF` fragment from a String.

`public static XmlIDREF` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlIDREF` fragment from a String.

`public static XmlIDREF` ***parse***(`File` *f*)
Parses a `XmlIDREF` fragment from a File.

`public static XmlIDREF` ***parse***(*File f, XmlOptions options*)
Parses a `XmlIDREF` fragment from a File.

`public static XmlIDREF` ***parse***(*URL u*)
Parses a `XmlIDREF` fragment from a URL.

`public static XmlIDREF` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlIDREF` fragment from a URL.

`public static XmlIDREF` ***parse***(*InputStream is*)
Parses a `XmlIDREF` fragment from an InputStream.

`public static XmlIDREF` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlIDREF` fragment from an InputStream.

`public static XmlIDREF` ***parse***(*Reader r*)
Parses a `XmlIDREF` fragment from a Reader.

`public static XmlIDREF` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlIDREF` fragment from a Reader.

`public static XmlIDREF` ***parse***(*Node node*)
Parses a `XmlIDREF` fragment from a DOM Node.

`public static XmlIDREF` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlIDREF` fragment from a DOM Node.

`public static XmlIDREF` ***parse***(*XMLInputStream xis*)
Parses a `XmlIDREF` fragment from an XMLInputStream.

`public static XmlIDREF` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlIDREF` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlIDREF newInstance()`

Creates an empty instance of `XmlIDREF`

---

newInstance(XmlOptions) Method

```
public static XmlIDREF newInstance(XmlOptions options)
```

Creates an empty instance of `XmlIDREF`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlIDREF newValue(Object obj)
```

Creates an immutable `XmlIDREF` value

---

parse(String) Method

```
public static XmlIDREF parse(String s)
     throws XmlException
```

Parses a `XmlIDREF` fragment from a String.

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlIDREF parse(String s,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlIDREF` fragment from a String.

Exceptions

XmlException

---

parse(File) Method

```
public static XmlIDREF parse(File f)
     throws XmlException, IOException
```

Parses a `XmlIDREF` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlIDREF parse(File f,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlIDREF` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlIDREF parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlIDREF` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlIDREF parse(URL u,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlIDREF` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlIDREF parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlIDREF` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlIDREF parse(InputStream is,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlIDREF` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlIDREF parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlIDREF` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlIDREF parse(Reader r,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlIDREF` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlIDREF parse(Node node)
     throws XmlException
```

Parses a `XmlIDREF` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlIDREF parse(Node node,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlIDREF` fragment from a DOM Node.

Exceptions

XmlException

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlIDREF parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlIDREF` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlIDREF parse(XMLInputStream xis,
                             XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlIDREF` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlIDREFS Interface

public interface ***XmlIDREFS***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:IDREFS type, a list type.

When validated, IDREF values must match an ID value that is present within the document. This rule is only verified when a whole document is validated at once.

Convertible to a `List`.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlIDREFS.Factory`** A class with methods for creating instances of `XmlIDREFS`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

public
List ***getListValue()***
Returns the value as a `List` of `String` values

public
List ***listValue()***
Returns the value as a `List` of `String` values

public
void ***set(List l)***
Sets the value as a `List`

public
void ***setListValue(List l)***
Sets the value as a `List`

public
List ***xgetListValue()***
Returns the value as a `List` of `XmlIDREF` values

public
List ***xlistValue()***
Returns the value as a `List` of `XmlIDREF` values

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getListValue() Method

`public List getListValue()`

Returns the value as a `List` of `String` values

---

listValue() Method

DEPRECATED replaced by `com.bea.xml.XmlIDREFS.getListValue()`

`public List listValue()`

Returns the value as a `List` of `String` values

---

set(List) Method

DEPRECATED replaced by `com.bea.xml.XmlIDREFS.getListValue()`

`public void set(List l)`

Sets the value as a `List`

---

setListValue(List) Method

`public void setListValue(List l)`

Sets the value as a `List`

---

xgetListValue() Method

`public List xgetListValue()`

Returns the value as a `List` of `XmlIDREF` values

---

xlistValue() Method

DEPRECATED replaced by `com.bea.xml.XmlIDREFS.getListValue()`

`public List xlistValue()`

Returns the value as a `List` of `XmlIDREF` values

XmlIDREFS.Factory Class

public static final class ***XmlIDREFS.Factory***

extends `Object`

A class with methods for creating instances of `XmlIDREFS`.

Hierarchy

```
Object
 XmlIDREFS.Factory
```

Enclosing interface

`XmlIDREFS`

Method Summary

`public static XmlIDREFS` ***newInstance***()
Creates an empty instance of `XmlIDREFS`

`public static XmlIDREFS` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlIDREFS`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlIDREFS` ***newValue***(`Object` *obj*)
Creates an immutable `XmlIDREFS` value

`public static XmlIDREFS` ***parse***(`String` *s*)
Parses a `XmlIDREFS` fragment from a String.

`public static XmlIDREFS` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlIDREFS` fragment from a String.

`public static XmlIDREFS` ***parse***(`File` *f*)
Parses a `XmlIDREFS` fragment from a File.

`public static XmlIDREFS` ***parse***(*File f, XmlOptions options*)
Parses a `XmlIDREFS` fragment from a File.

`public static XmlIDREFS` ***parse***(*URL u*)
Parses a `XmlIDREFS` fragment from a URL.

`public static XmlIDREFS` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlIDREFS` fragment from a URL.

`public static XmlIDREFS` ***parse***(*InputStream is*)
Parses a `XmlIDREFS` fragment from an InputStream.

`public static XmlIDREFS` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlIDREFS` fragment from an InputStream.

`public static XmlIDREFS` ***parse***(*Reader r*)
Parses a `XmlIDREFS` fragment from a Reader.

`public static XmlIDREFS` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlIDREFS` fragment from a Reader.

`public static XmlIDREFS` ***parse***(*Node node*)
Parses a `XmlIDREFS` fragment from a DOM Node.

`public static XmlIDREFS` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlIDREFS` fragment from a DOM Node.

`public static XmlIDREFS` ***parse***(*XMLInputStream xis*)
Parses a `XmlIDREFS` fragment from an XMLInputStream.

`public static XmlIDREFS` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlIDREFS` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlIDREFS newInstance()`

Creates an empty instance of `XmlIDREFS`

---

newInstance(XmlOptions) Method

```
public static XmlIDREFS newInstance(XmlOptions options)
```

Creates an empty instance of `XmlIDREFS`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlIDREFS newValue(Object obj)
```

Creates an immutable `XmlIDREFS` value

---

parse(String) Method

```
public static XmlIDREFS parse(String s)
     throws XmlException
```

Parses a `XmlIDREFS` fragment from a String.

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlIDREFS parse(String s,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlIDREFS` fragment from a String.

Exceptions

XmlException

---

parse(File) Method

```
public static XmlIDREFS parse(File f)
     throws XmlException, IOException
```

Parses a `XmlIDREFS` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlIDREFS parse(File f,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlIDREFS` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlIDREFS parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlIDREFS` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlIDREFS parse(URL u,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlIDREFS` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlIDREFS parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlIDREFS` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlIDREFS parse(InputStream is,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlIDREFS` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlIDREFS parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlIDREFS` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlIDREFS parse(Reader r,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlIDREFS` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlIDREFS parse(Node node)
     throws XmlException
```

Parses a `XmlIDREFS` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlIDREFS parse(Node node,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlIDREFS` fragment from a DOM Node.

Exceptions

XmlException

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlIDREFS parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlIDREFS` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlIDREFS parse(XMLInputStream xis,
                              XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlIDREFS` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlInt Interface

public interface ***XmlInt***

extends `XmlLong`

Corresponds to the XML Schema xs:int type. One of the derived types based on xs:decimal.

Naturally, convertible to a Java int.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlLong`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlByte`, `XmlShort`

Nested Class Summary

`public static final class`	**`XmlInt.Factory`** A class with methods for creating instances of `XmlInt`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlLong`

`XmlLong.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType` ***type***
The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlLong`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public int` ***getIntValue***()
Returns this value as an int

`public int` ***intValue***()
Returns this value as an int

`public void` ***set***(*int v*)
Sets this value as an int

`public void` ***setIntValue***(*int v*)
Sets this value as an int

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlLong`

`getLongValue, longValue, set, setLongValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getIntValue() Method

`public int getIntValue()`

Returns this value as an int

---

intValue() Method

DEPRECATED replaced with `com.bea.xml.XmlInt.getIntValue()`

`public int intValue()`

Returns this value as an int

---

set(int) Method

DEPRECATED replaced with `com.bea.xml.XmlInt.setIntValue(int)`

`public void set(int v)`

Sets this value as an int

---

setIntValue(int) Method

`public void setIntValue(int v)`

Sets this value as an int

XmlInt.Factory Class

public static final class ***XmlInt.Factory***

extends `Object`

A class with methods for creating instances of `XmlInt`.

Hierarchy

```
Object
 XmlInt.Factory
```

Enclosing interface

`XmlInt`

Method Summary

`public static XmlInt` ***newInstance***()
Creates an empty instance of `XmlInt`

`public static XmlInt` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlInt`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlInt` ***newValue***(`Object` *obj*)
Creates an immutable `XmlInt` value

`public static XmlInt` ***parse***(`String` *s*)
Parses a `XmlInt` fragment from a String.

`public static XmlInt` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlInt` fragment from a String.

`public static XmlInt` ***parse***(`File` *f*)
Parses a `XmlInt` fragment from a File.

`public static XmlInt` ***parse***(*File f, XmlOptions options*)
Parses a `XmlInt` fragment from a File.

`public static XmlInt` ***parse***(*URL u*)
Parses a `XmlInt` fragment from a URL.

`public static XmlInt` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlInt` fragment from a URL.

`public static XmlInt` ***parse***(*InputStream is*)
Parses a `XmlInt` fragment from an InputStream.

`public static XmlInt` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlInt` fragment from an InputStream.

`public static XmlInt` ***parse***(*Reader r*)
Parses a `XmlInt` fragment from a Reader.

`public static XmlInt` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlInt` fragment from a Reader.

`public static XmlInt` ***parse***(*Node node*)
Parses a `XmlInt` fragment from a DOM Node.

`public static XmlInt` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlInt` fragment from a DOM Node.

`public static XmlInt` ***parse***(*XMLInputStream xis*)
Parses a `XmlInt` fragment from an XMLInputStream.

`public static XmlInt` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlInt` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlInt newInstance()`

Creates an empty instance of `XmlInt`

---

newInstance(XmlOptions) Method

```
public static XmlInt newInstance(XmlOptions options)
```

Creates an empty instance of `XmlInt`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlInt newValue(Object obj)
```

Creates an immutable `XmlInt` value

---

parse(String) Method

```
public static XmlInt parse(String s)
     throws XmlException
```

Parses a `XmlInt` fragment from a String. For example: "`<xml-fragment>1234567</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlInt parse(String s,
                           XmlOptions options)
     throws XmlException
```

Parses a `XmlInt` fragment from a String. For example: "`<xml-fragment>1234567</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlInt parse(File f)
     throws XmlException, IOException
```

Parses a `XmlInt` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlInt parse(File f,
                           XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlInt` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlInt parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlInt` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlInt parse(URL u,
                           XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlInt` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlInt parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlInt` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlInt parse(InputStream is,
                           XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlInt` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlInt parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlInt` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlInt parse(Reader r,
                           XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlInt` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlInt parse(Node node)
     throws XmlException
```

Parses a `XmlInt` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlInt parse(Node node,
                           XmlOptions options)
     throws XmlException
```

Parses a `XmlInt` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlInt parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlInt` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlInt parse(XMLInputStream xis,
                           XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlInt` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlInteger Interface

public interface ***XmlInteger***

extends `XmlDecimal`

Corresponds to the XML Schema xs:integer type. One of the derived types based on xs:decimal.

This type should not be confused with xs:int or Java `Integer`. This type represents an arbitrary−precision integer with any number of digits, while a Java int or an xs:int is a 32−bit finite−precision integer.

Convertible to a Java `BigInteger`.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlByte`, `XmlInt`, `XmlLong`, `XmlNegativeInteger`, `XmlNonNegativeInteger`, `XmlNonPositiveInteger`, `XmlPositiveInteger`, `XmlShort`, `XmlUnsignedByte`, `XmlUnsignedInt`, `XmlUnsignedLong`, `XmlUnsignedShort`

Nested Class Summary

`public static final class`	**`XmlInteger.Factory`** A class with methods for creating instances of `XmlInteger`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType` ***type***
The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public BigInteger` ***bigIntegerValue()***
Returns this value as a `BigInteger`

`public BigInteger` ***getBigIntegerValue()***
Returns this value as a `BigInteger`

`public void` ***set****(BigInteger bi)*
Sets this value as a `BigInteger`

`public void` ***setBigIntegerValue****(BigInteger bi)*
Sets this value as a `BigInteger`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

bigIntegerValue() Method

DEPRECATED replaced with `com.bea.xml.XmlInteger.getBigIntegerValue()`

`public BigInteger bigIntegerValue()`

Returns this value as a `BigInteger`

---

getBigIntegerValue() Method

`public BigInteger getBigIntegerValue()`

Returns this value as a `BigInteger`

---

set(BigInteger) Method

DEPRECATED replaced with `com.bea.xml.XmlInteger.setBigIntegerValue(java.math.BigInteger)`

`public void set(BigInteger bi)`

Sets this value as a `BigInteger`

---

setBigIntegerValue(BigInteger) Method

`public void setBigIntegerValue(BigInteger bi)`

Sets this value as a `BigInteger`

XmlInteger.Factory Class

public static final class ***XmlInteger.Factory***

extends `Object`

A class with methods for creating instances of `XmlInteger`.

Hierarchy

```
Object
 XmlInteger.Factory
```

Enclosing interface

`XmlInteger`

Method Summary

`public static XmlInteger` ***newInstance***()
Creates an empty instance of `XmlInteger`

`public static XmlInteger` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlInteger`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlInteger` ***newValue***(`Object` *obj*)
Creates an immutable `XmlInteger` value

`public static XmlInteger` ***parse***(`String` *s*)
Parses a `XmlInteger` fragment from a String.

`public static XmlInteger` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlInteger` fragment from a String.

`public static XmlInteger` ***parse***(`File` *f*)
Parses a `XmlInteger` fragment from a File.

`public static XmlInteger` ***parse***(*File f, XmlOptions options*)
Parses a `XmlInteger` fragment from a File.

`public static XmlInteger` ***parse***(*URL u*)
Parses a `XmlInteger` fragment from a URL.

`public static XmlInteger` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlInteger` fragment from a URL.

`public static XmlInteger` ***parse***(*InputStream is*)
Parses a `XmlInteger` fragment from an InputStream.

`public static XmlInteger` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlInteger` fragment from an InputStream.

`public static XmlInteger` ***parse***(*Reader r*)
Parses a `XmlInteger` fragment from a Reader.

`public static XmlInteger` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlInteger` fragment from a Reader.

`public static XmlInteger` ***parse***(*Node node*)
Parses a `XmlInteger` fragment from a DOM Node.

`public static XmlInteger` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlInteger` fragment from a DOM Node.

`public static XmlInteger` ***parse***(*XMLInputStream xis*)
Parses a `XmlInteger` fragment from an XMLInputStream.

`public static XmlInteger` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlInteger` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlInteger newInstance()`

Creates an empty instance of `XmlInteger`

---

newInstance(XmlOptions) Method

```
public static XmlInteger newInstance(XmlOptions options)
```

Creates an empty instance of `XmlInteger`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlInteger newValue(Object obj)
```

Creates an immutable `XmlInteger` value

---

parse(String) Method

```
public static XmlInteger parse(String s)
     throws XmlException
```

Parses a `XmlInteger` fragment from a String. For example: "`<xml-fragment>1234567890</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlInteger parse(String s,
                               XmlOptions options)
     throws XmlException
```

Parses a `XmlInteger` fragment from a String. For example: "`<xml-fragment>1234567890</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlInteger parse(File f)
     throws XmlException, IOException
```

Parses a `XmlInteger` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlInteger parse(File f,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlInteger` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlInteger parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlInteger` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlInteger parse(URL u,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlInteger` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlInteger parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlInteger` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlInteger parse(InputStream is,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlInteger` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlInteger parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlInteger` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlInteger parse(Reader r,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlInteger` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlInteger parse(Node node)
     throws XmlException
```

Parses a `XmlInteger` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlInteger parse(Node node,
                               XmlOptions options)
     throws XmlException
```

Parses a `XmlInteger` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlInteger parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlInteger` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlInteger parse(XMLInputStream xis,
                               XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlInteger` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlLanguage Interface

public interface ***XmlLanguage***

extends `XmlToken`

Corresponds to the XML Schema xs:language type.

This type is intended to represent an standard ISO language code string.

Convertible to a Java `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlNormalizedString`, `XmlObject`, `XmlString`, `XmlToken`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlLanguage.Factory`** A class with methods for creating instances of `XmlLanguage`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlNormalizedString`

`XmlNormalizedString.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Nested classes from interface com.bea.xml.`XmlToken`

`XmlToken.Factory`

Field Summary

public static final SchemaType ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlNormalizedString`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Fields from interface com.bea.xml.`XmlToken`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlLanguage.Factory Class

public static final class ***XmlLanguage.Factory***

extends `Object`

A class with methods for creating instances of `XmlLanguage`.

Hierarchy

```
Object
 XmlLanguage.Factory
```

Enclosing interface

`XmlLanguage`

Method Summary

`public static XmlLanguage` ***newInstance***()
Creates an empty instance of `XmlLanguage`

`public static XmlLanguage` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlLanguage`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlLanguage` ***newValue***(`Object` *obj*)
Creates an immutable `XmlLanguage` value

`public static XmlLanguage` ***parse***(`String` *s*)
Parses a `XmlLanguage` fragment from a String.

`public static XmlLanguage` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlLanguage` fragment from a String.

`public static XmlLanguage` ***parse***(`File` *f*)
Parses a `XmlLanguage` fragment from a File.

`public static XmlLanguage` ***parse***(*File f, XmlOptions options*)
Parses a `XmlLanguage` fragment from a File.

`public static XmlLanguage` ***parse***(*URL u*)
Parses a `XmlLanguage` fragment from a URL.

`public static XmlLanguage` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlLanguage` fragment from a URL.

`public static XmlLanguage` ***parse***(*InputStream is*)
Parses a `XmlLanguage` fragment from an InputStream.

`public static XmlLanguage` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlLanguage` fragment from an InputStream.

`public static XmlLanguage` ***parse***(*Reader r*)
Parses a `XmlLanguage` fragment from a Reader.

`public static XmlLanguage` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlLanguage` fragment from a Reader.

`public static XmlLanguage` ***parse***(*Node node*)
Parses a `XmlLanguage` fragment from a DOM Node.

`public static XmlLanguage` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlLanguage` fragment from a DOM Node.

`public static XmlLanguage` ***parse***(*XMLInputStream xis*)
Parses a `XmlLanguage` fragment from an XMLInputStream.

`public static XmlLanguage` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlLanguage` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlLanguage newInstance()`

Creates an empty instance of `XmlLanguage`

---

newInstance(XmlOptions) Method

```
public static XmlLanguage newInstance(XmlOptions options)
```

Creates an empty instance of `XmlLanguage`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlLanguage newValue(Object obj)
```

Creates an immutable `XmlLanguage` value

---

parse(String) Method

```
public static XmlLanguage parse(String s)
     throws XmlException
```

Parses a `XmlLanguage` fragment from a String. For example: "`<xml-fragment>en-us</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlLanguage parse(String s,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlLanguage` fragment from a String. For example: "`<xml-fragment>en-us</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlLanguage parse(File f)
     throws XmlException, IOException
```

Parses a `XmlLanguage` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlLanguage parse(File f,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlLanguage` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlLanguage parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlLanguage` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlLanguage parse(URL u,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlLanguage` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlLanguage parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlLanguage` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlLanguage parse(InputStream is,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlLanguage` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlLanguage parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlLanguage` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlLanguage parse(Reader r,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlLanguage` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlLanguage parse(Node node)
     throws XmlException
```

Parses a `XmlLanguage` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlLanguage parse(Node node,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlLanguage` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlLanguage parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlLanguage` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlLanguage parse(XMLInputStream xis,
                                XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlLanguage` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlLineNumber Class

public class ***XmlLineNumber***

extends `XmlCursor.XmlBookmark`

A subclass of XmlBookmark that holds line number information. If a document is parsed with line numbers enabled, these bookmarks will be placed at appropriate locations within the document.

Related Topics

`XmlOptions.setLoadLineNumbers()`

Hierarchy

```
Object
 XmlCursor.XmlBookmark
  XmlLineNumber
```

Field Summary

Fields from com.bea.xml.`XmlCursor.XmlBookmark`

`_currentMark, _ref`

Constructor Summary

XmlLineNumber*(int line, int column, int offset)*

Constructs a line number and column with no file offset information.

XmlLineNumber*(int line, int column)*

Constructs a line number and column with no file offset information.

XmlLineNumber*(int line)*

Constructs a line number with no column or offset information.

Method Summary

public
int ***getColumn()***
Returns the 1−based column number, or −1 if not known.

public
int ***getLine()***
Returns the 1−based line number, or −1 if not known.

public
int ***getOffset()***
Returns the 0−based file offset number, or −1 if not known.

Methods from `com.bea.xml.XmlCursor.XmlBookmark`

`createCursor, getKey, toBookmark`

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

XmlLineNumber

```
public XmlLineNumber(int line,
                     int column,
                     int offset)
```

Constructs a line number and column with no file offset information.

XmlLineNumber

```
public XmlLineNumber(int line,
                     int column)
```

Constructs a line number and column with no file offset information.

XmlLineNumber

```
public XmlLineNumber(int line)
```

Constructs a line number with no column or offset information.

Method Detail

getColumn() Method

```
public int getColumn()
```

Returns the 1−based column number, or −1 if not known.

---

getLine() Method

```
public int getLine()
```

Returns the 1−based line number, or −1 if not known.

---

getOffset() Method

```
public int getOffset()
```

Returns the 0−based file offset number, or −1 if not known.

com.bea.xml

XmlLong Interface

public interface ***XmlLong***

extends `XmlInteger`

Corresponds to the XML Schema xs:long type. One of the derived types based on xs:decimal.

Naturally, convertible to a Java long.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlByte`, `XmlInt`, `XmlShort`

Nested Class Summary

`public static final class`	**`XmlLong.Factory`** A class with methods for creating instances of `XmlLong`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static`	***`type`***

`final SchemaType` The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public long` ***`getLongValue`***`()`
Returns this value as a long

`public long` ***`longValue`***`()`
Returns this value as a long

`public void` ***`set`***`(long v)`
Sets this value as a long

`public void` ***`setLongValue`***`(long v)`
Sets this value as a long

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getLongValue() Method

`public long getLongValue()`

Returns this value as a long

---

longValue() Method

DEPRECATED replaced with `com.bea.xml.XmlLong.getLongValue()`

`public long longValue()`

Returns this value as a long

---

set(long) Method

DEPRECATED replaced with `com.bea.xml.XmlLong.setLongValue(long)`

`public void set(long v)`

Sets this value as a long

---

setLongValue(long) Method

`public void setLongValue(long v)`

Sets this value as a long

com.bea.xml

XmlLong.Factory Class

public static final class ***XmlLong.Factory***

extends `Object`

A class with methods for creating instances of `XmlLong`.

Hierarchy

```
Object
 XmlLong.Factory
```

Enclosing interface

`XmlLong`

Method Summary

`public static XmlLong` ***newInstance***()
Creates an empty instance of `XmlLong`

`public static XmlLong` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlLong`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlLong` ***newValue***(`Object` *obj*)
Creates an immutable `XmlLong` value

`public static XmlLong` ***parse***(`String` *s*)
Parses a `XmlLong` fragment from a String.

`public static XmlLong` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlLong` fragment from a String.

`public static XmlLong` ***parse***(`File` *f*)
Parses a `XmlLong` fragment from a File.

`public static XmlLong` ***parse***(*File f, XmlOptions options*)
Parses a `XmlLong` fragment from a File.

`public static XmlLong` ***parse***(*URL u*)
Parses a `XmlLong` fragment from a URL.

`public static XmlLong` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlLong` fragment from a URL.

`public static XmlLong` ***parse***(*InputStream is*)
Parses a `XmlLong` fragment from an InputStream.

`public static XmlLong` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlLong` fragment from an InputStream.

`public static XmlLong` ***parse***(*Reader r*)
Parses a `XmlLong` fragment from a Reader.

`public static XmlLong` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlLong` fragment from a Reader.

`public static XmlLong` ***parse***(*Node node*)
Parses a `XmlLong` fragment from a DOM Node.

`public static XmlLong` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlLong` fragment from a DOM Node.

`public static XmlLong` ***parse***(*XMLInputStream xis*)
Parses a `XmlLong` fragment from an XMLInputStream.

`public static XmlLong` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlLong` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlLong newInstance()`

Creates an empty instance of `XmlLong`

---

newInstance(XmlOptions) Method

```
public static XmlLong newInstance(XmlOptions options)
```

Creates an empty instance of `XmlLong`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlLong newValue(Object obj)
```

Creates an immutable `XmlLong` value

---

parse(String) Method

```
public static XmlLong parse(String s)
     throws XmlException
```

Parses a `XmlLong` fragment from a String. For example: "`<xml-fragment>123456789</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlLong parse(String s,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlLong` fragment from a String. For example: "`<xml-fragment>123456789</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlLong parse(File f)
     throws XmlException, IOException
```

Parses a `XmlLong` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlLong parse(File f,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlLong` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlLong parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlLong` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlLong parse(URL u,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlLong` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlLong parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlLong` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlLong parse(InputStream is,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlLong` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlLong parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlLong` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlLong parse(Reader r,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlLong` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlLong parse(Node node)
     throws XmlException
```

Parses a `XmlLong` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlLong parse(Node node,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlLong` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlLong parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlLong` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlLong parse(XMLInputStream xis,
                            XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlLong` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

XmlName Interface

public interface ***XmlName***

extends `XmlToken`

Corresponds to the XML Schema xs:Name type. One of the derived types based on xs:string.

This kind of string is the same as the lexical representation used for XML 1.0 element and attribute names, e.g., "pre:a−b.cdef". It is just a physical string, however, and it should NOT be confused with `XmlQName`, which is a logical combination of localname and namespace URI.

Convertible to `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlNormalizedString`, `XmlObject`, `XmlString`, `XmlToken`, `XmlTokenSource`

All Known Subinterfaces

`XmlENTITY`, `XmlID`, `XmlIDREF`, `XmlNCName`

Nested Class Summary

`public static final class`	**`XmlName.Factory`** A class with methods for creating instances of `XmlName`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlNormalizedString`

`XmlNormalizedString.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Nested classes from interface com.bea.xml.`XmlToken`

`XmlToken.Factory`

Field Summary

`public static final SchemaType` ***type***

The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlNormalizedString`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Fields from interface com.bea.xml.`XmlToken`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlName.Factory Class

public static final class ***XmlName.Factory***

extends `Object`

A class with methods for creating instances of `XmlName`.

Hierarchy

```
Object
 XmlName.Factory
```

Enclosing interface

`XmlName`

Method Summary

`public static XmlName` ***newInstance***()
Creates an empty instance of `XmlName`

`public static XmlName` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlName`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlName` ***newValue***(`Object` *obj*)
Creates an immutable `XmlName` value

`public static XmlName` ***parse***(`String` *s*)
Parses a `XmlName` fragment from a String.

`public static XmlName` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlName` fragment from a String.

`public static XmlName` ***parse***(`File` *f*)
Parses a `XmlName` fragment from a File.

`public static XmlName` ***parse***(*File f, XmlOptions options*)
Parses a `XmlName` fragment from a File.

`public static XmlName` ***parse***(*URL u*)
Parses a `XmlName` fragment from a URL.

`public static XmlName` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlName` fragment from a URL.

`public static XmlName` ***parse***(*InputStream is*)
Parses a `XmlName` fragment from an InputStream.

`public static XmlName` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlName` fragment from an InputStream.

`public static XmlName` ***parse***(*Reader r*)
Parses a `XmlName` fragment from a Reader.

`public static XmlName` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlName` fragment from a Reader.

`public static XmlName` ***parse***(*Node node*)
Parses a `XmlName` fragment from a DOM Node.

`public static XmlName` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlName` fragment from a DOM Node.

`public static XmlName` ***parse***(*XMLInputStream xis*)
Parses a `XmlName` fragment from an XMLInputStream.

`public static XmlName` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlName` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlName newInstance()`

Creates an empty instance of `XmlName`

---

newInstance(XmlOptions) Method

```
public static XmlName newInstance(XmlOptions options)
```

Creates an empty instance of `XmlName`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlName newValue(Object obj)
```

Creates an immutable `XmlName` value

---

parse(String) Method

```
public static XmlName parse(String s)
     throws XmlException
```

Parses a `XmlName` fragment from a String. For example: "`<xml-fragment>My:Name</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlName parse(String s,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlName` fragment from a String. For example: "`<xml-fragment>My:Name</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlName parse(File f)
     throws XmlException, IOException
```

Parses a `XmlName` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlName parse(File f,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlName` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlName parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlName` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlName parse(URL u,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlName` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlName parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlName` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlName parse(InputStream is,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlName` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlName parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlName` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlName parse(Reader r,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlName` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlName parse(Node node)
     throws XmlException
```

Parses a `XmlName` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlName parse(Node node,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlName` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlName parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlName` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlName parse(XMLInputStream xis,
                            XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlName` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

XmlNCName Interface

public interface ***XmlNCName***

extends `XmlName`

Corresponds to the XML Schema xs:Name type. One of the derived types based on xs:string.

This kind of string is the same as the non−colonized strings that are used for XML localnames and prefixes, "my−href.2". It is just a physical string, however, and it should NOT be confused with `XmlQName`, which is a logical combination of localname and namespace URI.

Convertible to `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlName`, `XmlNormalizedString`, `XmlObject`, `XmlString`, `XmlToken`, `XmlTokenSource`

All Known Subinterfaces

`XmlENTITY`, `XmlID`, `XmlIDREF`

Nested Class Summary

`public static final class`	**`XmlNCName.Factory`** A class with methods for creating instances of `XmlNCName`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlName`

`XmlName.Factory`

Nested classes from interface com.bea.xml.`XmlNormalizedString`

`XmlNormalizedString.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Nested classes from interface com.bea.xml.`XmlToken`

`XmlToken.Factory`

Field Summary

`public static final SchemaType` ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlName`

`type`

Fields from interface com.bea.xml.`XmlNormalizedString`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Fields from interface com.bea.xml.`XmlToken`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil,`

`toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlNCName.Factory Class

public static final class ***XmlNCName.Factory***

extends `Object`

A class with methods for creating instances of `XmlNCName`.

Hierarchy

```
Object
 XmlNCName.Factory
```

Enclosing interface

`XmlNCName`

Method Summary

`public static XmlNCName` ***newInstance***()
Creates an empty instance of `XmlNCName`

`public static XmlNCName` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlNCName`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlNCName` ***newValue***(`Object` *obj*)
Creates an immutable `XmlNCName` value

`public static XmlNCName` ***parse***(`String` *s*)
Parses a `XmlNCName` fragment from a String.

`public static XmlNCName` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlNCName` fragment from a String.

`public static XmlNCName` ***parse***(`File` *f*)
Parses a `XmlNCName` fragment from a File.

`public static XmlNCName` ***parse***(*File f, XmlOptions options*)
Parses a `XmlNCName` fragment from a File.

`public static XmlNCName` ***parse***(*URL u*)
Parses a `XmlNCName` fragment from a URL.

`public static XmlNCName` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlNCName` fragment from a URL.

`public static XmlNCName` ***parse***(*InputStream is*)
Parses a `XmlNCName` fragment from an InputStream.

`public static XmlNCName` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlNCName` fragment from an InputStream.

`public static XmlNCName` ***parse***(*Reader r*)
Parses a `XmlNCName` fragment from a Reader.

`public static XmlNCName` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlNCName` fragment from a Reader.

`public static XmlNCName` ***parse***(*Node node*)
Parses a `XmlNCName` fragment from a DOM Node.

`public static XmlNCName` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlNCName` fragment from a DOM Node.

`public static XmlNCName` ***parse***(*XMLInputStream xis*)
Parses a `XmlNCName` fragment from an XMLInputStream.

`public static XmlNCName` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlNCName` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlNCName newInstance()`

Creates an empty instance of `XmlNCName`

---

newInstance(XmlOptions) Method

```
public static XmlNCName newInstance(XmlOptions options)
```

Creates an empty instance of `XmlNCName`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlNCName newValue(Object obj)
```

Creates an immutable `XmlNCName` value

---

parse(String) Method

```
public static XmlNCName parse(String s)
     throws XmlException
```

Parses a `XmlNCName` fragment from a String. For example: "`<xml-fragment>My-Name</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlNCName parse(String s,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlNCName` fragment from a String. For example: "`<xml-fragment>My-Name</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlNCName parse(File f)
     throws XmlException, IOException
```

Parses a `XmlNCName` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlNCName parse(File f,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNCName` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlNCName parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlNCName` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlNCName parse(URL u,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNCName` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlNCName parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlNCName` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlNCName parse(InputStream is,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNCName` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlNCName parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlNCName` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlNCName parse(Reader r,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNCName` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlNCName parse(Node node)
     throws XmlException
```

Parses a `XmlNCName` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlNCName parse(Node node,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlNCName` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNCName parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlNCName` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNCName parse(XMLInputStream xis,
                              XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlNCName` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlNegativeInteger Interface

public interface ***XmlNegativeInteger***

extends `XmlNonPositiveInteger`

Corresponds to the XML Schema xs:negativeInteger type. One of the derived types based on xs:decimal.

Verified to be negative when validating.

Convertible to `BigInteger`.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlNonPositiveInteger`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlNegativeInteger.Factory`** A class with methods for creating instances of `XmlNegativeInteger`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlNonPositiveInteger`

`XmlNonPositiveInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

public static final SchemaType ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlNonPositiveInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlNegativeInteger.Factory Class

public static final class ***XmlNegativeInteger.Factory***

extends `Object`

A class with methods for creating instances of `XmlNegativeInteger`.

Hierarchy

```
Object
 XmlNegativeInteger.Factory
```

Enclosing interface

`XmlNegativeInteger`

Method Summary

`public static XmlNegativeInteger` ***newInstance***()
Creates an empty instance of `XmlNegativeInteger`

`public static XmlNegativeInteger` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlNegativeInteger`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlNegativeInteger` ***newValue***(`Object` *obj*)
Creates an immutable `XmlNegativeInteger` value

`public static XmlNegativeInteger` ***parse***(`String` *s*)
Parses a `XmlNegativeInteger` fragment from a String.

`public static XmlNegativeInteger` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlNegativeInteger` fragment from a String.

`public static XmlNegativeInteger` ***parse***(`File` *f*)
Parses a `XmlNegativeInteger` fragment from a File.

public static XmlNegativeInteger **parse**(File f, XmlOptions options)
Parses a XmlNegativeInteger fragment from a File.

public static XmlNegativeInteger **parse**(URL u)
Parses a XmlNegativeInteger fragment from a URL.

public static XmlNegativeInteger **parse**(URL u, XmlOptions options)
Parses a XmlNegativeInteger fragment from a URL.

public static XmlNegativeInteger **parse**(InputStream is)
Parses a XmlNegativeInteger fragment from an InputStream.

public static XmlNegativeInteger **parse**(InputStream is, XmlOptions options)
Parses a XmlNegativeInteger fragment from an InputStream.

public static XmlNegativeInteger **parse**(Reader r)
Parses a XmlNegativeInteger fragment from a Reader.

public static XmlNegativeInteger **parse**(Reader r, XmlOptions options)
Parses a XmlNegativeInteger fragment from a Reader.

public static XmlNegativeInteger **parse**(Node node)
Parses a XmlNegativeInteger fragment from a DOM Node.

public static XmlNegativeInteger **parse**(Node node, XmlOptions options)
Parses a XmlNegativeInteger fragment from a DOM Node.

public static XmlNegativeInteger **parse**(XMLInputStream xis)
Parses a XmlNegativeInteger fragment from an XMLInputStream.

public static XmlNegativeInteger **parse**(XMLInputStream xis, XmlOptions options)
Parses a XmlNegativeInteger fragment from an XMLInputStream.

Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

newInstance() Method

`public static XmlNegativeInteger newInstance()`

Creates an empty instance of XmlNegativeInteger

newInstance(XmlOptions) Method

```
public static XmlNegativeInteger newInstance(XmlOptions options)
```

Creates an empty instance of `XmlNegativeInteger`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlNegativeInteger newValue(Object obj)
```

Creates an immutable `XmlNegativeInteger` value

---

parse(String) Method

```
public static XmlNegativeInteger parse(String s)
     throws XmlException
```

Parses a `XmlNegativeInteger` fragment from a String. For example: "`<xml-fragment>-1234567890</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlNegativeInteger parse(String s,
                                       XmlOptions options)
     throws XmlException
```

Parses a `XmlNegativeInteger` fragment from a String. For example: "`<xml-fragment>-1234567890</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlNegativeInteger parse(File f)
     throws XmlException, IOException
```

Parses a `XmlNegativeInteger` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlNegativeInteger parse(File f,
                                       XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNegativeInteger` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlNegativeInteger parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlNegativeInteger` fragment from a URL.

Exceptions

`XmlException`
`IOException`

---

parse(URL, XmlOptions) Method

```
public static XmlNegativeInteger parse(URL u,
                                       XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNegativeInteger` fragment from a URL.

Exceptions

`XmlException`
`IOException`

---

parse(InputStream) Method

```
public static XmlNegativeInteger parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlNegativeInteger` fragment from an InputStream.

Exceptions

`XmlException`
`IOException`

---

parse(InputStream, XmlOptions) Method

```
public static XmlNegativeInteger parse(InputStream is,
                                       XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNegativeInteger` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlNegativeInteger parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlNegativeInteger` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlNegativeInteger parse(Reader r,
                                       XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNegativeInteger` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlNegativeInteger parse(Node node)
     throws XmlException
```

Parses a `XmlNegativeInteger` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlNegativeInteger parse(Node node,
                                       XmlOptions options)
     throws XmlException
```

Parses a `XmlNegativeInteger` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNegativeInteger parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlNegativeInteger` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNegativeInteger parse(XMLInputStream xis,
                                       XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlNegativeInteger` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlNMTOKEN Interface

public interface ***XmlNMTOKEN***

extends `XmlToken`

Corresponds to the XML Schema xs:NMTOKEN type. One of the derived types based on xs:string.

An NMTOKEN is XML's closest concept to an "identifier"; for example, it does not permit spaces and only limited punctuation. So NMTOKEN is commonly used to describe a single token or enumerated string value.

Convertible to `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlNormalizedString`, `XmlObject`, `XmlString`, `XmlToken`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlNMTOKEN.Factory`** A class with methods for creating instances of `XmlNMTOKEN`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlNormalizedString`

`XmlNormalizedString.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Nested classes from interface com.bea.xml.`XmlToken`

`XmlToken.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlNormalizedString`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Fields from interface com.bea.xml.`XmlToken`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlNMTOKEN.Factory Class

public static final class ***XmlNMTOKEN.Factory***

extends `Object`

A class with methods for creating instances of `XmlNMTOKEN`.

Hierarchy

```
Object
 XmlNMTOKEN.Factory
```

Enclosing interface

`XmlNMTOKEN`

Method Summary

`public static XmlNMTOKEN` ***newInstance***()
Creates an empty instance of `XmlNMTOKEN`

`public static XmlNMTOKEN` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlNMTOKEN`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlNMTOKEN` ***newValue***(`Object` *obj*)
Creates an immutable `XmlNMTOKEN` value

`public static XmlNMTOKEN` ***parse***(`String` *s*)
Parses a `XmlNMTOKEN` fragment from a String.

`public static XmlNMTOKEN` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlNMTOKEN` fragment from a String.

`public static XmlNMTOKEN` ***parse***(`File` *f*)
Parses a `XmlNMTOKEN` fragment from a File.

`public static XmlNMTOKEN` ***parse***(*File f, XmlOptions options*)
Parses a `XmlNMTOKEN` fragment from a File.

`public static XmlNMTOKEN` ***parse***(*URL u*)
Parses a `XmlNMTOKEN` fragment from a URL.

`public static XmlNMTOKEN` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlNMTOKEN` fragment from a URL.

`public static XmlNMTOKEN` ***parse***(*InputStream is*)
Parses a `XmlNMTOKEN` fragment from an InputStream.

`public static XmlNMTOKEN` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlNMTOKEN` fragment from an InputStream.

`public static XmlNMTOKEN` ***parse***(*Reader r*)
Parses a `XmlNMTOKEN` fragment from a Reader.

`public static XmlNMTOKEN` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlNMTOKEN` fragment from a Reader.

`public static XmlNMTOKEN` ***parse***(*Node node*)
Parses a `XmlNMTOKEN` fragment from a DOM Node.

`public static XmlNMTOKEN` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlNMTOKEN` fragment from a DOM Node.

`public static XmlNMTOKEN` ***parse***(*XMLInputStream xis*)
Parses a `XmlNMTOKEN` fragment from an XMLInputStream.

`public static XmlNMTOKEN` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlNMTOKEN` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlNMTOKEN newInstance()`

Creates an empty instance of `XmlNMTOKEN`

---

newInstance(XmlOptions) Method

```
public static XmlNMTOKEN newInstance(XmlOptions options)
```

Creates an empty instance of `XmlNMTOKEN`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlNMTOKEN newValue(Object obj)
```

Creates an immutable `XmlNMTOKEN` value

---

parse(String) Method

```
public static XmlNMTOKEN parse(String s)
     throws XmlException
```

Parses a `XmlNMTOKEN` fragment from a String. For example: "`<xml-fragment>sample-1.2</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlNMTOKEN parse(String s,
                               XmlOptions options)
     throws XmlException
```

Parses a `XmlNMTOKEN` fragment from a String. For example: "`<xml-fragment>sample-1.2</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlNMTOKEN parse(File f)
     throws XmlException, IOException
```

Parses a `XmlNMTOKEN` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlNMTOKEN parse(File f,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNMTOKEN` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlNMTOKEN parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlNMTOKEN` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlNMTOKEN parse(URL u,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNMTOKEN` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlNMTOKEN parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlNMTOKEN` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlNMTOKEN parse(InputStream is,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNMTOKEN` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlNMTOKEN parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlNMTOKEN` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlNMTOKEN parse(Reader r,
                               XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNMTOKEN` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlNMTOKEN parse(Node node)
     throws XmlException
```

Parses a `XmlNMTOKEN` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlNMTOKEN parse(Node node,
                               XmlOptions options)
     throws XmlException
```

Parses a `XmlNMTOKEN` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNMTOKEN parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlNMTOKEN` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNMTOKEN parse(XMLInputStream xis,
                               XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlNMTOKEN` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlNMTOKENS Interface

public interface ***XmlNMTOKENS***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:NMTOKENS type, a list type.

Convertible to `List`.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlNMTOKENS.Factory`** A class with methods for creating instances of `XmlNMTOKENS`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

public
List ***getListValue()***
Returns the value as a `List` of `String` values

public
List ***listValue()***
Returns the value as a `List` of `String` values

public
void ***set(List l)***
Sets the value as a `List`

public
void ***setListValue(List l)***
Sets the value as a `List`

public
List ***xgetListValue()***
Returns the value as a `List` of `XmlNMTOKEN` values

public
List ***xlistValue()***
Returns the value as a `List` of `XmlNMTOKEN` values

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getListValue() Method

`public List getListValue()`

Returns the value as a `List` of `String` values

---

listValue() Method

DEPRECATED replaced by `com.bea.xml.XmlNMTOKENS.getListValue()`

`public List listValue()`

Returns the value as a `List` of `String` values

---

set(List) Method

DEPRECATED replaced by `com.bea.xml.XmlNMTOKENS.getListValue()`

`public void set(List l)`

Sets the value as a `List`

---

setListValue(List) Method

`public void setListValue(List l)`

Sets the value as a `List`

---

xgetListValue() Method

`public List xgetListValue()`

Returns the value as a `List` of `XmlNMTOKEN` values

---

xlistValue() Method

DEPRECATED replaced by `com.bea.xml.XmlNMTOKENS.getListValue()`

`public List xlistValue()`

Returns the value as a `List` of `XmlNMTOKEN` values

XmlNMTOKENS.Factory Class

public static final class ***XmlNMTOKENS.Factory***

extends `Object`

A class with methods for creating instances of `XmlNMTOKENS`.

Hierarchy

```
Object
 XmlNMTOKENS.Factory
```

Enclosing interface

`XmlNMTOKENS`

Method Summary

`public static XmlNMTOKENS` ***newInstance***()
Creates an empty instance of `XmlNMTOKENS`

`public static XmlNMTOKENS` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlNMTOKENS`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlNMTOKENS` ***newValue***(`Object` *obj*)
Creates an immutable `XmlNMTOKENS` value

`public static XmlNMTOKENS` ***parse***(`String` *s*)
Parses a `XmlNMTOKENS` fragment from a String.

`public static XmlNMTOKENS` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlNMTOKENS` fragment from a String.

`public static XmlNMTOKENS` ***parse***(`File` *f*)
Parses a `XmlNMTOKENS` fragment from a File.

`public static XmlNMTOKENS` ***parse***(*File f, XmlOptions options*)
Parses a `XmlNMTOKENS` fragment from a File.

`public static XmlNMTOKENS` ***parse***(*URL u*)
Parses a `XmlNMTOKENS` fragment from a URL.

`public static XmlNMTOKENS` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlNMTOKENS` fragment from a URL.

`public static XmlNMTOKENS` ***parse***(*InputStream is*)
Parses a `XmlNMTOKENS` fragment from an InputStream.

`public static XmlNMTOKENS` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlNMTOKENS` fragment from an InputStream.

`public static XmlNMTOKENS` ***parse***(*Reader r*)
Parses a `XmlNMTOKENS` fragment from a Reader.

`public static XmlNMTOKENS` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlNMTOKENS` fragment from a Reader.

`public static XmlNMTOKENS` ***parse***(*Node node*)
Parses a `XmlNMTOKENS` fragment from a DOM Node.

`public static XmlNMTOKENS` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlNMTOKENS` fragment from a DOM Node.

`public static XmlNMTOKENS` ***parse***(*XMLInputStream xis*)
Parses a `XmlNMTOKENS` fragment from an XMLInputStream.

`public static XmlNMTOKENS` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlNMTOKENS` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlNMTOKENS newInstance()`

Creates an empty instance of `XmlNMTOKENS`

newInstance(XmlOptions) Method

```
public static XmlNMTOKENS newInstance(XmlOptions options)
```

Creates an empty instance of `XmlNMTOKENS`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlNMTOKENS newValue(Object obj)
```

Creates an immutable `XmlNMTOKENS` value

---

parse(String) Method

```
public static XmlNMTOKENS parse(String s)
     throws XmlException
```

Parses a `XmlNMTOKENS` fragment from a String. For example: "`<xml-fragment>sample-1.1 sample-1.2 sample-1.3</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlNMTOKENS parse(String s,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlNMTOKENS` fragment from a String. For example: "`<xml-fragment>sample-1.1 sample-1.2 sample-1.3</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlNMTOKENS parse(File f)
     throws XmlException, IOException
```

Parses a `XmlNMTOKENS` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlNMTOKENS parse(File f,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNMTOKENS` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlNMTOKENS parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlNMTOKENS` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlNMTOKENS parse(URL u,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNMTOKENS` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlNMTOKENS parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlNMTOKENS` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlNMTOKENS parse(InputStream is,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNMTOKENS` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlNMTOKENS parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlNMTOKENS` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlNMTOKENS parse(Reader r,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNMTOKENS` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlNMTOKENS parse(Node node)
     throws XmlException
```

Parses a `XmlNMTOKENS` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlNMTOKENS parse(Node node,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlNMTOKENS` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNMTOKENS parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlNMTOKENS` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNMTOKENS parse(XMLInputStream xis,
                                XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlNMTOKENS` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlNonNegativeInteger Interface

public interface ***XmlNonNegativeInteger***

extends `XmlInteger`

Corresponds to the XML Schema xs:nonNegativeInteger type. One of the derived types based on xs:decimal.

Verified to be zero or positive when validating.

Convertible to `BigInteger`.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlPositiveInteger`, `XmlUnsignedByte`, `XmlUnsignedInt`, `XmlUnsignedLong`, `XmlUnsignedShort`

Nested Class Summary

`public static final class`	**`XmlNonNegativeInteger.Factory`** A class with methods for creating instances of `XmlNonNegativeInteger`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

public static final SchemaType ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlNonNegativeInteger.Factory Class

public static final class ***XmlNonNegativeInteger.Factory***

extends `Object`

A class with methods for creating instances of `XmlNonNegativeInteger`.

Hierarchy

```
Object
 XmlNonNegativeInteger.Factory
```

Enclosing interface

`XmlNonNegativeInteger`

Method Summary

public static XmlNonNegativeInteger	***newInstance***() Creates an empty instance of `XmlNonNegativeInteger`
public static XmlNonNegativeInteger	***newInstance***(`XmlOptions` *options*) Creates an empty instance of `XmlNonNegativeInteger`
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*) Returns a validating XMLInputStream.
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*) Returns a validating XMLInputStream.
public static XmlNonNegativeInteger	***newValue***(`Object` *obj*) Creates an immutable `XmlNonNegativeInteger` value
public static XmlNonNegativeInteger	***parse***(`String` *s*) Parses a `XmlNonNegativeInteger` fragment from a String.
public static XmlNonNegativeInteger	***parse***(`String` *s*, `XmlOptions` *options*) Parses a `XmlNonNegativeInteger` fragment from a String.
public static XmlNonNegativeInteger	***parse***(`File` *f*) Parses a `XmlNonNegativeInteger` fragment from a File.

public static XmlNonNegativeInteger **parse**(*File f, XmlOptions options*)
Parses a XmlNonNegativeInteger fragment from a File.

public static XmlNonNegativeInteger **parse**(*URL u*)
Parses a XmlNonNegativeInteger fragment from a URL.

public static XmlNonNegativeInteger **parse**(*URL u, XmlOptions options*)
Parses a XmlNonNegativeInteger fragment from a URL.

public static XmlNonNegativeInteger **parse**(*InputStream is*)
Parses a XmlNonNegativeInteger fragment from an InputStream.

public static XmlNonNegativeInteger **parse**(*InputStream is, XmlOptions options*)
Parses a XmlNonNegativeInteger fragment from an InputStream.

public static XmlNonNegativeInteger **parse**(*Reader r*)
Parses a XmlNonNegativeInteger fragment from a Reader.

public static XmlNonNegativeInteger **parse**(*Reader r, XmlOptions options*)
Parses a XmlNonNegativeInteger fragment from a Reader.

public static XmlNonNegativeInteger **parse**(*Node node*)
Parses a XmlNonNegativeInteger fragment from a DOM Node.

public static XmlNonNegativeInteger **parse**(*Node node, XmlOptions options*)
Parses a XmlNonNegativeInteger fragment from a DOM Node.

public static XmlNonNegativeInteger **parse**(*XMLInputStream xis*)
Parses a XmlNonNegativeInteger fragment from an XMLInputStream.

public static XmlNonNegativeInteger **parse**(*XMLInputStream xis, XmlOptions options*)
Parses a XmlNonNegativeInteger fragment from an XMLInputStream.

Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

newInstance() Method

```
public static XmlNonNegativeInteger newInstance()
```

Creates an empty instance of `XmlNonNegativeInteger`

---

newInstance(XmlOptions) Method

```
public static XmlNonNegativeInteger newInstance(XmlOptions options)
```

Creates an empty instance of `XmlNonNegativeInteger`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlNonNegativeInteger newValue(Object obj)
```

Creates an immutable `XmlNonNegativeInteger` value

---

parse(String) Method

```
public static XmlNonNegativeInteger parse(String s)
     throws XmlException
```

Parses a `XmlNonNegativeInteger` fragment from a String. For example: "`<xml-fragment>1234567890</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlNonNegativeInteger parse(String s,
                                          XmlOptions options)
     throws XmlException
```

Parses a `XmlNonNegativeInteger` fragment from a String. For example: "`<xml-fragment>1234567890</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlNonNegativeInteger parse(File f)
     throws XmlException, IOException
```

Parses a `XmlNonNegativeInteger` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlNonNegativeInteger parse(File f,
                                          XmlOptions options)
```

```
     throws XmlException, IOException
```

Parses a `XmlNonNegativeInteger` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlNonNegativeInteger parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlNonNegativeInteger` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlNonNegativeInteger parse(URL u,
                                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNonNegativeInteger` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlNonNegativeInteger parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlNonNegativeInteger` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlNonNegativeInteger parse(InputStream is,
                                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNonNegativeInteger` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlNonNegativeInteger parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlNonNegativeInteger` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlNonNegativeInteger parse(Reader r,
                                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNonNegativeInteger` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlNonNegativeInteger parse(Node node)
     throws XmlException
```

Parses a `XmlNonNegativeInteger` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlNonNegativeInteger parse(Node node,
                                          XmlOptions options)
     throws XmlException
```

Parses a `XmlNonNegativeInteger` fragment from a DOM Node.

Exceptions

XmlException

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNonNegativeInteger parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlNonNegativeInteger` fragment from an XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNonNegativeInteger parse(XMLInputStream xis,
                                          XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlNonNegativeInteger` fragment from an XMLInputStream.

Exceptions

XmlException
XMLStreamException

com.bea.xml

XmlNonPositiveInteger Interface

public interface ***XmlNonPositiveInteger***

extends `XmlInteger`

Corresponds to the XML Schema xs:nonPositiveInteger type. One of the derived types based on xs:decimal.

Verified to be zero or negative when validating.

Convertible to `BigInteger`.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlNegativeInteger`

Nested Class Summary

`public static final class`	**`XmlNonPositiveInteger.Factory`** A class with methods for creating instances of `XmlNonPositiveInteger`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

public static final SchemaType ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlNonPositiveInteger.Factory Class

public static final class ***XmlNonPositiveInteger.Factory***

extends `Object`

A class with methods for creating instances of `XmlNonPositiveInteger`.

Hierarchy

```
Object
 XmlNonPositiveInteger.Factory
```

Enclosing interface

`XmlNonPositiveInteger`

Method Summary

public static XmlNonPositiveInteger	***newInstance***() Creates an empty instance of `XmlNonPositiveInteger`
public static XmlNonPositiveInteger	***newInstance***(`XmlOptions` *options*) Creates an empty instance of `XmlNonPositiveInteger`
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*) Returns a validating XMLInputStream.
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*) Returns a validating XMLInputStream.
public static XmlNonPositiveInteger	***newValue***(`Object` *obj*) Creates an immutable `XmlNonPositiveInteger` value
public static XmlNonPositiveInteger	***parse***(`String` *s*) Parses a `XmlNonPositiveInteger` fragment from a String.
public static XmlNonPositiveInteger	***parse***(`String` *s*, `XmlOptions` *options*) Parses a `XmlNonPositiveInteger` fragment from a String.
public static XmlNonPositiveInteger	***parse***(`File` *f*) Parses a `XmlNonPositiveInteger` fragment from a File.

public static XmlNonPositiveInteger **parse**(*File f, XmlOptions options*)
Parses a XmlNonPositiveInteger fragment from a File.

public static XmlNonPositiveInteger **parse**(*URL u*)
Parses a XmlNonPositiveInteger fragment from a URL.

public static XmlNonPositiveInteger **parse**(*URL u, XmlOptions options*)
Parses a XmlNonPositiveInteger fragment from a URL.

public static XmlNonPositiveInteger **parse**(*InputStream is*)
Parses a XmlNonPositiveInteger fragment from an InputStream.

public static XmlNonPositiveInteger **parse**(*InputStream is, XmlOptions options*)
Parses a XmlNonPositiveInteger fragment from an InputStream.

public static XmlNonPositiveInteger **parse**(*Reader r*)
Parses a XmlNonPositiveInteger fragment from a Reader.

public static XmlNonPositiveInteger **parse**(*Reader r, XmlOptions options*)
Parses a XmlNonPositiveInteger fragment from a Reader.

public static XmlNonPositiveInteger **parse**(*Node node*)
Parses a XmlNonPositiveInteger fragment from a DOM Node.

public static XmlNonPositiveInteger **parse**(*Node node, XmlOptions options*)
Parses a XmlNonPositiveInteger fragment from a DOM Node.

public static XmlNonPositiveInteger **parse**(*XMLInputStream xis*)
Parses a XmlNonPositiveInteger fragment from an XMLInputStream.

public static XmlNonPositiveInteger **parse**(*XMLInputStream xis, XmlOptions options*)
Parses a XmlNonPositiveInteger fragment from an XMLInputStream.

Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

newInstance() Method

```
public static XmlNonPositiveInteger newInstance()
```

Creates an empty instance of `XmlNonPositiveInteger`

---

newInstance(XmlOptions) Method

```
public static XmlNonPositiveInteger newInstance(XmlOptions options)
```

Creates an empty instance of `XmlNonPositiveInteger`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlNonPositiveInteger newValue(Object obj)
```

Creates an immutable `XmlNonPositiveInteger` value

---

parse(String) Method

```
public static XmlNonPositiveInteger parse(String s)
     throws XmlException
```

Parses a `XmlNonPositiveInteger` fragment from a String. For example: "`<xml-fragment>-1234567890</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlNonPositiveInteger parse(String s,
                                          XmlOptions options)
     throws XmlException
```

Parses a `XmlNonPositiveInteger` fragment from a String. For example: "`<xml-fragment>-1234567890</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlNonPositiveInteger parse(File f)
     throws XmlException, IOException
```

Parses a `XmlNonPositiveInteger` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlNonPositiveInteger parse(File f,
                                          XmlOptions options)
```

```
     throws XmlException, IOException
```

Parses a `XmlNonPositiveInteger` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlNonPositiveInteger parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlNonPositiveInteger` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlNonPositiveInteger parse(URL u,
                                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNonPositiveInteger` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlNonPositiveInteger parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlNonPositiveInteger` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlNonPositiveInteger parse(InputStream is,
                                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNonPositiveInteger` fragment from an InputStream.

Exceptions

`XmlException`
`IOException`

---

parse(Reader) Method

```
public static XmlNonPositiveInteger parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlNonPositiveInteger` fragment from a Reader.

Exceptions

`XmlException`
`IOException`

---

parse(Reader, XmlOptions) Method

```
public static XmlNonPositiveInteger parse(Reader r,
                                          XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNonPositiveInteger` fragment from a Reader.

Exceptions

`XmlException`
`IOException`

---

parse(Node) Method

```
public static XmlNonPositiveInteger parse(Node node)
     throws XmlException
```

Parses a `XmlNonPositiveInteger` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlNonPositiveInteger parse(Node node,
                                          XmlOptions options)
     throws XmlException
```

Parses a `XmlNonPositiveInteger` fragment from a DOM Node.

Exceptions

XmlException

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNonPositiveInteger parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlNonPositiveInteger` fragment from an XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNonPositiveInteger parse(XMLInputStream xis,
                                          XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlNonPositiveInteger` fragment from an XMLInputStream.

Exceptions

XmlException
XMLStreamException

XmlNormalizedString Interface

public interface ***XmlNormalizedString***

extends `XmlString`

Corresponds to the XML Schema xs:normalizedString type. One of the derived types based on xs:string.

An normalizedString simply is a string where all the carriage return, linefeed, and tab characters have been normalized (switched to) ordinary space characters. Use normalizedString for long strings to make them insensitive to line breaking. If you wish to often be insensitive to runs of whitespace (as is often the case), use xs:token (aka `XmlToken`) instead.

Convertible to `String`. When obtaining the stringValue, the whitespace−normalized value is returned.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlString`, `XmlTokenSource`

All Known Subinterfaces

`XmlENTITY`, `XmlID`, `XmlIDREF`, `XmlLanguage`, `XmlName`, `XmlNCName`, `XmlNMTOKEN`, `XmlToken`

Nested Class Summary

`public static final class`	**`XmlNormalizedString.Factory`** A class with methods for creating instances of `XmlNormalizedString`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Field Summary

public static final SchemaType ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlNormalizedString.Factory Class

public static final class ***XmlNormalizedString.Factory***

extends `Object`

A class with methods for creating instances of `XmlNormalizedString`.

Hierarchy

```
Object
 XmlNormalizedString.Factory
```

Enclosing interface

`XmlNormalizedString`

Method Summary

`public static XmlNormalizedString` ***newInstance***()
Creates an empty instance of `XmlNormalizedString`

`public static XmlNormalizedString` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlNormalizedString`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlNormalizedString` ***newValue***(`Object` *obj*)
Creates an immutable `XmlNormalizedString` value

`public static XmlNormalizedString` ***parse***(`String` *s*)
Parses a `XmlNormalizedString` fragment from a String.

`public static XmlNormalizedString` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlNormalizedString` fragment from a String.

`public static XmlNormalizedString` ***parse***(`File` *f*)
Parses a `XmlNormalizedString` fragment from a File.

public static XmlNormalizedString	***parse***(*File f, XmlOptions options*) Parses a XmlNormalizedString fragment from a File.
public static XmlNormalizedString	***parse***(*URL u*) Parses a XmlNormalizedString fragment from a URL.
public static XmlNormalizedString	***parse***(*URL u, XmlOptions options*) Parses a XmlNormalizedString fragment from a URL.
public static XmlNormalizedString	***parse***(*InputStream is*) Parses a XmlNormalizedString fragment from an InputStream.
public static XmlNormalizedString	***parse***(*InputStream is, XmlOptions options*) Parses a XmlNormalizedString fragment from an InputStream.
public static XmlNormalizedString	***parse***(*Reader r*) Parses a XmlNormalizedString fragment from a Reader.
public static XmlNormalizedString	***parse***(*Reader r, XmlOptions options*) Parses a XmlNormalizedString fragment from a Reader.
public static XmlNormalizedString	***parse***(*Node node*) Parses a XmlNormalizedString fragment from a DOM Node.
public static XmlNormalizedString	***parse***(*Node node, XmlOptions options*) Parses a XmlNormalizedString fragment from a DOM Node.
public static XmlNormalizedString	***parse***(*XMLInputStream xis*) Parses a XmlNormalizedString fragment from an XMLInputStream.
public static XmlNormalizedString	***parse***(*XMLInputStream xis, XmlOptions options*) Parses a XmlNormalizedString fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlNormalizedString newInstance()`

Creates an empty instance of `XmlNormalizedString`

newInstance(XmlOptions) Method

```
public static XmlNormalizedString newInstance(XmlOptions options)
```

Creates an empty instance of `XmlNormalizedString`

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

newValue(Object) Method

```
public static XmlNormalizedString newValue(Object obj)
```

Creates an immutable `XmlNormalizedString` value

parse(String) Method

```
public static XmlNormalizedString parse(String s)
     throws XmlException
```

Parses a `XmlNormalizedString` fragment from a String. For example: "`<xml-fragment> string to normalize </xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlNormalizedString parse(String s,
                                        XmlOptions options)
     throws XmlException
```

Parses a `XmlNormalizedString` fragment from a String. For example: "`<xml-fragment> string to normalize </xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlNormalizedString parse(File f)
     throws XmlException, IOException
```

Parses a `XmlNormalizedString` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlNormalizedString parse(File f,
                                        XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNormalizedString` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlNormalizedString parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlNormalizedString` fragment from a URL.

Exceptions

`XmlException`
`IOException`

---

parse(URL, XmlOptions) Method

```
public static XmlNormalizedString parse(URL u,
                                        XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNormalizedString` fragment from a URL.

Exceptions

`XmlException`
`IOException`

---

parse(InputStream) Method

```
public static XmlNormalizedString parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlNormalizedString` fragment from an InputStream.

Exceptions

`XmlException`
`IOException`

---

parse(InputStream, XmlOptions) Method

```
public static XmlNormalizedString parse(InputStream is,
                                        XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNormalizedString` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlNormalizedString parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlNormalizedString` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlNormalizedString parse(Reader r,
                                        XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNormalizedString` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlNormalizedString parse(Node node)
     throws XmlException
```

Parses a `XmlNormalizedString` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlNormalizedString parse(Node node,
                                        XmlOptions options)
     throws XmlException
```

Parses a `XmlNormalizedString` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNormalizedString parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlNormalizedString` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNormalizedString parse(XMLInputStream xis,
                                        XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlNormalizedString` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

XmlNOTATION Interface

public interface ***XmlNOTATION***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:NOTATION type.

NOTATIONs are a mechanism in XML Schema that is provided for partial backward compatibility with NOTATIONs in DTDs. XML Schema users should almost never have any reason to use this data type.

Convertible to `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlNOTATION.Factory`** A class with methods for creating instances of `XmlNOTATION`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlNOTATION.Factory Class

public static final class ***XmlNOTATION.Factory***

extends `Object`

A class with methods for creating instances of `XmlNOTATION`.

Hierarchy

```
Object
 XmlNOTATION.Factory
```

Enclosing interface

`XmlNOTATION`

Method Summary

`public static XmlNOTATION` ***newInstance***()
Creates an empty instance of `XmlNOTATION`

`public static XmlNOTATION` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlNOTATION`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlNOTATION` ***newValue***(`Object` *obj*)
Creates an immutable `XmlNOTATION` value

`public static XmlNOTATION` ***parse***(`String` *s*)
Parses a `XmlNOTATION` fragment from a String.

`public static XmlNOTATION` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlNOTATION` fragment from a String.

`public static XmlNOTATION` ***parse***(`File` *f*)
Parses a `XmlNOTATION` fragment from a File.

`public static XmlNOTATION` ***parse***(*File f, XmlOptions options*)
Parses a `XmlNOTATION` fragment from a File.

`public static XmlNOTATION` ***parse***(*URL u*)
Parses a `XmlNOTATION` fragment from a URL.

`public static XmlNOTATION` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlNOTATION` fragment from a URL.

`public static XmlNOTATION` ***parse***(*InputStream is*)
Parses a `XmlNOTATION` fragment from an InputStream.

`public static XmlNOTATION` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlNOTATION` fragment from an InputStream.

`public static XmlNOTATION` ***parse***(*Reader r*)
Parses a `XmlNOTATION` fragment from a Reader.

`public static XmlNOTATION` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlNOTATION` fragment from a Reader.

`public static XmlNOTATION` ***parse***(*Node node*)
Parses a `XmlNOTATION` fragment from a DOM Node.

`public static XmlNOTATION` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlNOTATION` fragment from a DOM Node.

`public static XmlNOTATION` ***parse***(*XMLInputStream xis*)
Parses a `XmlNOTATION` fragment from an XMLInputStream.

`public static XmlNOTATION` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlNOTATION` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlNOTATION newInstance()`

Creates an empty instance of `XmlNOTATION`

---

newInstance(XmlOptions) Method

```
public static XmlNOTATION newInstance(XmlOptions options)
```

Creates an empty instance of `XmlNOTATION`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlNOTATION newValue(Object obj)
```

Creates an immutable `XmlNOTATION` value

---

parse(String) Method

```
public static XmlNOTATION parse(String s)
     throws XmlException
```

Parses a `XmlNOTATION` fragment from a String.

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlNOTATION parse(String s,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlNOTATION` fragment from a String.

Exceptions

XmlException

---

parse(File) Method

```
public static XmlNOTATION parse(File f)
     throws XmlException, IOException
```

Parses a `XmlNOTATION` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlNOTATION parse(File f,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNOTATION` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlNOTATION parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlNOTATION` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlNOTATION parse(URL u,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNOTATION` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlNOTATION parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlNOTATION` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlNOTATION parse(InputStream is,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNOTATION` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlNOTATION parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlNOTATION` fragment from a Reader.

Exceptions

`XmlException`
`IOException`

---

parse(Reader, XmlOptions) Method

```
public static XmlNOTATION parse(Reader r,
                                XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlNOTATION` fragment from a Reader.

Exceptions

`XmlException`
`IOException`

---

parse(Node) Method

```
public static XmlNOTATION parse(Node node)
     throws XmlException
```

Parses a `XmlNOTATION` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(Node, XmlOptions) Method

```
public static XmlNOTATION parse(Node node,
                                XmlOptions options)
     throws XmlException
```

Parses a `XmlNOTATION` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNOTATION parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlNOTATION` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlNOTATION parse(XMLInputStream xis,
                                XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlNOTATION` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlObject Interface

public interface ***XmlObject***

extends `XmlTokenSource`

Corresponds to the XML Schema xs:anyType, the base type for all XML Beans.

Since all XML Schema types are translated into corresponding XML Bean classes, and all Schema type derivation corresponds to Java class inheritance, the fact that all Schema types derive from xs:anyType means that all XML Bean classes derive from XmlObject.

On this base class you will find a number of common facilities that all XML Bean classes provide:

- Every XML Bean class has an inner Factory class for creating and parsing instances, including XmlObject. Use `XmlObject.Factory` itself to produce untyped XML trees or XML trees that implement specific subtypes of XmlObject depending on a recognized root document element. If you depend on the automatic type inference, you will want to understand the type inference issues described below.
- To write out an accurate XML file for the XML tree under this XmlObject, use one of the `XmlObject.save(ContentHandler, LexicalHandler)` methods, or `XmlObject.newInputStream()` or `XmlObject.newReader()`. Use `XmlObject.toString()` to produce a pretty−printed representation of the XML subtree under this XML Object. If you save interior portions of an XML document, you will want to understand the inner contents versus outer container issues described below.
- It is also simple to copy an XmlObject instance to or from a standard DOM tree or SAX stream. Use `XmlObject.Factory.parse(Node)`, for example, to load from DOM; use `XmlObject.Factory.newXmlSaxHandler()` to load from SAX; use `XmlObject.newDomNode()` to save to DOM; and use `XmlObject.save(ContentHandler, LexicalHandler)` to save to SAX.
- Use `XmlObject.validate()` to validate the subtree of XML under this XML Object. If you wish to get information about the location and reason for validation errors, see `XmlOptions.setErrorListener(Collection)`, and use `XmlObject.validate(XmlOptions)`.
- Use `XmlObject.newCursor()` to access the full XML infoset, for example, if you need to determine interleaved element order or manipulate annotations, comments, or mixed content. See `XmlCursor`.
- Use `XmlObject.selectPath(String)` to find other XmlObjects in the subtree underneath this XmlObject using relative XPaths. (In selectPath, "." indicates the current element or attribute.)

Type inference. When using `XmlObject.Factory` to parse XML documents, the actual document type is not `XmlObject.type` itself, but a subtype based on the contents of the parsed document. If the parsed document contains a recognized root document element, then the actual type of the loaded instance will be the matching Document type. For example:

```
XmlObject xobj = XmlObject.Factory.parse(myDocument);
if (xobj instanceof MyOrderDocument) // starts w/ <my-order>
{
```

```
    MyOrderDocument mydoc = (MyOrderDocument)xobj;
    if (!xobj.validate())
        System.out.println("Not a valid my-order document");
}
else
{
    System.out.println("Not a my-order document");
}
```

Every XML Bean class has its own inner Factory class, so if you actually know exactly which XML Bean document type you want to load as in the example above, you should use the the specific XML Bean Factory class instead. For example:

```
MyOrderDocument mydoc = MyOrderDocument.Factory.parse(myDocument);
```

The code above will throw an exception if the parsed document does not begin with the proper (my-order) element.

Inner versus outer. An XmlObject represents the *contents* of an element or attribute, *not* the element or attribute itself. So when you validate or save an XmlObject, you are validating or saving its contents, not its container. For example, if the XmlObject represents the contents of an element which happens to itself be in the wrong order relative to its siblings, validate will not complain about the misplacement of the element itself. On the other hand, if elements *within* the XmlObject are in the wrong order, validate will complain. Similarly, when saving the contents of an interior XmlObject, it is the contents of an element, not the element itself, which is saved by default.

Reading and writing fragments. When reading or writing the contents of a whole XML document, the standard XML reprentation for a document is used. However, there is no standard concrete XML representation for "just the contents" of an interior element or attribute. So when one is needed, the tag <xml-fragment> is used to wrap the contents. This tag is used can also be used to load just the contents for an XmlObject document fragment of arbitrary type. If you wish to save out the XmlObject's container element along with its contents, use `XmlOptions.setSaveOuter()`.

Implementing XmlObject. The XMLBeans library does not support arbitrary implementations of XmlObject - in almost all cases, you should only use the implementations of XmlObject provided by the XMLBeans compiler itself. If you need to implement XmlObject yourself, you should subclass FilterXmlObject in order to delegate to another underlying XmlObject implementation. This technique will allow you to use your code unchanged with future versions of XMLBeans that add additional methods on XmlObject.

All Known Implementing Classes

`FilterXmlObject`

All Superinterfaces

`XmlTokenSource`

All Known Subinterfaces

`SimpleValue`, `XmlAnySimpleType`, `XmlAnyURI`, `XmlBase64Binary`, `XmlBoolean`, `XmlByte`, `XmlDate`, `XmlDateTime`, `XmlDecimal`, `XmlDouble`, `XmlDuration`,

`XmlENTITIES`, `XmlENTITY`, `XmlFloat`, `XmlGDay`, `XmlGMonth`, `XmlGMonthDay`, `XmlGYear`, `XmlGYearMonth`, `XmlHexBinary`, `XmlID`, `XmlIDREF`, `XmlIDREFS`, `XmlInt`, `XmlInteger`, `XmlLanguage`, `XmlLong`, `XmlName`, `XmlNCName`, `XmlNegativeInteger`, `XmlNMTOKEN`, `XmlNMTOKENS`, `XmlNonNegativeInteger`, `XmlNonPositiveInteger`, `XmlNormalizedString`, `XmlNOTATION`, `XmlPositiveInteger`, `XmlQName`, `XmlShort`, `XmlString`, `XmlTime`, `XmlToken`, `XmlUnsignedByte`, `XmlUnsignedInt`, `XmlUnsignedLong`, `XmlUnsignedShort`

Nested Class Summary

`public static final class`	**`XmlObject.Factory`** Static factory class for creating new instances.

Field Summary

`public static final int`	***`EQUAL`*** EQUAL is 0.
`public static final int`	***`GREATER_THAN`*** GREATER_THAN is 1.
`public static final int`	***`LESS_THAN`*** LESS_THAN is −1.
`public static final int`	***`NOT_EQUAL`*** NOT_EQUAL is 2.
`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Method Summary

`public XmlObject`	***`changeType`***`(SchemaType` *newType)* Changes the schema type associated with this data and returns a new XmlObject instance whose schemaType is the new type.
`public int`	***`compareTo`***`(Object` *obj)* Impelements the Comparable interface by comparing two simple xml values based on their standard XML schema ordering.
`public int`	***`compareValue`***`(XmlObject` *obj)*

This comparison method is similar to compareTo, but rather than throwing a ClassCastException when two values are incomparable, it returns the number 2.

`public XmlObject` ***copy()***

Returns a deep copy of this XmlObject.

`public XmlObject[]` ***execQuery***(`String` *query*)

Executes a query.

`public XmlObject[]` ***execQuery***(`String` *query*, `XmlOptions` *options*)

Executes a query with options.

`public boolean` ***isImmutable()***

True if the value is an immutable value.

`public boolean` ***isNil()***

True if the value is nil.

`public SchemaType` ***schemaType()***

The schema type for this instance.

`public XmlObject[]` ***selectPath***(`String` *path*)

Selects a path.

`public XmlObject[]` ***selectPath***(`String` *path*, `XmlOptions` *options*)

Selects a path, applying options.

`public XmlObject` ***set***(`XmlObject` *srcObj*)

Set the value/type of this XmlObject to be a copy of the source XmlObject.

`public void` ***setNil()***

Sets the value to nil.

`public String` ***toString()***

Returns an XML string for this XML object.

`public boolean` ***validate()***

Returns true if the contents of this object are valid accoring to schemaType().

`public boolean` ***validate***(`XmlOptions` *options*)

Just like validate(), but with options.

`public boolean` ***valueEquals***(`XmlObject` *obj*)

True if the xml values are equal.

`public int` ***valueHashCode()***

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

EQUAL

`public static final int EQUAL`

EQUAL is 0. See `XmlObject.compareValue(XmlObject)`.

GREATER_THAN

`public static final int GREATER_THAN`

GREATER_THAN is 1. See `XmlObject.compareValue(XmlObject)`.

LESS_THAN

`public static final int LESS_THAN`

LESS_THAN is −1. See `XmlObject.compareValue(XmlObject)`.

NOT_EQUAL

`public static final int NOT_EQUAL`

NOT_EQUAL is 2. See `XmlObject.compareValue(XmlObject)`.

---

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

changeType(SchemaType) Method

public XmlObject changeType(SchemaType *newType***)**

Changes the schema type associated with this data and returns a new XmlObject instance whose schemaType is the new type.

Returns null if the type change is not allowed. Certain type changes may be prohibited on the interior of an xml tree due to schema type system constraints (that is, due to a parent container within which the newly specified type is not permissible), but there are no constraints at the roottype changes are never prohibited at the root of an xml tree.

If the type change is allowed, then the new XmlObject should be used rather than the old one. The old XmlObject instance and any other XmlObject instances in the subtree are permanently invalidated and should not be used. (They will return InvalidStateException if you try to use them.) If a type change is done on the interior of an Xml tree, then xsi:type attributes are updated as needed.

---

compareTo(Object) Method

public int compareTo(Object *obj***)**

Impelements the Comparable interface by comparing two simple xml values based on their standard XML schema ordering. Throws a ClassCastException if no standard ordering applies, or if the two values are incomparable within a partial order.

---

compareValue(XmlObject) Method

public int compareValue(XmlObject *obj***)**

This comparison method is similar to compareTo, but rather than throwing a ClassCastException when two values are incomparable, it returns the number 2. The result codes are −1 if this object is less than obj, 1 if this object is greater than obj, zero if the objects are equal, and 2 if the objects are incomparable.

---

copy() Method

public XmlObject copy()

Returns a deep copy of this XmlObject. The returned object has the same type as the current object, and has all the content of the XML document underneath the current object. Note that any parts of the XML document above or outside this XmlObject are not copied.

---

execQuery(String) Method

```
public XmlObject[] execQuery(String query)
```

Executes a query. Query can be a string or precompiled query String.

An XQuery is very similar to an XPath, except that it also permits construction of new XML. As a result, the XmlObjects that are returned from execQuery are in newly created documents, separate from the XmlObject on which the query is executed.

Syntax and usage is otherwise similar to selectPath.

Related Topics

`XmlObject.selectPath(String)`

---

execQuery(String, XmlOptions) Method

```
public XmlObject[] execQuery(String query,
                             XmlOptions options)
```

Executes a query with options. Use the *options* parameter to specify the following:

To specify this	Use this method
The document type for the root element.	`XmlOptions.setDocumentType(SchemaType)`
To replace the document element with the specified QName when constructing the resulting document.	`XmlOptions.setLoadReplaceDocumentElement(QName)`
To strip all insignificant whitespace when constructing a document.	`XmlOptions.setLoadStripWhitespace()`
To strip all comments when constructing a document.	`XmlOptions.setLoadStripComments()`
To strip all processing instructions when constructing a document.	`XmlOptions.setLoadStripProcinsts()`
A map of namespace URI substitutions to use when constructing a document.	`XmlOptions.setLoadSubstituteNamespaces(Map)`
Additional namespace mappings to be added when constructing a document.	`XmlOptions.setLoadAdditionalNamespaces(Map)`
To trim the underlying XML text buffer immediately after constructing a document, resulting in a smaller memory footprint.	`XmlOptions.setLoadTrimTextBuffer()`
Whether value facets should be checked as they are set.	`XmlOptions.setValidateOnSet()`

Parameters

query
The XQuery expression.

options
Options as described.

Related Topics

`XmlObject.execQuery(String)`

---

isImmutable() Method

`public boolean isImmutable()`

True if the value is an immutable value. Immutable values do not have a position in a tree; rather, they are stand−alone simple type values. If the object is immutable, the equals() methods tests for value equality, and the object can be used as the key for a hash.

---

isNil() Method

`public boolean isNil()`

True if the value is nil. Note that in order to be nil, the value must be in an element, and the element containing the value must be marked as nillable in the schema.

---

schemaType() Method

`public SchemaType schemaType()`

The schema type for this instance. This is a permanent, unchanging property of the instance.

---

selectPath(String) Method

`public XmlObject[] selectPath(String` *`path`***`)`**

Selects a path. Path can be a string or precompiled path String.

The path must be a relative path, where "." represents the element or attribute containg this XmlObject, and it must select only other elements or attributes. If a non−element or non−attribute is selected, an unchecked exception is thrown.

The array that is returned contains all the selected XmlObjects, within the same document, listed in document order. The actual array type of the result is inferred from the closest common base type of selected results.

Here is an example of usage. Suppose we have a global element definition for "owner" whose type is "person":

```
<schema targetNamespace="http://openuri.org/sample">
   <element name="owner" type="person"/>
   <complexType name="person">
      [...]
   </complexType>
</schema>
```

and suppose "owner" tags can be scattered throughout the document. Then we can write the following code to find them all:

```
import org.openuri.sample.Person;
import com.bea.xml.*;
[...]
  XmlObject xobj = XmlObject.Factory.parse(myFile);
  Person[] results;
  results = (Person[])xobj.selectPath(
     "declare namespace s='http://www.openuri.org/sample' " +
     ".//s:owner");
```

Notice the way in which namespace declarations are done in XPath 2.0. Since XPath can only navigate within an XML document − it cannot construct new XML − the resulting XmlObjects all reside in the same XML document as this XmlObject itself.

---

selectPath(String, XmlOptions) Method

```
public XmlObject[] selectPath(String path,
                              XmlOptions options)
```

Selects a path, applying options.

Related Topics

`XmlObject.selectPath(String)`

---

set(XmlObject) Method

```
public XmlObject set(XmlObject srcObj)
```

Set the value/type of this XmlObject to be a copy of the source XmlObject. Because the type of the source may be different than this target, this XmlObject may become defunct. In this case the new XmlObject is returned. If no type change happens, the same this will be returned.

---

setNil() Method

public void setNil()

Sets the value to nil. The element containing the value must be marked as nillable in the schema.

---

toString() Method

public String toString()

Returns an XML string for this XML object.

The string is pretty−printed. If you want a non−pretty−printed string, or if you want to control options precisely, use the xmlText() methods.

Note that when producing XML any object other than very root of the document, then you are guaranteed to be looking at only a fragment of XML, i.e., just the contents of an element or attribute, and and we will produce a string that starts with an `<xml−fragment>` tag. The XmlOptions.setSaveOuter() option on xmlText can be used to produce the actual element name above the object if you wish.

Overrides

`Object.toString()`

---

validate() Method

public boolean validate()

Returns true if the contents of this object are valid accoring to schemaType().

Does a deep validation of the entire subtree under the object, but does not validate the parents or siblings of the object if the object is in the interior of an xml tree.

---

validate(XmlOptions) Method

public boolean validate(XmlOptions *options*)

Just like validate(), but with options.

If you wish to collect error messages and locations while validating, use the `XmlOptions.setErrorListener(Collection)` method. With that method, you can specify an object in which to store messages related to validation. The following is a simple example.

```
// Create an XmlOptions instance and set the error listener.
XmlOptions validateOptions = new XmlOptions();
ArrayList errorList = new ArrayList();
validateOptions.setErrorListener(errorList);

// Validate the XML.
boolean isValid = newEmp.validate(validateOptions);
```

```
// If the XML isn't valid, loop through the listener's contents,
// printing contained messages.
if (!isValid)
{
     for (int i = 0; i <errorList.size(); i++)
     {
         XmlError error = (XmlError)errorList.get(i);

         System.out.println("\n");
         System.out.println("Message: " + error.getMessage() + "\n");
         System.out.println("Location of invalid XML: " +
             error.getCursorLocation().xmlText() + "\n");
     }
}
```

Parameters

options
An object that implements the `Collection` interface.

---

valueEquals(XmlObject) Method

`public boolean valueEquals(XmlObject` *`obj`*`)`

True if the xml values are equal. Two different objects (which are distinguished by equals(obj) == false) may of course have equal values (valueEquals(obj) == true).

Usually this method can be treated as an ordinary equvalence relation, but actually it is not is not transitive. Here is a precise specification:

There are two categories of XML object: objects with a known instance type, and objects whose only known type is one of the ur−types (either AnyType or AnySimpleType). The first category is compared in terms of logical value spaces, and the second category is compared lexically.

Within each of these two categories, valueEquals is a well−behaved equivalence relation. However, when comparing an object of known type with an object with ur−type, the comparison is done by attempting to convert the lexical form of the ur−typed object into the other type, and then comparing the results. Ur−typed objects are therefore treated as lexical wildcards and may be equal to objects in different value spaces, even though the objects in different value spaces are not equal to each other.

For example, the anySimpleType value "1" will compare as an equalValue to the string "1", the float value "1.0", the double value "1.0", the decimal "1", and the GYear "1", even though all these objects will compare unequal to each other since they lie in different value spaces.

---

valueHashCode() Method

`public int valueHashCode()`

XmlObject.Factory Class

public static final class ***XmlObject.Factory***

extends `Object`

Static factory class for creating new instances. Note that if a type can be inferred from the XML being loaded (for example, by recognizing the document element QName), then the instance returned by a factory will have the inferred type. Otherwise the Factory will returned an untyped document.

Hierarchy

```
Object
 XmlObject.Factory
```

Enclosing interface

`XmlObject`

Method Summary

`public static XmlObject` ***newInstance***()
Creates a new, completely empty instance.

`public static XmlObject` ***newInstance***(`XmlOptions` *options*)
Creates a new, completely empty instance, specifying options for the root element's document type and/or whether to validate value facets as they are set.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a new validating `XMLInputStream` that throws exceptions when the input is not valid.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a new validating `XMLInputStream` that throws exceptions when the input is not valid, specifying options for the root element's document type and/or the collection object to use as an error listener while validating.

`public static XmlObject` ***newValue***(`Object` *obj*)
Creates an immutable `XmlObject` value

`public static XmlSaxHandler` ***newXmlSaxHandler***()
Returns an `XmlSaxHandler` that can load an XmlObject from SAX events.

`public static XmlSaxHandler` ***newXmlSaxHandler***(*XmlOptions options*)
Returns an `XmlSaxHandler` that can load an XmlObject from SAX events.

`public static XmlObject` ***parse***(*String s*)
Parses the given `String` as XML.

`public static XmlObject` ***parse***(*String s, XmlOptions options*)
Parses the given `String` as XML.

`public static XmlObject` ***parse***(*File f*)
Parses the given `File` as XML.

`public static XmlObject` ***parse***(*File f, XmlOptions options*)
Parses the given `File` as XML.

`public static XmlObject` ***parse***(*URL u*)
Downloads the given `URL` as XML.

`public static XmlObject` ***parse***(*URL u, XmlOptions options*)
Downloads the given `URL` as XML.

`public static XmlObject` ***parse***(*InputStream is*)
Decodes and parses the given `InputStream` as XML.

`public static XmlObject` ***parse***(*InputStream is, XmlOptions options*)
Decodes and parses the given `InputStream` as XML.

`public static XmlObject` ***parse***(*Reader r*)
Parses the given `Reader` as XML.

`public static XmlObject` ***parse***(*Reader r, XmlOptions options*)
Parses the given `Reader` as XML.

`public static XmlObject` ***parse***(*Node node*)
Converts the given DOM `Node` into an XmlObject.

`public static XmlObject` ***parse***(*Node node, XmlOptions options*)
Converts the given DOM `Node` into an XmlObject.

`public static XmlObject` ***parse***(*XMLInputStream xis*)
Loads the given `XMLInputStream` into an XmlObject.

`public static XmlObject` ***parse***(*XMLInputStream xis, XmlOptions options*)
Loads the given `XMLInputStream` into an XmlObject.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlObject newInstance()`

Creates a new, completely empty instance.

---

newInstance(XmlOptions) Method

`public static XmlObject newInstance(XmlOptions options)`

Creates a new, completely empty instance, specifying options for the root element's document type and/or whether to validate value facets as they are set.

Use the *options* parameter to specify the following:

To specify this	Use this method
The document type for the root element.	`XmlOptions.setDocumentType(SchemaType)`
Whether value facets should be checked as they are set.	`XmlOptions.setValidateOnSet()`

Parameters

options
Options specifying root document type and/or value facet checking.

Returns

A new, empty instance of XmlObject.

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

`public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)`
`throws XmlException, XMLStreamException`

Returns a new validating `XMLInputStream` that throws exceptions when the input is not valid.

Exceptions

`XmlException`
`XMLStreamException`

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a new validating `XMLInputStream` that throws exceptions when the input is not valid, specifying options for the root element's document type and/or the collection object to use as an error listener while validating.

Use the *options* parameter to specify the following:

- A collection instance that should be used as an error listener during compilation, as described in `XmlOptions.setErrorListener(Collection)`.
- The document type for the root element, as described in `XmlOptions.setDocumentType(SchemaType)`.

Parameters

xis
: The basis for the new XMLInputStream.

options
: Options specifying root document type and/or an error listener.

Returns

A new validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

newValue(Object) Method

```
public static XmlObject newValue(Object obj)
```

Creates an immutable `XmlObject` value

newXmlSaxHandler() Method

```
public static XmlSaxHandler newXmlSaxHandler()
```

Returns an `XmlSaxHandler` that can load an XmlObject from SAX events.

newXmlSaxHandler(XmlOptions) Method

```
public static XmlSaxHandler newXmlSaxHandler(XmlOptions options)
```

Returns an `XmlSaxHandler` that can load an XmlObject from SAX events.

parse(String) Method

```
public static XmlObject parse(String s)
     throws XmlException
```

Parses the given `String` as XML.

Exceptions

`XmlException`

parse(String, XmlOptions) Method

```
public static XmlObject parse(String s,
                              XmlOptions options)
     throws XmlException
```

Parses the given `String` as XML. Use the *options* parameter to specify the following:

To specify this	Use this method
The document type for the root element.	`XmlOptions.setDocumentType(SchemaType)`
To place line number annotations in the store when parsing a document.	`XmlOptions.setLoadLineNumbers()`
To replace the document element with the specified QName when parsing.	`XmlOptions.setLoadReplaceDocumentElement(QName)`
To strip all insignificant whitespace when parsing a document.	`XmlOptions.setLoadStripWhitespace()`
To strip all comments when parsing a document.	`XmlOptions.setLoadStripComments()`
To strip all processing instructions when parsing a document.	`XmlOptions.setLoadStripProcinsts()`
A map of namespace URI substitutions to use when parsing a document.	`XmlOptions.setLoadSubstituteNamespaces(Map)`
Additional namespace mappings to be added when parsing a document.	`XmlOptions.setLoadAdditionalNamespaces(Map)`
To trim the underlying XML text buffer immediately after parsing a	`XmlOptions.setLoadTrimTextBuffer()`

document, resulting in a smaller memory footprint.

Parameters

s
The string to parse.

options
Options as specified.

Returns

A new instance containing the specified XML.

Exceptions

XmlException

---

parse(File) Method

```
public static XmlObject parse(File f)
     throws XmlException, IOException
```

Parses the given `File` as XML.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlObject parse(File f,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses the given `File` as XML.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlObject parse(URL u)
     throws XmlException, IOException
```

Downloads the given URL as XML.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlObject parse(URL u,
                              XmlOptions options)
     throws XmlException, IOException
```

Downloads the given URL as XML.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlObject parse(InputStream is)
     throws XmlException, IOException
```

Decodes and parses the given `InputStream` as XML.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlObject parse(InputStream is,
                              XmlOptions options)
     throws XmlException, IOException
```

Decodes and parses the given `InputStream` as XML. Use the *options* parameter to specify the following:

To specify this	Use this method
The character encoding to use when parsing or writing a document.	`XmlOptions.setCharacterEncoding(String)`
The document type for the root element.	`XmlOptions.setDocumentType(SchemaType)`
Place line number annotations in the store when parsing a document.	`XmlOptions.setLoadLineNumbers()`

Replace the document element with the specified QName when parsing.	`XmlOptions.setLoadReplaceDocumentElement(QName)`
Strip all insignificant whitespace when parsing a document.	`XmlOptions.setLoadStripWhitespace()`
Strip all comments when parsing a document.	`XmlOptions.setLoadStripComments()`
Strip all processing instructions when parsing a document.	`XmlOptions.setLoadStripProcinsts()`
Set a map of namespace URI substitutions to use when parsing a document.	`XmlOptions.setLoadSubstituteNamespaces(Map)`
Set additional namespace mappings to be added when parsing a document.	`XmlOptions.setLoadAdditionalNamespaces(Map)`
Trim the underlying XML text buffer immediately after parsing a document, resulting in a smaller memory footprint.	`XmlOptions.setLoadTrimTextBuffer()`

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlObject parse(Reader r)
     throws XmlException, IOException
```

Parses the given `Reader` as XML.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlObject parse(Reader r,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses the given `Reader` as XML.

Exceptions

XmlException
IOException

parse(Node) Method

```
public static XmlObject parse(Node node)
     throws XmlException
```

Converts the given DOM `Node` into an XmlObject.

Exceptions

`XmlException`

parse(Node, XmlOptions) Method

```
public static XmlObject parse(Node node,
                              XmlOptions options)
     throws XmlException
```

Converts the given DOM `Node` into an XmlObject.

Exceptions

`XmlException`

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlObject parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Loads the given `XMLInputStream` into an XmlObject.

Exceptions

`XmlException`
`XMLStreamException`

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlObject parse(XMLInputStream xis,
                              XmlOptions options)
     throws XmlException, XMLStreamException
```

Loads the given `XMLInputStream` into an XmlObject.

Exceptions

XmlException
XMLStreamException

com.bea.xml

XmlOptions Class

public class ***XmlOptions***

extends `Object`
implements `Serializable`

Used to supply options for loading, saving, and compiling, and validating.

There are two styles for using XmlOptions: multiline setup, and single−line use. Here are two examples. First, multiline style:

```
XmlOptions opts = new XmlOptions();
opts.setSavePrettyPrint();
opts.setSavePrettyPrintIndent(4);
System.out.println(xobj.xmlText(opts));
```

The alternative is single−line usage:

```
System.out.println(xobj.xmlText(
    new XmlOptions().setSavePrettyPrint().setSavePrettyPrintIndent(4)));
```

Hierarchy

```
Object
 XmlOptions
```

All Implemented Interfaces

```
Serializable
```

Field Summary

Constructor Summary

XmlOptions()

Construct a new blank XmlOptions.

XmlOptions*(`XmlOptions` other)*

Construct a new XmlOptions, copying the options.

Method Summary

public Object **get**(Object *option*)
Used to get a generic option

public boolean **hasOption**(Object *option*)
Used to test a generic option

public static XmlOptions **maskNull**(XmlOptions *o*)
If passed null, returns an empty options object.

public void **put**(Object *option*)
Used to set a generic option

public void **put**(Object *option*, Object *value*)
Used to set a generic option

public void **put**(Object *option*, int *value*)
Used to set a generic option

public static Object **safeGet**(XmlOptions *o*, Object *option*)
Used to test a generic option on an options object that may be null

public XmlOptions **setCharacterEncoding**(String *encoding*)
When parsing or writing a document, this sets the character encoding to use.

public XmlOptions **setCompileDownloadUrls**()
If this option is set, then the schema compiler will try to download schemas that appear in imports and includes from network based URLs.

public XmlOptions **setCompileMdefNamespaces**(Set *mdefNamespaces*)
If this option is set, then the schema compiler will permit and ignore multiple definitions of the same component (element, attribute, type, etc) names in the given namespaces.

public XmlOptions **setCompileNoPvrRule**()
If this option is set, the particle valid (restriciton) rule is not enforced when building a SchemaTypeSystem.

public XmlOptions **setCompileNoUpaRule**()
If this option is set, the unique particle attribution rule is not enforced when building a SchemaTypeSystem.

public XmlOptions **setCompileNoValidation**()
If this option is set, validation is not done on the Schema XmlBeans when building a `SchemaTypeSystem`

public XmlOptions **setCompileSubstituteNames**(`Map` *nameMap*)
This option allows for `QName` substitution during schema compilation.

public XmlOptions **setDocumentSourceName**(`String` *documentSourceName*)
This option sets the document source name into the xml store when parsing a document.

public XmlOptions **setDocumentType**(`SchemaType` *type*)
When parsing a document, this sets the type of the root element.

public XmlOptions **setEntityResolver**(`EntityResolver` *resolver*)
If this option is set when compiling a schema, then the given EntityResolver will be consulted in order to resolve any URIs while downloading imported schemas.

public XmlOptions **setErrorListener**(`Collection` *c*)
Sets a collection object for collecting `XmlError` objects during parsing, validation, and compilation.

public XmlOptions **setLoadAdditionalNamespaces**(`Map` *nses*)
Set additional namespace mappings to be added when parsing a document.

public XmlOptions **setLoadLineNumbers**()
If this option is set, line number annotations are placed in the store when parsing a document.

public XmlOptions **setLoadMessageDigest**()
If this option is set when loading from an InputStream or File, then the loader will compute a 160−bit SHA−1 message digest of the XML file while loading it and make it available via XmlObject.documentProperties().getMessageDigest();
The schema compiler uses message digests to detect and eliminate duplicate imported xsd files.

public XmlOptions **setLoadReplaceDocumentElement**(`QName` *replacement*)
If this option is set, the document element is replaced with the given QName when parsing.

public XmlOptions **setLoadStripComments**()
If this option is set, all comments are stripped when parsing a document.

public XmlOptions **setLoadStripProcinsts**()
If this option is set, all processing instructions are stripped when parsing a document.

public XmlOptions **setLoadStripWhitespace**()
If this option is set, all insignificant whitespace is stripped when parsing a document.

public XmlOptions **setLoadSubstituteNamespaces**(Map *substNamespaces*)
This option sets a map of namespace uri substitutions that happen when parsing a document.

public XmlOptions **setLoadTrimTextBuffer**()
If this option is set, the underlying xml text buffer is trimmed immediately after parsing a document resulting in a smaller memory footprint.

public XmlOptions **setSaveAggresiveNamespaces**()
Causes the saver to reduce the number of namespace prefix declarations.

public XmlOptions **setSaveFilterProcinst**(String *filterProcinst*)
This option causes the saver to filter a Processing Instruction with the given target

public XmlOptions **setSaveImplicitNamespaces**(Map *implicitNamespaces*)
If namespaces have already been declared outside the scope of the fragment being saved, this allows those mappings to be passed down to the saver, so the prefixes are not re−declared.

public XmlOptions **setSaveInner**()
This option controls whether saving begins on the element or its contents

public XmlOptions **setSaveNamespacesFirst**()
This option will cause the saver to save namespace attributes first.

public XmlOptions **setSaveOuter**()
This option controls whether saving begins on the element or its contents

public XmlOptions **setSavePrettyPrint**()
This option will cause the saver to reformat white space for easier reading.

public XmlOptions **setSavePrettyPrintIndent**(int *indent*)
When used with `setSavePrettyPrint` this sets the indent amount to use.

public XmlOptions **setSavePrettyPrintOffset**(int *offset*)
When used with `setSavePrettyPrint` this sets the offset amount to use.

public XmlOptions **setSaveSuggestedPrefixes**(Map *suggestedPrefixes*)
A map of hints to pass to the saver for which prefixes to use for which namespace URI.

public XmlOptions **setSaveSyntheticDocumentElement**(QName *name*)
This option causes the saver to wrap the current fragment in an element with the given name.

public XmlOptions **setSaveUseOpenFrag**()
When saving a fragment, this option changes the qname of the synthesized root element.

public XmlOptions **setUseDefaultNamespace()**
If this option is set, the saver will try to use the default namespace for the most commonly used URI.

public XmlOptions **setValidateOnSet()**
If this option is set when an instance is created, then value facets will be checked on each call to a setter or getter method on instances of XmlObject within the instance document.

public XmlOptions **setXqueryCurrentNodeVar**(String *varName*)
Sets the name of the variable that represents the current node in a query expression.

public XmlOptions **setXqueryVariables**(Map *varMap*)
Map the names and values of external variables in an xquery expression.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Field Detail

Constructor Detail

XmlOptions

public XmlOptions()

Construct a new blank XmlOptions.

XmlOptions

public XmlOptions(XmlOptions *other***)**

Construct a new XmlOptions, copying the options.

Method Detail

get(Object) Method

public Object get(Object *option***)**

Used to get a generic option

---

hasOption(Object) Method

```
public boolean hasOption(Object option)
```

Used to test a generic option

---

maskNull(XmlOptions) Method

```
public static XmlOptions maskNull(XmlOptions o)
```

If passed null, returns an empty options object. Otherwise, returns its argument.

---

put(Object) Method

```
public void put(Object option)
```

Used to set a generic option

---

put(Object, Object) Method

```
public void put(Object option,
                Object value)
```

Used to set a generic option

---

put(Object, int) Method

```
public void put(Object option,
                int value)
```

Used to set a generic option

---

safeGet(XmlOptions, Object) Method

```
public static Object safeGet(XmlOptions o,
                             Object option)
```

Used to test a generic option on an options object that may be null

---

setCharacterEncoding(String) Method

```
public XmlOptions setCharacterEncoding(String encoding)
```

When parsing or writing a document, this sets the character encoding to use.

Parameters

encoding
the character encoding

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`
`XmlTokenSource.save(File, XmlOptions)`

---

setCompileDownloadUrls() Method

`public XmlOptions setCompileDownloadUrls()`

If this option is set, then the schema compiler will try to download schemas that appear in imports and includes from network based URLs.

Related Topics

`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`

---

setCompileMdefNamespaces(Set) Method

`public XmlOptions setCompileMdefNamespaces(Set` *`mdefNamespaces`*`)`

If this option is set, then the schema compiler will permit and ignore multiple definitions of the same component (element, attribute, type, etc) names in the given namespaces. If multiple definitions with the same name appear, the definitions that happen to be processed last will be ignored.

Parameters

mdefNamespaces
a set of namespace URIs as Strings

Related Topics

`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`

---

setCompileNoPvrRule() Method

`public XmlOptions setCompileNoPvrRule()`

If this option is set, the particle valid (restriciton) rule is not enforced when building a `SchemaTypeSystem`. See Section 3.9.6 of the XML Schema specification for information on the PVR rule.

Related Topics

`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`

---

setCompileNoUpaRule() Method

`public XmlOptions setCompileNoUpaRule()`

If this option is set, the unique particle attribution rule is not enforced when building a `SchemaTypeSystem`. See Appendix H of the XML Schema specification for information on the UPA rule.

Related Topics

`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`

---

setCompileNoValidation() Method

`public XmlOptions setCompileNoValidation()`

If this option is set, validation is not done on the Schema XmlBeans when building a `SchemaTypeSystem`

Related Topics

`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`

---

setCompileSubstituteNames(Map) Method

`public XmlOptions setCompileSubstituteNames(Map` *`nameMap`*`)`

This option allows for `QName` substitution during schema compilation.

Parameters

nameMap
: a map from `QName`s to substitute `QName`s.

Related Topics

`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`

---

setDocumentSourceName(String) Method

`public XmlOptions setDocumentSourceName(String` *`documentSourceName`*`)`

This option sets the document source name into the xml store when parsing a document. If a document is parsed from a File or URI, it is automatically set to the URI of the source; otherwise, for example, when parsing a String, you can use this option to specify the source name yourself.

Related Topics

`XmlObject.Factory.parse(String, XmlOptions)`

---

setDocumentType(SchemaType) Method

`public XmlOptions setDocumentType(SchemaType type)`

When parsing a document, this sets the type of the root element. If this is set, the parser will not try to guess the type based on the document's `QName`.

Parameters

type
The root element's document type.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setEntityResolver(EntityResolver) Method

`public XmlOptions setEntityResolver(EntityResolver resolver)`

If this option is set when compiling a schema, then the given EntityResolver will be consulted in order to resolve any URIs while downloading imported schemas. EntityResolvers are currently only used by compileXsd; they are not consulted by other functions, for example, parse. This will likely change in the future.

Related Topics

`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`

---

setErrorListener(Collection) Method

`public XmlOptions setErrorListener(Collection c)`

Sets a collection object for collecting `XmlError` objects during parsing, validation, and compilation. When set, the collection will contain all the errors after the operation takes place. Notice that the errors will only have line numbers if the document was loaded with line numbers enabled.

The following simple example illustrates using an error listener during validation.

```
// Create an XmlOptions instance and set the error listener.
XmlOptions validateOptions = new XmlOptions();
ArrayList errorList = new ArrayList();
validateOptions.setErrorListener(errorList);

// Validate the XML.
```

```
boolean isValid = newEmp.validate(validateOptions);

// If the XML isn't valid, loop through the listener's contents,
// printing contained messages.
if (!isValid)
{
     for (int i = 0; i <errorList.size(); i++)
     {
         XmlError error = (XmlError)errorList.get(i);

         System.out.println("\n");
         System.out.println("Message: " + error.getMessage() + "\n");
         System.out.println("Location of invalid XML: " +
             error.getCursorLocation().xmlText() + "\n");
     }
}
```

Parameters

c
: A collection that will be filled with `XmlError` objects via `Collection.add(Object)`

Related Topics

`XmlError`
`XmlObject.Factory.parse(File, XmlOptions)`
`XmlObject.validate(XmlOptions)`
`XmlBeans.compileXsd(XmlObject[], SchemaTypeLoader, XmlOptions)`
`XmlOptions.setLoadLineNumbers()`

---

setLoadAdditionalNamespaces(Map) Method

`public XmlOptions setLoadAdditionalNamespaces(Map nses)`

Set additional namespace mappings to be added when parsing a document.

Parameters

nses
: additional namespace mappings

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setLoadLineNumbers() Method

`public XmlOptions setLoadLineNumbers()`

If this option is set, line number annotations are placed in the store when parsing a document. This is

particularly useful when you want `XmlError` objects to contain line numbers.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`
`XmlError`

---

setLoadMessageDigest() Method

`public XmlOptions setLoadMessageDigest()`

If this option is set when loading from an InputStream or File, then the loader will compute a 160−bit SHA−1 message digest of the XML file while loading it and make it available via XmlObject.documentProperties().getMessageDigest();
The schema compiler uses message digests to detect and eliminate duplicate imported xsd files.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setLoadReplaceDocumentElement(QName) Method

`public XmlOptions setLoadReplaceDocumentElement(QName` *`replacement`*`)`

If this option is set, the document element is replaced with the given QName when parsing. If null is supplied, the document element is removed.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setLoadStripComments() Method

`public XmlOptions setLoadStripComments()`

If this option is set, all comments are stripped when parsing a document.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setLoadStripProcinsts() Method

`public XmlOptions setLoadStripProcinsts()`

If this option is set, all processing instructions are stripped when parsing a document.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setLoadStripWhitespace() Method

`public XmlOptions setLoadStripWhitespace()`

If this option is set, all insignificant whitespace is stripped when parsing a document. Can be used to save memory on large documents when you know there is no mixed content.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setLoadSubstituteNamespaces(Map) Method

`public XmlOptions setLoadSubstituteNamespaces(Map` *`substNamespaces`***`)`**

This option sets a map of namespace uri substitutions that happen when parsing a document.

This is particularly useful if you have documents that use no namespace, but you wish to avoid the name collision problems that occur when you introduce schema definitions without a target namespace.

By mapping the empty string "" (the absence of a URI) to a specific namespace, you can force the parser to behave as if a no−namespace document were actually in the specified namespace. This allows you to type the instance according to a schema in a nonempty namespace, and therefore avoid the problematic practice of using schema definitions without a target namespace.

Parameters

substNamespaces
: a map of document URIs to replacement URIs

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setLoadTrimTextBuffer() Method

`public XmlOptions setLoadTrimTextBuffer()`

If this option is set, the underlying xml text buffer is trimmed immediately after parsing a document resulting in a smaller memory footprint. Use this option if you are loading a large number of unchanging documents that will stay in memory for some time.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setSaveAggresiveNamespaces() Method

`public XmlOptions setSaveAggresiveNamespaces()`

Causes the saver to reduce the number of namespace prefix declarations. The saver will do this by passing over the document twice, first to collect the set of needed namespace declarations, and then second to actually save the document with the declarations collected at the root.

Related Topics

`XmlTokenSource.save(File, XmlOptions)`
`XmlTokenSource.xmlText(XmlOptions)`

---

setSaveFilterProcinst(String) Method

`public XmlOptions setSaveFilterProcinst(String` *`filterProcinst`***`)`**

This option causes the saver to filter a Processing Instruction with the given target

Parameters

filterProcinst
the name of a Processing Instruction to filter on save

Related Topics

`XmlTokenSource.save(File, XmlOptions)`
`XmlTokenSource.xmlText(XmlOptions)`

---

setSaveImplicitNamespaces(Map) Method

`public XmlOptions setSaveImplicitNamespaces(Map` *`implicitNamespaces`***`)`**

If namespaces have already been declared outside the scope of the fragment being saved, this allows those mappings to be passed down to the saver, so the prefixes are not re−declared.

Parameters

implicitNamespaces
a map of prefixes to uris that can be used by the saver without being declared

Related Topics

`XmlTokenSource.save(File, XmlOptions)`
`XmlTokenSource.xmlText(XmlOptions)`

setSaveInner() Method

public XmlOptions setSaveInner()

This option controls whether saving begins on the element or its contents

Related Topics

```
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

setSaveNamespacesFirst() Method

public XmlOptions setSaveNamespacesFirst()

This option will cause the saver to save namespace attributes first.

Related Topics

```
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

setSaveOuter() Method

public XmlOptions setSaveOuter()

This option controls whether saving begins on the element or its contents

Related Topics

```
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

setSavePrettyPrint() Method

public XmlOptions setSavePrettyPrint()

This option will cause the saver to reformat white space for easier reading.

Related Topics

```
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

setSavePrettyPrintIndent(int) Method

public XmlOptions setSavePrettyPrintIndent(int *indent*)

When used with `setSavePrettyPrint` this sets the indent amount to use.

Parameters

indent
the indent amount to use

Related Topics

```
XmlOptions.setSavePrettyPrint()
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

---

setSavePrettyPrintOffset(int) Method

public XmlOptions setSavePrettyPrintOffset(int *offset*)

When used with `setSavePrettyPrint` this sets the offset amount to use.

Parameters

offset
the offset amount to use

Related Topics

```
XmlOptions.setSavePrettyPrint()
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

---

setSaveSuggestedPrefixes(Map) Method

public XmlOptions setSaveSuggestedPrefixes(Map *suggestedPrefixes*)

A map of hints to pass to the saver for which prefixes to use for which namespace URI.

Parameters

suggestedPrefixes
a map from URIs to prefixes

Related Topics

```
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

setSaveSyntheticDocumentElement(QName) Method

public XmlOptions setSaveSyntheticDocumentElement(QName *name*)

This option causes the saver to wrap the current fragment in an element with the given name.

Parameters

name
the name to use for the top level element

Related Topics

```
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

setSaveUseOpenFrag() Method

public XmlOptions setSaveUseOpenFrag()

When saving a fragment, this option changes the qname of the synthesized root element. Normally <xml−fragment> is used.

Related Topics

```
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

setUseDefaultNamespace() Method

public XmlOptions setUseDefaultNamespace()

If this option is set, the saver will try to use the default namespace for the most commonly used URI. If it is not set the saver will always created named prefixes.

Related Topics

```
XmlTokenSource.save(File, XmlOptions)
XmlTokenSource.xmlText(XmlOptions)
```

setValidateOnSet() Method

public XmlOptions setValidateOnSet()

If this option is set when an instance is created, then value facets will be checked on each call to a setter or

getter method on instances of XmlObject within the instance document. If the facets are not satisfied, then an unchecked exception is thrown immediately. This option is useful for finding code that is introducing invalid values in an XML document, but it slows performance.

Related Topics

`XmlObject.Factory.parse(File, XmlOptions)`

---

setXqueryCurrentNodeVar(String) Method

`public XmlOptions setXqueryCurrentNodeVar(String` *`varName`***`)`**

Sets the name of the variable that represents the current node in a query expression.

Parameters

varName
The new variable name to use for the query.

Related Topics

`XmlObject.execQuery(String)`
`XmlCursor.execQuery(String)`

---

setXqueryVariables(Map) Method

`public XmlOptions setXqueryVariables(Map` *`varMap`***`)`**

Map the names and values of external variables in an xquery expression. The keys of the map are the variable names in the query without the '$' prefix. The values of the map are objects and can be any of the primitive wrapper classes, String, XmlObject, or XmlCursor. The mapping only applies to xquery and has no effect on xpath expressions.

Parameters

varMap
a map from `String`s to variable instances.

Related Topics

`XmlObject.execQuery(String)`
`XmlCursor.execQuery(String)`

com.bea.xml

XmlPositiveInteger Interface

public interface ***XmlPositiveInteger***

extends `XmlNonNegativeInteger`

Corresponds to the XML Schema xs:positiveInteger type. One of the derived types based on xs:decimal.

Verified to be positive when validating.

Convertible to `BigInteger`.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlNonNegativeInteger`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlPositiveInteger.Factory`** A class with methods for creating instances of `XmlPositiveInteger`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlNonNegativeInteger`

`XmlNonNegativeInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

public static final SchemaType ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlNonNegativeInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlPositiveInteger.Factory Class

public static final class ***XmlPositiveInteger.Factory***

extends `Object`

A class with methods for creating instances of `XmlPositiveInteger`.

Hierarchy

```
Object
 XmlPositiveInteger.Factory
```

Enclosing interface

`XmlPositiveInteger`

Method Summary

`public static XmlPositiveInteger` ***newInstance***()
Creates an empty instance of `XmlPositiveInteger`

`public static XmlPositiveInteger` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlPositiveInteger`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlPositiveInteger` ***newValue***(`Object` *obj*)
Creates an immutable `XmlPositiveInteger` value

`public static XmlPositiveInteger` ***parse***(`String` *s*)
Parses a `XmlPositiveInteger` fragment from a String.

`public static XmlPositiveInteger` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlPositiveInteger` fragment from a String.

`public static XmlPositiveInteger` ***parse***(`File` *f*)
Parses a `XmlPositiveInteger` fragment from a File.

public static XmlPositiveInteger **parse**(File f, XmlOptions options)
Parses a XmlPositiveInteger fragment from a File.

public static XmlPositiveInteger **parse**(URL u)
Parses a XmlPositiveInteger fragment from a URL.

public static XmlPositiveInteger **parse**(URL u, XmlOptions options)
Parses a XmlPositiveInteger fragment from a URL.

public static XmlPositiveInteger **parse**(InputStream is)
Parses a XmlPositiveInteger fragment from an InputStream.

public static XmlPositiveInteger **parse**(InputStream is, XmlOptions options)
Parses a XmlPositiveInteger fragment from an InputStream.

public static XmlPositiveInteger **parse**(Reader r)
Parses a XmlPositiveInteger fragment from a Reader.

public static XmlPositiveInteger **parse**(Reader r, XmlOptions options)
Parses a XmlPositiveInteger fragment from a Reader.

public static XmlPositiveInteger **parse**(Node node)
Parses a XmlPositiveInteger fragment from a DOM Node.

public static XmlPositiveInteger **parse**(Node node, XmlOptions options)
Parses a XmlPositiveInteger fragment from a DOM Node.

public static XmlPositiveInteger **parse**(XMLInputStream xis)
Parses a XmlPositiveInteger fragment from an XMLInputStream.

public static XmlPositiveInteger **parse**(XMLInputStream xis, XmlOptions options)
Parses a XmlPositiveInteger fragment from an XMLInputStream.

Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

newInstance() Method

public static XmlPositiveInteger newInstance()

Creates an empty instance of XmlPositiveInteger

---

newInstance(XmlOptions) Method

```
public static XmlPositiveInteger newInstance(XmlOptions options)
```

Creates an empty instance of `XmlPositiveInteger`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlPositiveInteger newValue(Object obj)
```

Creates an immutable `XmlPositiveInteger` value

---

parse(String) Method

```
public static XmlPositiveInteger parse(String s)
     throws XmlException
```

Parses a `XmlPositiveInteger` fragment from a String. For example: "`<xml-fragment>1234567890</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlPositiveInteger parse(String s,
                                       XmlOptions options)
     throws XmlException
```

Parses a `XmlPositiveInteger` fragment from a String. For example: "`<xml-fragment>1234567890</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlPositiveInteger parse(File f)
     throws XmlException, IOException
```

Parses a `XmlPositiveInteger` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlPositiveInteger parse(File f,
                                       XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlPositiveInteger` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlPositiveInteger parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlPositiveInteger` fragment from a URL.

Exceptions

`XmlException`
`IOException`

---

parse(URL, XmlOptions) Method

```
public static XmlPositiveInteger parse(URL u,
                                       XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlPositiveInteger` fragment from a URL.

Exceptions

`XmlException`
`IOException`

---

parse(InputStream) Method

```
public static XmlPositiveInteger parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlPositiveInteger` fragment from an InputStream.

Exceptions

`XmlException`
`IOException`

---

parse(InputStream, XmlOptions) Method

```
public static XmlPositiveInteger parse(InputStream is,
                                       XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlPositiveInteger` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlPositiveInteger parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlPositiveInteger` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlPositiveInteger parse(Reader r,
                                       XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlPositiveInteger` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlPositiveInteger parse(Node node)
     throws XmlException
```

Parses a `XmlPositiveInteger` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlPositiveInteger parse(Node node,
                                       XmlOptions options)
     throws XmlException
```

Parses a `XmlPositiveInteger` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlPositiveInteger parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlPositiveInteger` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlPositiveInteger parse(XMLInputStream xis,
                                       XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlPositiveInteger` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlQName Interface

public interface ***XmlQName***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:QName type.

A QName is the logical combination of an XML namespace URI and a localName. Although in an XML instance document, a QName appears as "prefix:localName", the logical value of a QName does NOT contain any information about the prefix, only the namespace URI to which the prefix maps. For example, two QNames "a:hello" and "b:hello" are perfectly equivalent if "a:" in the first instance maps to the same URI as "b:" in the second instance.

Convertible to `QName`.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlQName.Factory`** A class with methods for creating instances of `XmlQName`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public QName` ***`getQNameValue`***`()`
Returns this value as a `QName`

`public QName` ***`qNameValue`***`()`
Returns this value as a `QName`

`public void` ***`set`***`(QName name)`
Sets this value as a `QName`

`public void` ***`setQNameValue`***`(QName name)`
Sets this value as a `QName`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getQNameValue() Method

`public QName getQNameValue()`

Returns this value as a `QName`

---

qNameValue() Method

DEPRECATED replaced with `com.bea.xml.XmlQName.getQNameValue()`

`public QName qNameValue()`

Returns this value as a `QName`

---

set(QName) Method

DEPRECATED replaced with `com.bea.xml.XmlQName.setQNameValue(javax.xml.namespace.QName)`

`public void set(QName` *`name`***`)`**

Sets this value as a `QName`

---

setQNameValue(QName) Method

`public void setQNameValue(QName` *`name`***`)`**

Sets this value as a `QName`

XmlQName.Factory Class

public static final class ***XmlQName.Factory***

extends `Object`

A class with methods for creating instances of `XmlQName`.

Hierarchy

```
Object
 XmlQName.Factory
```

Enclosing interface

`XmlQName`

Method Summary

`public static XmlQName` ***newInstance***()
Creates an empty instance of `XmlQName`

`public static XmlQName` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlQName`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlQName` ***newValue***(`Object` *obj*)
Creates an immutable `XmlQName` value

`public static XmlQName` ***parse***(`String` *s*)
Parses a `XmlQName` fragment from a String.

`public static XmlQName` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlQName` fragment from a String.

`public static XmlQName` ***parse***(`File` *f*)
Parses a `XmlQName` fragment from a File.

`public static XmlQName` ***parse***(*File f, XmlOptions options*)
Parses a `XmlQName` fragment from a File.

`public static XmlQName` ***parse***(*URL u*)
Parses a `XmlQName` fragment from a URL.

`public static XmlQName` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlQName` fragment from a URL.

`public static XmlQName` ***parse***(*InputStream is*)
Parses a `XmlQName` fragment from an InputStream.

`public static XmlQName` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlQName` fragment from an InputStream.

`public static XmlQName` ***parse***(*Reader r*)
Parses a `XmlQName` fragment from a Reader.

`public static XmlQName` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlQName` fragment from a Reader.

`public static XmlQName` ***parse***(*Node node*)
Parses a `XmlQName` fragment from a DOM Node.

`public static XmlQName` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlQName` fragment from a DOM Node.

`public static XmlQName` ***parse***(*XMLInputStream xis*)
Parses a `XmlQName` fragment from an XMLInputStream.

`public static XmlQName` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlQName` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlQName newInstance()`

Creates an empty instance of `XmlQName`

---

newInstance(XmlOptions) Method

```
public static XmlQName newInstance(XmlOptions options)
```

Creates an empty instance of `XmlQName`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlQName newValue(Object obj)
```

Creates an immutable `XmlQName` value

---

parse(String) Method

```
public static XmlQName parse(String s)
     throws XmlException
```

Parses a `XmlQName` fragment from a String. For example: "`<xml-fragment xmlns:x="http://openuri.org/">x:sample</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlQName parse(String s,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlQName` fragment from a String. For example: "`<xml-fragment xmlns:x="http://openuri.org/">x:sample</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlQName parse(File f)
     throws XmlException, IOException
```

Parses a `XmlQName` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlQName parse(File f,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlQName` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlQName parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlQName` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlQName parse(URL u,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlQName` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlQName parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlQName` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlQName parse(InputStream is,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlQName` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlQName parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlQName` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlQName parse(Reader r,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlQName` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlQName parse(Node node)
     throws XmlException
```

Parses a `XmlQName` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlQName parse(Node node,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlQName` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlQName parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlQName` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlQName parse(XMLInputStream xis,
                             XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlQName` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlRuntimeException Class

public class ***XmlRuntimeException***

extends `RuntimeException`

An unchecked XML exception. May contain any number of `XmlError` objects.

Related Topics

`XmlError`
`XmlException`

Hierarchy

```
Object
 Throwable
  Exception
   RuntimeException
    XmlRuntimeException
```

All Implemented Interfaces

`Serializable`

Direct Known Subclasses

`SchemaTypeLoaderException`

Constructor Summary

XmlRuntimeException*(`XmlError` error)*

Constructs an XmlRuntimeException from an XmlError.

XmlRuntimeException*(`XmlException` xmlException)*

Constructs an XmlRuntimeException from an `XmlException`.

XmlRuntimeException*(`String` m, `Throwable` t, `XmlError` error)*

Constructs an XmlRuntimeException from a message, a cause, and an XmlError.

XmlRuntimeException(`String` *m*, `Throwable` *t*, `Collection` *errors)*

Constructs an XmlRuntimeException from a message, a cause, and a collection of XmlErrors.

XmlRuntimeException(`String` *m*, `Throwable` *t)*

Constructs an XmlRuntimeException from a message and a cause.

XmlRuntimeException(`String` *m)*

Constructs an XmlRuntimeException from a message.

XmlRuntimeException(`Throwable` *t)*

Constructs an XmlRuntimeException from a cause.

Method Summary

`public XmlError` ***getError***()
Returns the first `XmlError` that caused this exception, if any.

`public Collection` ***getErrors***()
Returns the collection of `XmlError` that caused this exception, if any.

Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

XmlRuntimeException

`public XmlRuntimeException(XmlError` *`error`*`)`

Constructs an XmlRuntimeException from an XmlError.

XmlRuntimeException

```
public XmlRuntimeException(XmlException xmlException)
```

Constructs an XmlRuntimeException from an `XmlException`.

XmlRuntimeException

```
public XmlRuntimeException(String m,
                           Throwable t,
                           XmlError error)
```

Constructs an XmlRuntimeException from a message, a cause, and an XmlError.

XmlRuntimeException

```
public XmlRuntimeException(String m,
                           Throwable t,
                           Collection errors)
```

Constructs an XmlRuntimeException from a message, a cause, and a collection of XmlErrors.

XmlRuntimeException

```
public XmlRuntimeException(String m,
                           Throwable t)
```

Constructs an XmlRuntimeException from a message and a cause.

XmlRuntimeException

```
public XmlRuntimeException(String m)
```

Constructs an XmlRuntimeException from a message.

XmlRuntimeException

```
public XmlRuntimeException(Throwable t)
```

Constructs an XmlRuntimeException from a cause.

Method Detail

getError() Method

`public XmlError getError()`

Returns the first `XmlError` that caused this exception, if any.

---

getErrors() Method

`public Collection getErrors()`

Returns the collection of `XmlError` that caused this exception, if any.

XmlSaxHandler Interface

public interface ***XmlSaxHandler***

A holder for a SAX `ContentHandler` and `LexicalHandler` that are capable of loading an `XmlObject` instance. Once all the SAX events are pushed to the handlers, call `XmlSaxHandler.getObject()` to get the loaded XmlObject.

Related Topics

```
XmlObject.Factory.newXmlSaxHandler()
SchemaTypeLoader.newXmlSaxHandler(SchemaType, XmlOptions)
```

Method Summary

public ContentHandler ***getContentHandler()***	The ContentHandler expecting SAX content events.
public LexicalHandler ***getLexicalHandler()***	The LexicalHandler expecting SAX lexical events.
public XmlObject ***getObject()***	Returns the loaded XmlObject after all the SAX events have been finished

Method Detail

getContentHandler() Method

`public ContentHandler getContentHandler()`

The ContentHandler expecting SAX content events.

Related Topics

`ContentHandler`

---

getLexicalHandler() Method

`public LexicalHandler getLexicalHandler()`

The LexicalHandler expecting SAX lexical events.

Related Topics

getObject() Method

```
public XmlObject getObject()
     throws XmlException
```

Returns the loaded XmlObject after all the SAX events have been finished

Exceptions

XmlException

com.bea.xml

XmlShort Interface

public interface ***XmlShort***

extends `XmlInt`

Corresponds to the XML Schema xs:short type. One of the derived types based on xs:decimal.

Naturally, convertible to a Java short.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInt`, `XmlInteger`, `XmlLong`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlByte`

Nested Class Summary

`public static final class`	**`XmlShort.Factory`** A class with methods for creating instances of `XmlShort`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInt`

`XmlInt.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlLong`

`XmlLong.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType` ***type***
The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInt`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlLong`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public short` ***getShortValue***()
Returns this value as a short

`public void` ***set***(*short s*)
Sets this value as a short

`public void` ***setShortValue***(*short s*)
Sets this value as a short

`public short` ***shortValue***()

Returns this value as a short

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInt`

`getIntValue, intValue, set, setIntValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlLong`

`getLongValue, longValue, set, setLongValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getShortValue() Method

`public short getShortValue()`

Returns this value as a short

---

set(short) Method

DEPRECATED replaced with `com.bea.xml.XmlShort.setShortValue(short)`

`public void set(short s)`

Sets this value as a short

---

setShortValue(short) Method

`public void setShortValue(short s)`

Sets this value as a short

---

shortValue() Method

DEPRECATED replaced with `com.bea.xml.XmlShort.getShortValue()`

`public short shortValue()`

Returns this value as a short

com.bea.xml

XmlShort.Factory Class

public static final class ***XmlShort.Factory***

extends `Object`

A class with methods for creating instances of `XmlShort`.

Hierarchy

```
Object
 XmlShort.Factory
```

Enclosing interface

`XmlShort`

Method Summary

`public static XmlShort` ***newInstance***()
Creates an empty instance of `XmlShort`

`public static XmlShort` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlShort`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlShort` ***newValue***(`Object` *obj*)
Creates an immutable `XmlShort` value

`public static XmlShort` ***parse***(`String` *s*)
Parses a `XmlShort` fragment from a String.

`public static XmlShort` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlShort` fragment from a String.

`public static XmlShort` ***parse***(`File` *f*)
Parses a `XmlShort` fragment from a File.

`public static XmlShort` ***parse***(*File f, XmlOptions options*)
Parses a `XmlShort` fragment from a File.

`public static XmlShort` ***parse***(*URL u*)
Parses a `XmlShort` fragment from a URL.

`public static XmlShort` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlShort` fragment from a URL.

`public static XmlShort` ***parse***(*InputStream is*)
Parses a `XmlShort` fragment from an InputStream.

`public static XmlShort` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlShort` fragment from an InputStream.

`public static XmlShort` ***parse***(*Reader r*)
Parses a `XmlShort` fragment from a Reader.

`public static XmlShort` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlShort` fragment from a Reader.

`public static XmlShort` ***parse***(*Node node*)
Parses a `XmlShort` fragment from a DOM Node.

`public static XmlShort` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlShort` fragment from a DOM Node.

`public static XmlShort` ***parse***(*XMLInputStream xis*)
Parses a `XmlShort` fragment from an XMLInputStream.

`public static XmlShort` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlShort` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlShort newInstance()`

Creates an empty instance of `XmlShort`

---

newInstance(XmlOptions) Method

```
public static XmlShort newInstance(XmlOptions options)
```

Creates an empty instance of `XmlShort`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlShort newValue(Object obj)
```

Creates an immutable `XmlShort` value

---

parse(String) Method

```
public static XmlShort parse(String s)
     throws XmlException
```

Parses a `XmlShort` fragment from a String. For example: "`<xml-fragment>12345</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlShort parse(String s,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlShort` fragment from a String. For example: "`<xml-fragment>12345</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlShort parse(File f)
     throws XmlException, IOException
```

Parses a `XmlShort` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlShort parse(File f,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlShort` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlShort parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlShort` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlShort parse(URL u,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlShort` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlShort parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlShort` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlShort parse(InputStream is,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlShort` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlShort parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlShort` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlShort parse(Reader r,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlShort` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlShort parse(Node node)
     throws XmlException
```

Parses a `XmlShort` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlShort parse(Node node,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlShort` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlShort parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlShort` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlShort parse(XMLInputStream xis,
                             XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlShort` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlSimpleList Class

public class ***XmlSimpleList***

extends `Object`
implements `List`, `Serializable`

The immutable `List` returned for XML simple list values. XmlSimpleList implements an equals() and hashCode() that compare list contents, so two XmlSimpleLists are the same if they have the same values in the same order.

Hierarchy

```
Object
 XmlSimpleList
```

All Implemented Interfaces

`Collection`, `List`, `Serializable`

Constructor Summary

XmlSimpleList*(`List` list)*

Constructs an immutable XmlSimpleList that wraps (does not copy) the given `List`.

Method Summary

`public boolean` ***add****(`Object` o)*
Unsupported because this list is immutable.

`public void` ***add****(int index, `Object` element)*
Unsupported because this list is immutable.

`public boolean` ***addAll****(`Collection` coll)*
Unsupported because this list is immutable.

`public boolean` ***addAll****(int index, `Collection` c)*
Unsupported because this list is immutable.

`public void`

clear()
Unsupported because this list is immutable.

public boolean **contains**(Object o)
True if the list is contains an object equal to o.

public boolean **containsAll**(Collection coll)
True if the list is contains all the objects in the given collection.

public boolean **equals**(Object o)
Two XmlSimpleLists are equal if all their items are equal.

public Object **get**(int index)
Returns the object at the specified position in this list.

public int **hashCode**()
Combines the hash codes of all the list items.

public int **indexOf**(Object o)
Returns index of the first occurance of an object equal to o.

public boolean **isEmpty**()
True if the list is empty.

public Iterator **iterator**()
Returns an iterator over the elements in this list in proper sequence.

public int **lastIndexOf**(Object o)
Returns index of the last occurance of an object equal to o.

public ListIterator **listIterator**()
Returns a list iterator of the elements in this list in proper sequence.

public ListIterator **listIterator**(int index)
Returns a list iterator of the elements in this list in proper sequence, starting at the specified position in this list.

public boolean **remove**(Object o)
Unsupported because this list is immutable.

public Object **remove**(int index)
Unsupported because this list is immutable.

public boolean **removeAll**(Collection coll)
Unsupported because this list is immutable.

public boolean **retainAll**(Collection coll)

Unsupported because this list is immutable.

`public Object` ***set***(*int index, `Object` element*)

Unsupported because this list is immutable.

`public int` ***size***()

Returns the number of elements in this list.

`public List` ***subList***(*int from, int to*)

Returns a view of the portion of this list between the specified fromIndex, inclusive, and toIndex, exclusive.

`public Object[]` ***toArray***()

Copies the collection to an array.

`public Object[]` ***toArray***(*`Object[]` a*)

Copies the collection to an array of a specified type.

`public String` ***toString***()

Returns a space−separated list of the string representations of all the items in the list.

Methods from class `java.lang.Object`

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Methods from interface `java.util.Collection`

`add, addAll, clear, contains, containsAll, equals, hashCode, isEmpty, iterator, remove, removeAll, retainAll, size, toArray, toArray`

Methods from interface `java.util.List`

`add, add, addAll, addAll, clear, contains, containsAll, equals, get, hashCode, indexOf, isEmpty, iterator, lastIndexOf, listIterator, listIterator, remove, remove, removeAll, retainAll, set, size, subList, toArray, toArray`

Constructor Detail

XmlSimpleList

`public XmlSimpleList(List` *`list`*`)`

Constructs an immutable XmlSimpleList that wraps (does not copy) the given `List`. All non−mutating methods delegate to the underlying List instance.

Method Detail

add(Object) Method

```
public boolean add(Object o)
```

Unsupported because this list is immutable.

---

add(int, Object) Method

```
public void add(int index,
                Object element)
```

Unsupported because this list is immutable.

---

addAll(Collection) Method

```
public boolean addAll(Collection coll)
```

Unsupported because this list is immutable.

---

addAll(int, Collection) Method

```
public boolean addAll(int index,
                      Collection c)
```

Unsupported because this list is immutable.

---

clear() Method

```
public void clear()
```

Unsupported because this list is immutable.

---

contains(Object) Method

```
public boolean contains(Object o)
```

True if the list is contains an object equal to o.

---

containsAll(Collection) Method

```
public boolean containsAll(Collection coll)
```

True if the list is contains all the objects in the given collection.

---

equals(Object) Method

public boolean equals(Object *o*)

Two XmlSimpleLists are equal if all their items are equal. (They must have the same number of items, and the items must be in the same order.)

Overrides

`Object.equals(Object)`

---

get(int) Method

public Object get(int *index*)

Returns the object at the specified position in this list.

---

hashCode() Method

public int hashCode()

Combines the hash codes of all the list items.

Overrides

`Object.hashCode()`

---

indexOf(Object) Method

public int indexOf(Object *o*)

Returns index of the first occurance of an object equal to o.

---

isEmpty() Method

public boolean isEmpty()

True if the list is empty.

---

iterator() Method

public Iterator iterator()

Returns an iterator over the elements in this list in proper sequence.

lastIndexOf(Object) Method

public int lastIndexOf(Object *o*)

Returns index of the last occurance of an object equal to o.

listIterator() Method

public ListIterator listIterator()

Returns a list iterator of the elements in this list in proper sequence.

listIterator(int) Method

public ListIterator listIterator(int *index*)

Returns a list iterator of the elements in this list in proper sequence, starting at the specified position in this list.

remove(Object) Method

public boolean remove(Object *o*)

Unsupported because this list is immutable.

remove(int) Method

public Object remove(int *index*)

Unsupported because this list is immutable.

removeAll(Collection) Method

public boolean removeAll(Collection *coll*)

Unsupported because this list is immutable.

retainAll(Collection) Method

public boolean retainAll(Collection *coll*)

Unsupported because this list is immutable.

set(int, Object) Method

```
public Object set(int index,
                  Object element)
```

Unsupported because this list is immutable.

size() Method

```
public int size()
```

Returns the number of elements in this list.

subList(int, int) Method

```
public List subList(int from,
                    int to)
```

Returns a view of the portion of this list between the specified fromIndex, inclusive, and toIndex, exclusive.

toArray() Method

```
public Object[] toArray()
```

Copies the collection to an array.

toArray(Object[]) Method

```
public Object[] toArray(Object[] a)
```

Copies the collection to an array of a specified type.

toString() Method

```
public String toString()
```

Returns a space−separated list of the string representations of all the items in the list. For most lists, this is a valid xml lexical value for the list. (The notable exception is a list of QNames.)

Overrides

`Object.toString()`

DEPRECATED Superceded by JSR 173

com.bea.xml

XMLStreamValidationException Class

public class ***XMLStreamValidationException***

extends `XMLStreamException`

An exception thrown from a validating XMLInputStream.

Related Topics

`XmlObject.Factory.newValidatingXMLInputStream(XMLInputStream)`
`SchemaTypeLoader.newValidatingXMLInputStream(XMLInputStream, SchemaType, XmlOptions)`

Hierarchy

```
Object
 Throwable
  Exception
   IOException
    XMLStreamException
     XMLStreamValidationException
```

All Implemented Interfaces

`NestedThrowable`, `Serializable`

Field Summary

Fields from weblogic.xml.stream.`XMLStreamException`

`th`

Constructor Summary

XMLStreamValidationException*(`XmlError` xmlError)*

Method Summary

public XmlError ***getXmlError()***

Methods from `weblogic.xml.stream.XMLStreamException`

`getMessage, getNested, getNestedException, printStackTrace, printStackTrace, printStackTrace, superPrintStackTrace, superPrintStackTrace, superToString, toString`

Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods from interface `weblogic.utils.NestedThrowable`

`getNested, superPrintStackTrace, superPrintStackTrace, superToString`

Constructor Detail

XMLStreamValidationException

public XMLStreamValidationException(XmlError *xmlError*)

Method Detail

getXmlError() Method

public XmlError getXmlError()

com.bea.xml

XmlString Interface

public interface ***XmlString***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:string type.

A basic string in XML schema is not whitespace normalized. If you want your string type to be insensitive to variations in runs of whitespace, consider using xs:token (aka `XmlToken`) instead. To forbid whitespace and permit just alphanumeric and other common identifier characters consider xs:NMTOKEN (aka `XmlNMTOKEN`) instead.

Convertible to `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlENTITY`, `XmlID`, `XmlIDREF`, `XmlLanguage`, `XmlName`, `XmlNCName`, `XmlNMTOKEN`, `XmlNormalizedString`, `XmlToken`

Nested Class Summary

`public static final class`	**`XmlString.Factory`** A class with methods for creating instances of `XmlString`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***`type`*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlString.Factory Class

public static final class ***XmlString.Factory***

extends `Object`

A class with methods for creating instances of `XmlString`.

Hierarchy

```
Object
 XmlString.Factory
```

Enclosing interface

`XmlString`

Method Summary

`public static XmlString` ***newInstance***()
Creates an empty instance of `XmlString`

`public static XmlString` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlString`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlString` ***newValue***(`Object` *obj*)
Creates an immutable `XmlString` value

`public static XmlString` ***parse***(`String` *s*)
Parses a `XmlString` fragment from a String.

`public static XmlString` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlString` fragment from a String.

`public static XmlString` ***parse***(`File` *f*)
Parses a `XmlString` fragment from a File.

`public static XmlString` ***parse***(*File f, XmlOptions options*)
Parses a `XmlString` fragment from a File.

`public static XmlString` ***parse***(*URL u*)
Parses a `XmlString` fragment from a URL.

`public static XmlString` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlString` fragment from a URL.

`public static XmlString` ***parse***(*InputStream is*)
Parses a `XmlString` fragment from an InputStream.

`public static XmlString` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlString` fragment from an InputStream.

`public static XmlString` ***parse***(*Reader r*)
Parses a `XmlString` fragment from a Reader.

`public static XmlString` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlString` fragment from a Reader.

`public static XmlString` ***parse***(*Node node*)
Parses a `XmlString` fragment from a DOM Node.

`public static XmlString` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlString` fragment from a DOM Node.

`public static XmlString` ***parse***(*XMLInputStream xis*)
Parses a `XmlString` fragment from an XMLInputStream.

`public static XmlString` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlString` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlString newInstance()`

Creates an empty instance of `XmlString`

---

newInstance(XmlOptions) Method

```
public static XmlString newInstance(XmlOptions options)
```

Creates an empty instance of `XmlString`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlString newValue(Object obj)
```

Creates an immutable `XmlString` value

---

parse(String) Method

```
public static XmlString parse(String s)
     throws XmlException
```

Parses a `XmlString` fragment from a String. For example: "`<xml-fragment> arbitrary string </xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlString parse(String s,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlString` fragment from a String. For example: "`<xml-fragment> arbitrary string </xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlString parse(File f)
     throws XmlException, IOException
```

Parses a `XmlString` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlString parse(File f,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlString` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlString parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlString` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlString parse(URL u,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlString` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlString parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlString` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlString parse(InputStream is,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlString` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlString parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlString` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlString parse(Reader r,
                              XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlString` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlString parse(Node node)
     throws XmlException
```

Parses a `XmlString` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlString parse(Node node,
                              XmlOptions options)
     throws XmlException
```

Parses a `XmlString` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlString parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlString` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlString parse(XMLInputStream xis,
                              XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlString` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlTime Interface

public interface ***XmlTime***

extends `XmlAnySimpleType`

Corresponds to the XML Schema xs:time type. A gDay specifies only a day−of−month.

Convertible to `Calendar` or `GDate`.

Related Topics

`XmlCalendar`
`GDate`

All Superinterfaces

`XmlAnySimpleType`, `XmlObject`, `XmlTokenSource`

Nested Class Summary

`public static final class`	**`XmlTime.Factory`** A class with methods for creating instances of `XmlTime`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType`	***type*** The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

`public Calendar` ***`calendarValue`***`()`
Returns this value as a `Calendar`

`public GDate` ***`gDateValue`***`()`
Returns this value as a `GDate`

`public Calendar` ***`getCalendarValue`***`()`
Returns this value as a `Calendar`

`public GDate` ***`getGDateValue`***`()`
Returns this value as a `GDate`

`public void` ***`set`***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***`set`***`(GDateSpecification gd)`
Sets this value as a `GDateSpecification`

`public void` ***`setCalendarValue`***`(Calendar c)`
Sets this value as a `Calendar`

`public void` ***`setGDateValue`***`(GDate gd)`
Sets this value as a `GDateSpecification`

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save,`

`xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

calendarValue() Method

DEPRECATED replaced with `com.bea.xml.XmlTime.getCalendarValue()`

`public Calendar calendarValue()`

Returns this value as a `Calendar`

---

gDateValue() Method

DEPRECATED replaced with `com.bea.xml.XmlTime.getGDateValue()`

`public GDate gDateValue()`

Returns this value as a `GDate`

---

getCalendarValue() Method

`public Calendar getCalendarValue()`

Returns this value as a `Calendar`

---

getGDateValue() Method

`public GDate getGDateValue()`

Returns this value as a `GDate`

---

set(Calendar) Method

DEPRECATED replaced with
`com.bea.xml.XmlTime.setCalendarValue(java.util.Calendar)`

`public void set(Calendar c)`

Sets this value as a `Calendar`

---

set(GDateSpecification) Method

DEPRECATED replaced with `com.bea.xml.XmlTime.setGDateValue(com.bea.xml.GDate)`

`public void set(GDateSpecification gd)`

Sets this value as a `GDateSpecification`

---

setCalendarValue(Calendar) Method

`public void setCalendarValue(Calendar c)`

Sets this value as a `Calendar`

---

setGDateValue(GDate) Method

`public void setGDateValue(GDate gd)`

Sets this value as a `GDateSpecification`

com.bea.xml

XmlTime.Factory Class

public static final class ***XmlTime.Factory***

extends `Object`

A class with methods for creating instances of `XmlTime`.

Hierarchy

```
Object
 XmlTime.Factory
```

Enclosing interface

`XmlTime`

Method Summary

`public static XmlTime` ***newInstance***()
Creates an empty instance of `XmlTime`

`public static XmlTime` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlTime`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlTime` ***newValue***(`Object` *obj*)
Creates an immutable `XmlTime` value

`public static XmlTime` ***parse***(`String` *s*)
Parses a `XmlTime` fragment from a String.

`public static XmlTime` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlTime` fragment from a String.

`public static XmlTime` ***parse***(`File` *f*)
Parses a `XmlTime` fragment from a File.

`public static XmlTime` ***parse***(*File f, XmlOptions options*)
Parses a `XmlTime` fragment from a File.

`public static XmlTime` ***parse***(*URL u*)
Parses a `XmlTime` fragment from a URL.

`public static XmlTime` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlTime` fragment from a URL.

`public static XmlTime` ***parse***(*InputStream is*)
Parses a `XmlTime` fragment from an InputStream.

`public static XmlTime` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlTime` fragment from an InputStream.

`public static XmlTime` ***parse***(*Reader r*)
Parses a `XmlTime` fragment from a Reader.

`public static XmlTime` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlTime` fragment from a Reader.

`public static XmlTime` ***parse***(*Node node*)
Parses a `XmlTime` fragment from a DOM Node.

`public static XmlTime` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlTime` fragment from a DOM Node.

`public static XmlTime` ***parse***(*XMLInputStream xis*)
Parses a `XmlTime` fragment from an XMLInputStream.

`public static XmlTime` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlTime` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlTime newInstance()`

Creates an empty instance of `XmlTime`

---

newInstance(XmlOptions) Method

```
public static XmlTime newInstance(XmlOptions options)
```

Creates an empty instance of `XmlTime`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlTime newValue(Object obj)
```

Creates an immutable `XmlTime` value

---

parse(String) Method

```
public static XmlTime parse(String s)
     throws XmlException
```

Parses a `XmlTime` fragment from a String. For example: "`<xml-fragment>12:00:00</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlTime parse(String s,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlTime` fragment from a String. For example: "`<xml-fragment>12:00:00</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlTime parse(File f)
     throws XmlException, IOException
```

Parses a `XmlTime` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlTime parse(File f,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlTime` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlTime parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlTime` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlTime parse(URL u,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlTime` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlTime parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlTime` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlTime parse(InputStream is,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlTime` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlTime parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlTime` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlTime parse(Reader r,
                            XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlTime` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlTime parse(Node node)
     throws XmlException
```

Parses a `XmlTime` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlTime parse(Node node,
                            XmlOptions options)
     throws XmlException
```

Parses a `XmlTime` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlTime parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlTime` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlTime parse(XMLInputStream xis,
                            XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlTime` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlToken Interface

public interface ***XmlToken***

extends `XmlNormalizedString`

Corresponds to the XML Schema xs:token type. One of the derived types based on xs:string.

A token is XML's best representation for a "whitespace insensitive string." All carriage returns, linefeeds, and tabs are converted to ordinary space characters (as with xs:normalizedString), and furthermore, all contiguous runs of space are collapsed to single spaces, and leading and trailing spaces are trimmed.

If you want `"  high  priority  "` to be equivalent to `"high priority"`, you should consider using xs:token or a subtype of xs:token.

When the `XmlToken.stringValue()` is obtained from an XmlToken, the normalized, trimmed, whitespace collapsed value is returned.

Convertible to `String`.

All Superinterfaces

`XmlAnySimpleType`, `XmlNormalizedString`, `XmlObject`, `XmlString`, `XmlTokenSource`

All Known Subinterfaces

`XmlENTITY`, `XmlID`, `XmlIDREF`, `XmlLanguage`, `XmlName`, `XmlNCName`, `XmlNMTOKEN`

Nested Class Summary

`public static final class` **`XmlToken.Factory`**
A class with methods for creating instances of `XmlToken`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlNormalizedString`

`XmlNormalizedString.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlString`

`XmlString.Factory`

Field Summary

`public static final SchemaType` ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlNormalizedString`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlString`

`type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlToken.Factory Class

public static final class ***XmlToken.Factory***

extends `Object`

A class with methods for creating instances of `XmlToken`.

Hierarchy

```
Object
 XmlToken.Factory
```

Enclosing interface

`XmlToken`

Method Summary

`public static XmlToken` ***newInstance***()
Creates an empty instance of `XmlToken`

`public static XmlToken` ***newInstance***(`XmlOptions` *options*)
Creates an empty instance of `XmlToken`

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*)
Returns a validating XMLInputStream.

`public static XMLInputStream` ***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*)
Returns a validating XMLInputStream.

`public static XmlToken` ***newValue***(`Object` *obj*)
Creates an immutable `XmlToken` value

`public static XmlToken` ***parse***(`String` *s*)
Parses a `XmlToken` fragment from a String.

`public static XmlToken` ***parse***(`String` *s*, `XmlOptions` *options*)
Parses a `XmlToken` fragment from a String.

`public static XmlToken` ***parse***(`File` *f*)
Parses a `XmlToken` fragment from a File.

`public static XmlToken` ***parse***(*File f, XmlOptions options*)
Parses a `XmlToken` fragment from a File.

`public static XmlToken` ***parse***(*URL u*)
Parses a `XmlToken` fragment from a URL.

`public static XmlToken` ***parse***(*URL u, XmlOptions options*)
Parses a `XmlToken` fragment from a URL.

`public static XmlToken` ***parse***(*InputStream is*)
Parses a `XmlToken` fragment from an InputStream.

`public static XmlToken` ***parse***(*InputStream is, XmlOptions options*)
Parses a `XmlToken` fragment from an InputStream.

`public static XmlToken` ***parse***(*Reader r*)
Parses a `XmlToken` fragment from a Reader.

`public static XmlToken` ***parse***(*Reader r, XmlOptions options*)
Parses a `XmlToken` fragment from a Reader.

`public static XmlToken` ***parse***(*Node node*)
Parses a `XmlToken` fragment from a DOM Node.

`public static XmlToken` ***parse***(*Node node, XmlOptions options*)
Parses a `XmlToken` fragment from a DOM Node.

`public static XmlToken` ***parse***(*XMLInputStream xis*)
Parses a `XmlToken` fragment from an XMLInputStream.

`public static XmlToken` ***parse***(*XMLInputStream xis, XmlOptions options*)
Parses a `XmlToken` fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlToken newInstance()`

Creates an empty instance of `XmlToken`

---

newInstance(XmlOptions) Method

```
public static XmlToken newInstance(XmlOptions options)
```

Creates an empty instance of `XmlToken`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlToken newValue(Object obj)
```

Creates an immutable `XmlToken` value

---

parse(String) Method

```
public static XmlToken parse(String s)
     throws XmlException
```

Parses a `XmlToken` fragment from a String. For example: "`<xml-fragment>string to collapse</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlToken parse(String s,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlToken` fragment from a String. For example: "`<xml-fragment>string to collapse</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlToken parse(File f)
     throws XmlException, IOException
```

Parses a `XmlToken` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlToken parse(File f,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlToken` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlToken parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlToken` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlToken parse(URL u,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlToken` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlToken parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlToken` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlToken parse(InputStream is,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlToken` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlToken parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlToken` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlToken parse(Reader r,
                             XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlToken` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlToken parse(Node node)
     throws XmlException
```

Parses a `XmlToken` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlToken parse(Node node,
                             XmlOptions options)
     throws XmlException
```

Parses a `XmlToken` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlToken parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlToken` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlToken parse(XMLInputStream xis,
                             XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlToken` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

XmlTokenSource Interface

public interface ***XmlTokenSource***

Represents a holder of XML that can return an `XmlCursor` or copy itself to various media such as `Writer` or `File`. Both `XmlObject` (and thus all XML Beans) and `XmlCursor` are XmlTokenSource implementations.

Related Topics

`XmlObject`
`XmlCursor`

All Known Implementing Classes

`FilterXmlObject`

All Known Subinterfaces

`SimpleValue`, `XmlAnySimpleType`, `XmlAnyURI`, `XmlBase64Binary`, `XmlBoolean`, `XmlByte`, `XmlCursor`, `XmlDate`, `XmlDateTime`, `XmlDecimal`, `XmlDouble`, `XmlDuration`, `XmlENTITIES`, `XmlENTITY`, `XmlFloat`, `XmlGDay`, `XmlGMonth`, `XmlGMonthDay`, `XmlGYear`, `XmlGYearMonth`, `XmlHexBinary`, `XmlID`, `XmlIDREF`, `XmlIDREFS`, `XmlInt`, `XmlInteger`, `XmlLanguage`, `XmlLong`, `XmlName`, `XmlNCName`, `XmlNegativeInteger`, `XmlNMTOKEN`, `XmlNMTOKENS`, `XmlNonNegativeInteger`, `XmlNonPositiveInteger`, `XmlNormalizedString`, `XmlNOTATION`, `XmlObject`, `XmlPositiveInteger`, `XmlQName`, `XmlShort`, `XmlString`, `XmlTime`, `XmlToken`, `XmlUnsignedByte`, `XmlUnsignedInt`, `XmlUnsignedLong`, `XmlUnsignedShort`

Method Summary

`public XmlDocumentProperties` ***documentProperties()***
Returns the XmlDocumentProperties object for the document this token source is associated with.

`public Object` ***monitor()***
Returns the synchronization object for the document.

`public XmlCursor` ***newCursor()***
Returns a new XML cursor.

`public Node` ***newDomNode()***
Returns a W3C DOM Node containing the XML represented by this source.

`public Node`

newDomNode(`XmlOptions` *options)*
Just like newDomNode() but with options.

`public InputStream`
newInputStream()
Returns a new stream containing standard XML text, encoded according to the given encoding.

`public InputStream`
newInputStream(`XmlOptions` *options)*
Just like newInputStream(String encoding) but with options.

`public Reader`
newReader()
Returns a new character reader containing XML text.

`public Reader`
newReader(`XmlOptions` *options)*
Just like newReader() but with options.

`public XMLInputStream`
newXMLInputStream()
Returns a new XmlInputStream.

`public XMLInputStream`
newXMLInputStream(`XmlOptions` *options)*
Just like newXMLInputStream() but with any of a number of options.

`public void`
save(`ContentHandler` *ch*, `LexicalHandler` *lh)*
Writes the XML represented by this source to the given SAX content and lexical handlers.

`public void`
save(`File` *file)*
Writes the XML represented by this source to the given File.

`public void`
save(`OutputStream` *os)*
Writes the XML represented by this source to the given output stream.

`public void`
save(`Writer` *w)*
Writes the XML represented by this source to the given writer.

`public void`
save(`ContentHandler` *ch*, `LexicalHandler` *lh*, `XmlOptions` *options)*
Writes the XML represented by this source to the given SAX content and lexical handlers.

`public void`
save(`File` *file*, `XmlOptions` *options)*
Writes the XML represented by this source to the given File.

`public void`
save(`OutputStream` *os*, `XmlOptions` *options)*
Writes the XML represented by this source to the given output stream.

public void ***save***(*Writer w, XmlOptions options*)
Writes the XML represented by this source to the given writer.

public String ***xmlText***()
Returns standard XML text.

public String ***xmlText***(*XmlOptions options*)
Just like xmlText() but with options.

Method Detail

documentProperties() Method

public XmlDocumentProperties documentProperties()

Returns the XmlDocumentProperties object for the document this token source is associated with.

---

monitor() Method

public Object monitor()

Returns the synchronization object for the document. If concurrent multithreaded access to a document is required, the access should should be protected by synchronizing on this monitor() object. There is one monitor per XML document tree.

---

newCursor() Method

public XmlCursor newCursor()

Returns a new XML cursor. A cursor provides random access to all the tokens in the XML data, plus the ability to extract strongly−typed XmlObjects for the data. If the data is not read−only, the XML cursor also allows modifications to the data. Using a cursor for the first time typically forces the XML document into memory.

---

newDomNode() Method

public Node newDomNode()

Returns a W3C DOM Node containing the XML represented by this source. This is a copy of the XML, it is not a live with the underlying store of this token source. If this is the document node, then a Document is returned, else a DocumentFragment is returned.

---

newDomNode(XmlOptions) Method

public Node newDomNode(XmlOptions *options***)**

Just like newDomNode() but with options. Options map may be null.

Related Topics

`XmlOptions`

---

newInputStream() Method

`public InputStream newInputStream()`

Returns a new stream containing standard XML text, encoded according to the given encoding. The byte stream contains contents starting at the current begin−tag or begin−document and ending at the matching end−tag or end−document. The specified encoding is used and also emitted in a PI at the beginning of the stream. This is a fail−fast stream, so if the underlying data is changed while the stream is being read, the stream throws a ConcurrentModificationException. Throws an IllegalStateException if the XmlTokenSource is not positioned at begin−tag or begin−document (e.g., if it is at an attribute).

---

newInputStream(XmlOptions) Method

`public InputStream newInputStream(XmlOptions options)`

Just like newInputStream(String encoding) but with options. Options map may be null.

Related Topics

`XmlOptions`

---

newReader() Method

`public Reader newReader()`

Returns a new character reader containing XML text. The contents of the reader represents the document contents starting at the current begin−tag or begin−document and ending at the matching end−tag or end−document. No encoding annotation will be made in the text itself. This is a fail−fast reader, so if the underlying data is changed while the reader is being read, the reader throws a ConcurrentModificationException. Throws an IllegalStateException if the XmlTokenSource is not positioned at begin−tag or begin−document (e.g., if it is at an attribute).

---

newReader(XmlOptions) Method

`public Reader newReader(XmlOptions options)`

Just like newReader() but with options. Options map may be null.

Related Topics

newXMLInputStream() Method

DEPRECATED Superceded by JSR 173

public XMLInputStream newXMLInputStream()

Returns a new XmlInputStream. The stream starts at the current begin−tag or begin−document position and ends at the matching end−tag or end−document. This is a fail−fast stream, so if the underlying data is changed while the stream is being read, the stream throws a ConcurrentModificationException. Throws an IllegalStateException if the XmlTokenSource is not positioned at begin−tag or begin−document (e.g., if it is at an attribute).

newXMLInputStream(XmlOptions) Method

DEPRECATED Superceded by JSR 173

public XMLInputStream newXMLInputStream(XmlOptions *options***)**

Just like newXMLInputStream() but with any of a number of options. Use the *options* parameter to specify the following:

To specify this	Use this method
The character encoding to use when converting the character data in the XML to bytess.	`XmlOptions.setCharacterEncoding(String)`
Prefix−to−namespace mappings that should be assumed when saving this XML. This is useful when the resulting XML will be part of a larger XML document, ensuring that this inner document will take advantage of namespaces defined in the outer document.	`XmlOptions.setSaveImplicitNamespaces(Map)`
Suggested namespace prefixes to use when saving. Used only when a namespace attribute needs to be synthesized.	`XmlOptions.setSaveSuggestedPrefixes(Map)`
That namespace attributes should occur first in elements when the XML is saved. By default, they occur last.	`XmlOptions.setSaveNamespacesFirst()`
The XML should be pretty printed when saved. Note that this should only be used for debugging.	`XmlOptions.setSavePrettyPrint()` `XmlOptions.setSavePrettyPrintIndent(int)`

The number of spaces to use when indenting for pretty printing. The default is 2.	
The additional number of spaces indented from the left for pretty printed XML.	`XmlOptions.setSavePrettyPrintOffset(int)`
To minimize the number of namespace attributes generated for the saved XML. Note that this can reduce performance significantly.	`XmlOptions.setSaveAggresiveNamespaces()`
To reduce the size of the saved document by allowing the use of the default namespace. Note that this can potentially change the semantic meaning of the XML if unprefixed QNames are present as the value of an attribute or element.	`XmlOptions.setUseDefaultNamespace()`
To filter out processing instructions with the specified target name.	`XmlOptions.setSaveFilterProcinst(String)`
Change the QName of the synthesized root element when saving. This replaces "xml−fragment" with "fragment" in the namespace http://www.openuri.org/fragment	`XmlOptions.setSaveUseOpenFrag()`
Saving should begin on the element's contents.	`XmlOptions.setSaveInner()`
Saving should begin on the element, rather than its contents.	`XmlOptions.setSaveOuter()`
To rename the document element, or to specify the document element for this XML.	`XmlOptions.setSaveSyntheticDocumentElement(QName)`

Parameters

options
Any of the described options.

Returns

A new validating XMLInputStream.

Related Topics

`XmlOptions`

---

save(ContentHandler, LexicalHandler) Method

```
public void save(ContentHandler ch,
                 LexicalHandler lh)
     throws SAXException
```

Writes the XML represented by this source to the given SAX content and lexical handlers.

Exceptions

SAXException

---

save(File) Method

```
public void save(File file)
     throws IOException
```

Writes the XML represented by this source to the given File.

Exceptions

IOException

---

save(OutputStream) Method

```
public void save(OutputStream os)
     throws IOException
```

Writes the XML represented by this source to the given output stream.

Exceptions

IOException

---

save(Writer) Method

```
public void save(Writer w)
     throws IOException
```

Writes the XML represented by this source to the given writer.

Exceptions

IOException

---

save(ContentHandler, LexicalHandler, XmlOptions) Method

```
public void save(ContentHandler ch,
                 LexicalHandler lh,
                 XmlOptions options)
     throws SAXException
```

Writes the XML represented by this source to the given SAX content and lexical handlers.

Exceptions

SAXException

---

save(File, XmlOptions) Method

```
public void save(File file,
                 XmlOptions options)
     throws IOException
```

Writes the XML represented by this source to the given File.

Exceptions

IOException

---

save(OutputStream, XmlOptions) Method

```
public void save(OutputStream os,
                 XmlOptions options)
     throws IOException
```

Writes the XML represented by this source to the given output stream.

Exceptions

IOException

---

save(Writer, XmlOptions) Method

```
public void save(Writer w,
                 XmlOptions options)
     throws IOException
```

Writes the XML represented by this source to the given writer.

Exceptions

IOException

---

xmlText() Method

`public String xmlText()`

Returns standard XML text. The text returned represents the document contents starting at the current begin−tag or begin−document and ending at the matching end−tag or end−document. This is same content as newReader, but it is returned as a single string. Throws an IllegalStateException if the XmlTokenSource is not positioned at begin−tag or begin−document (e.g., if it is at an attribute).

---

xmlText(XmlOptions) Method

`public String xmlText(XmlOptions` *`options`***`)`**

Just like xmlText() but with options. Options map may be null.

Related Topics

`XmlOptions`

com.bea.xml

XmlUnsignedByte Interface

public interface ***XmlUnsignedByte***

extends `XmlUnsignedShort`

Corresponds to the XML Schema xs:unsignedByte type. One of the derived types based on xs:decimal.

Verified to be in the range 0..255 when validating.

As suggested by JAXB, convertible to Java short.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlNonNegativeInteger`, `XmlObject`, `XmlTokenSource`, `XmlUnsignedInt`, `XmlUnsignedLong`, `XmlUnsignedShort`

Nested Class Summary

`public static final class`	**`XmlUnsignedByte.Factory`** A class with methods for creating instances of `XmlUnsignedByte`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlNonNegativeInteger`

`XmlNonNegativeInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlUnsignedInt`

`XmlUnsignedInt.Factory`

Nested classes from interface com.bea.xml.`XmlUnsignedLong`

`XmlUnsignedLong.Factory`

Nested classes from interface com.bea.xml.`XmlUnsignedShort`

`XmlUnsignedShort.Factory`

Field Summary

`public static final SchemaType` ***`type`***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlNonNegativeInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlUnsignedInt`

`type`

Fields from interface com.bea.xml.`XmlUnsignedLong`

`type`

Fields from interface com.bea.xml.`XmlUnsignedShort`

`type`

Method Summary

`public short` ***getShortValue***()
Returns this value as a short

`public void` ***set***(*short s*)
Sets this value as a short

`public void` ***setShortValue***(*short s*)
Sets this value as a short

`public short` ***shortValue***()
Returns this value as a short

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Methods from interface `com.bea.xml.XmlUnsignedInt`

`getLongValue, longValue, set, setLongValue`

Methods from interface `com.bea.xml.XmlUnsignedShort`

`getIntValue, intValue, set, setIntValue`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getShortValue() Method

`public short getShortValue()`

Returns this value as a short

---

set(short) Method

DEPRECATED replaced with `com.bea.xml.XmlUnsignedByte.setShortValue(short)`

`public void set(short` *`s`***`)`**

Sets this value as a short

---

setShortValue(short) Method

`public void setShortValue(short` *`s`***`)`**

Sets this value as a short

---

shortValue() Method

DEPRECATED replaced with `com.bea.xml.XmlUnsignedByte.getShortValue()`

`public short shortValue()`

Returns this value as a short

XmlUnsignedByte.Factory Class

public static final class ***XmlUnsignedByte.Factory***

extends `Object`

A class with methods for creating instances of `XmlUnsignedByte`.

Hierarchy

```
Object
 XmlUnsignedByte.Factory
```

Enclosing interface

`XmlUnsignedByte`

Method Summary

public static XmlUnsignedByte	***newInstance***() Creates an empty instance of `XmlUnsignedByte`
public static XmlUnsignedByte	***newInstance***(`XmlOptions` *options*) Creates an empty instance of `XmlUnsignedByte`
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*) Returns a validating XMLInputStream.
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*) Returns a validating XMLInputStream.
public static XmlUnsignedByte	***newValue***(`Object` *obj*) Creates an immutable `XmlUnsignedByte` value
public static XmlUnsignedByte	***parse***(`String` *s*) Parses a `XmlUnsignedByte` fragment from a String.
public static XmlUnsignedByte	***parse***(`String` *s*, `XmlOptions` *options*) Parses a `XmlUnsignedByte` fragment from a String.
public static XmlUnsignedByte	***parse***(`File` *f*) Parses a `XmlUnsignedByte` fragment from a File.

public static XmlUnsignedByte **parse**(File f, XmlOptions options)
Parses a XmlUnsignedByte fragment from a File.

public static XmlUnsignedByte **parse**(URL u)
Parses a XmlUnsignedByte fragment from a URL.

public static XmlUnsignedByte **parse**(URL u, XmlOptions options)
Parses a XmlUnsignedByte fragment from a URL.

public static XmlUnsignedByte **parse**(InputStream is)
Parses a XmlUnsignedByte fragment from an InputStream.

public static XmlUnsignedByte **parse**(InputStream is, XmlOptions options)
Parses a XmlUnsignedByte fragment from an InputStream.

public static XmlUnsignedByte **parse**(Reader r)
Parses a XmlUnsignedByte fragment from a Reader.

public static XmlUnsignedByte **parse**(Reader r, XmlOptions options)
Parses a XmlUnsignedByte fragment from a Reader.

public static XmlUnsignedByte **parse**(Node node)
Parses a XmlUnsignedByte fragment from a DOM Node.

public static XmlUnsignedByte **parse**(Node node, XmlOptions options)
Parses a XmlUnsignedByte fragment from a DOM Node.

public static XmlUnsignedByte **parse**(XMLInputStream xis)
Parses a XmlUnsignedByte fragment from an XMLInputStream.

public static XmlUnsignedByte **parse**(XMLInputStream xis, XmlOptions options)
Parses a XmlUnsignedByte fragment from an XMLInputStream.

Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

newInstance() Method

public static XmlUnsignedByte newInstance()

Creates an empty instance of XmlUnsignedByte

---

newInstance(XmlOptions) Method

```
public static XmlUnsignedByte newInstance(XmlOptions options)
```

Creates an empty instance of `XmlUnsignedByte`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlUnsignedByte newValue(Object obj)
```

Creates an immutable `XmlUnsignedByte` value

---

parse(String) Method

```
public static XmlUnsignedByte parse(String s)
     throws XmlException
```

Parses a `XmlUnsignedByte` fragment from a String. For example: "`<xml-fragment>123</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlUnsignedByte parse(String s,
                                    XmlOptions options)
     throws XmlException
```

Parses a `XmlUnsignedByte` fragment from a String. For example: "`<xml-fragment>123</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlUnsignedByte parse(File f)
     throws XmlException, IOException
```

Parses a `XmlUnsignedByte` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlUnsignedByte parse(File f,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedByte` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlUnsignedByte parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlUnsignedByte` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlUnsignedByte parse(URL u,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedByte` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlUnsignedByte parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlUnsignedByte` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlUnsignedByte parse(InputStream is,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedByte` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlUnsignedByte parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlUnsignedByte` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlUnsignedByte parse(Reader r,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedByte` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlUnsignedByte parse(Node node)
     throws XmlException
```

Parses a `XmlUnsignedByte` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlUnsignedByte parse(Node node,
                                    XmlOptions options)
     throws XmlException
```

Parses a `XmlUnsignedByte` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlUnsignedByte parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlUnsignedByte` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlUnsignedByte parse(XMLInputStream xis,
                                    XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlUnsignedByte` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlUnsignedInt Interface

public interface ***XmlUnsignedInt***

extends `XmlUnsignedLong`

Corresponds to the XML Schema xs:unsignedInt type. One of the derived types based on xs:decimal.

Verified to be in the range 0..4294967295 when validating.

Convertible to Java long.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlNonNegativeInteger`, `XmlObject`, `XmlTokenSource`, `XmlUnsignedLong`

All Known Subinterfaces

`XmlUnsignedByte`, `XmlUnsignedShort`

Nested Class Summary

`public static final class`	**`XmlUnsignedInt.Factory`** A class with methods for creating instances of `XmlUnsignedInt`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlNonNegativeInteger`

`XmlNonNegativeInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlUnsignedLong`

`XmlUnsignedLong.Factory`

Field Summary

`public static final SchemaType` ***type***
The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlNonNegativeInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlUnsignedLong`

`type`

Method Summary

`public long` ***getLongValue()***
Returns this value as a long

`public long` ***longValue()***
Returns this value as a long

`public void` ***set****(long v)*

Sets this value as a long

public
void ***setLongValue***(*long v*)
Sets this value as a long

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getLongValue() Method

`public long getLongValue()`

Returns this value as a long

longValue() Method

DEPRECATED replaced with `com.bea.xml.XmlUnsignedInt.getLongValue()`

`public long longValue()`

Returns this value as a long

---

set(long) Method

DEPRECATED replaced with `com.bea.xml.XmlUnsignedInt.setLongValue(long)`

`public void set(long v)`

Sets this value as a long

---

setLongValue(long) Method

`public void setLongValue(long v)`

Sets this value as a long

XmlUnsignedInt.Factory Class

public static final class ***XmlUnsignedInt.Factory***

extends `Object`

A class with methods for creating instances of `XmlUnsignedInt`.

Hierarchy

```
Object
 XmlUnsignedInt.Factory
```

Enclosing interface

`XmlUnsignedInt`

Method Summary

public static XmlUnsignedInt	***newInstance***() Creates an empty instance of `XmlUnsignedInt`
public static XmlUnsignedInt	***newInstance***(`XmlOptions` *options*) Creates an empty instance of `XmlUnsignedInt`
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*) Returns a validating XMLInputStream.
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*) Returns a validating XMLInputStream.
public static XmlUnsignedInt	***newValue***(`Object` *obj*) Creates an immutable `XmlUnsignedInt` value
public static XmlUnsignedInt	***parse***(`String` *s*) Parses a `XmlUnsignedInt` fragment from a String.
public static XmlUnsignedInt	***parse***(`String` *s*, `XmlOptions` *options*) Parses a `XmlUnsignedInt` fragment from a String.
public static XmlUnsignedInt	***parse***(`File` *f*) Parses a `XmlUnsignedInt` fragment from a File.

public static XmlUnsignedInt	***parse***(*File f, XmlOptions options*) Parses a XmlUnsignedInt fragment from a File.
public static XmlUnsignedInt	***parse***(*URL u*) Parses a XmlUnsignedInt fragment from a URL.
public static XmlUnsignedInt	***parse***(*URL u, XmlOptions options*) Parses a XmlUnsignedInt fragment from a URL.
public static XmlUnsignedInt	***parse***(*InputStream is*) Parses a XmlUnsignedInt fragment from an InputStream.
public static XmlUnsignedInt	***parse***(*InputStream is, XmlOptions options*) Parses a XmlUnsignedInt fragment from an InputStream.
public static XmlUnsignedInt	***parse***(*Reader r*) Parses a XmlUnsignedInt fragment from a Reader.
public static XmlUnsignedInt	***parse***(*Reader r, XmlOptions options*) Parses a XmlUnsignedInt fragment from a Reader.
public static XmlUnsignedInt	***parse***(*Node node*) Parses a XmlUnsignedInt fragment from a DOM Node.
public static XmlUnsignedInt	***parse***(*Node node, XmlOptions options*) Parses a XmlUnsignedInt fragment from a DOM Node.
public static XmlUnsignedInt	***parse***(*XMLInputStream xis*) Parses a XmlUnsignedInt fragment from an XMLInputStream.
public static XmlUnsignedInt	***parse***(*XMLInputStream xis, XmlOptions options*) Parses a XmlUnsignedInt fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlUnsignedInt newInstance()`

Creates an empty instance of XmlUnsignedInt

---

newInstance(XmlOptions) Method

```
public static XmlUnsignedInt newInstance(XmlOptions options)
```

Creates an empty instance of `XmlUnsignedInt`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlUnsignedInt newValue(Object obj)
```

Creates an immutable `XmlUnsignedInt` value

---

parse(String) Method

```
public static XmlUnsignedInt parse(String s)
     throws XmlException
```

Parses a `XmlUnsignedInt` fragment from a String. For example: "`<xml-fragment>1234567</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlUnsignedInt parse(String s,
                                   XmlOptions options)
     throws XmlException
```

Parses a `XmlUnsignedInt` fragment from a String. For example: "`<xml-fragment>1234567</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlUnsignedInt parse(File f)
     throws XmlException, IOException
```

Parses a `XmlUnsignedInt` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlUnsignedInt parse(File f,
                                   XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedInt` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlUnsignedInt parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlUnsignedInt` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlUnsignedInt parse(URL u,
                                   XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedInt` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlUnsignedInt parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlUnsignedInt` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlUnsignedInt parse(InputStream is,
                                   XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedInt` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlUnsignedInt parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlUnsignedInt` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlUnsignedInt parse(Reader r,
                                   XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedInt` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlUnsignedInt parse(Node node)
     throws XmlException
```

Parses a `XmlUnsignedInt` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlUnsignedInt parse(Node node,
                                   XmlOptions options)
     throws XmlException
```

Parses a `XmlUnsignedInt` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlUnsignedInt parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlUnsignedInt` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlUnsignedInt parse(XMLInputStream xis,
                                   XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlUnsignedInt` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlUnsignedLong Interface

public interface ***XmlUnsignedLong***

extends `XmlNonNegativeInteger`

Corresponds to the XML Schema xs:unsignedLong type. One of the derived types based on xs:decimal.

Verified to be in the range 0..264−1 when validating.

Convertible to `BigInteger`.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlNonNegativeInteger`, `XmlObject`, `XmlTokenSource`

All Known Subinterfaces

`XmlUnsignedByte`, `XmlUnsignedInt`, `XmlUnsignedShort`

Nested Class Summary

`public static final class`	**`XmlUnsignedLong.Factory`** A class with methods for creating instances of `XmlUnsignedLong`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlNonNegativeInteger`

`XmlNonNegativeInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Field Summary

`public static final SchemaType` ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlNonNegativeInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Method Summary

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

XmlUnsignedLong.Factory Class

public static final class ***XmlUnsignedLong.Factory***

extends `Object`

A class with methods for creating instances of `XmlUnsignedLong`.

Hierarchy

```
Object
 XmlUnsignedLong.Factory
```

Enclosing interface

`XmlUnsignedLong`

Method Summary

public static XmlUnsignedLong	***newInstance***() Creates an empty instance of `XmlUnsignedLong`
public static XmlUnsignedLong	***newInstance***(`XmlOptions` *options*) Creates an empty instance of `XmlUnsignedLong`
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*) Returns a validating XMLInputStream.
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*) Returns a validating XMLInputStream.
public static XmlUnsignedLong	***newValue***(`Object` *obj*) Creates an immutable `XmlUnsignedLong` value
public static XmlUnsignedLong	***parse***(`String` *s*) Parses a `XmlUnsignedLong` fragment from a String.
public static XmlUnsignedLong	***parse***(`String` *s*, `XmlOptions` *options*) Parses a `XmlUnsignedLong` fragment from a String.
public static XmlUnsignedLong	***parse***(`File` *f*) Parses a `XmlUnsignedLong` fragment from a File.

public static XmlUnsignedLong	***parse***(File f, XmlOptions options) Parses a XmlUnsignedLong fragment from a File.
public static XmlUnsignedLong	***parse***(URL u) Parses a XmlUnsignedLong fragment from a URL.
public static XmlUnsignedLong	***parse***(URL u, XmlOptions options) Parses a XmlUnsignedLong fragment from a URL.
public static XmlUnsignedLong	***parse***(InputStream is) Parses a XmlUnsignedLong fragment from an InputStream.
public static XmlUnsignedLong	***parse***(InputStream is, XmlOptions options) Parses a XmlUnsignedLong fragment from an InputStream.
public static XmlUnsignedLong	***parse***(Reader r) Parses a XmlUnsignedLong fragment from a Reader.
public static XmlUnsignedLong	***parse***(Reader r, XmlOptions options) Parses a XmlUnsignedLong fragment from a Reader.
public static XmlUnsignedLong	***parse***(Node node) Parses a XmlUnsignedLong fragment from a DOM Node.
public static XmlUnsignedLong	***parse***(Node node, XmlOptions options) Parses a XmlUnsignedLong fragment from a DOM Node.
public static XmlUnsignedLong	***parse***(XMLInputStream xis) Parses a XmlUnsignedLong fragment from an XMLInputStream.
public static XmlUnsignedLong	***parse***(XMLInputStream xis, XmlOptions options) Parses a XmlUnsignedLong fragment from an XMLInputStream.

Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

newInstance() Method

`public static XmlUnsignedLong newInstance()`

Creates an empty instance of XmlUnsignedLong

---

newInstance(XmlOptions) Method

```
public static XmlUnsignedLong newInstance(XmlOptions options)
```

Creates an empty instance of `XmlUnsignedLong`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

newValue(Object) Method

```
public static XmlUnsignedLong newValue(Object obj)
```

Creates an immutable `XmlUnsignedLong` value

---

parse(String) Method

```
public static XmlUnsignedLong parse(String s)
     throws XmlException
```

Parses a `XmlUnsignedLong` fragment from a String. For example: "`<xml-fragment>123456789</xml-fragment>`".

Exceptions

`XmlException`

---

parse(String, XmlOptions) Method

```
public static XmlUnsignedLong parse(String s,
                                    XmlOptions options)
     throws XmlException
```

Parses a `XmlUnsignedLong` fragment from a String. For example: "`<xml-fragment>123456789</xml-fragment>`".

Exceptions

`XmlException`

---

parse(File) Method

```
public static XmlUnsignedLong parse(File f)
     throws XmlException, IOException
```

Parses a `XmlUnsignedLong` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(File, XmlOptions) Method

```
public static XmlUnsignedLong parse(File f,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedLong` fragment from a File.

Exceptions

`XmlException`
`IOException`

---

parse(URL) Method

```
public static XmlUnsignedLong parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlUnsignedLong` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlUnsignedLong parse(URL u,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedLong` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlUnsignedLong parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlUnsignedLong` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlUnsignedLong parse(InputStream is,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedLong` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlUnsignedLong parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlUnsignedLong` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlUnsignedLong parse(Reader r,
                                    XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedLong` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlUnsignedLong parse(Node node)
     throws XmlException
```

Parses a `XmlUnsignedLong` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlUnsignedLong parse(Node node,
                                    XmlOptions options)
     throws XmlException
```

Parses a `XmlUnsignedLong` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlUnsignedLong parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlUnsignedLong` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlUnsignedLong parse(XMLInputStream xis,
                                    XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlUnsignedLong` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

com.bea.xml

XmlUnsignedShort Interface

public interface ***XmlUnsignedShort***

extends `XmlUnsignedInt`

Corresponds to the XML Schema xs:unsignedShort type. One of the derived types based on xs:decimal.

Verified to be in the range 0..65535 when validating.

Convertible to a Java int.

All Superinterfaces

`XmlAnySimpleType`, `XmlDecimal`, `XmlInteger`, `XmlNonNegativeInteger`, `XmlObject`, `XmlTokenSource`, `XmlUnsignedInt`, `XmlUnsignedLong`

All Known Subinterfaces

`XmlUnsignedByte`

Nested Class Summary

`public static final class` **`XmlUnsignedShort.Factory`**
A class with methods for creating instances of `XmlUnsignedShort`.

Nested classes from interface com.bea.xml.`XmlAnySimpleType`

`XmlAnySimpleType.Factory`

Nested classes from interface com.bea.xml.`XmlDecimal`

`XmlDecimal.Factory`

Nested classes from interface com.bea.xml.`XmlInteger`

`XmlInteger.Factory`

Nested classes from interface com.bea.xml.`XmlNonNegativeInteger`

`XmlNonNegativeInteger.Factory`

Nested classes from interface com.bea.xml.`XmlObject`

`XmlObject.Factory`

Nested classes from interface com.bea.xml.`XmlUnsignedInt`

`XmlUnsignedInt.Factory`

Nested classes from interface com.bea.xml.`XmlUnsignedLong`

`XmlUnsignedLong.Factory`

Field Summary

`public static final SchemaType` ***type***	The constant `SchemaType` object representing this schema type.

Fields from interface com.bea.xml.`XmlAnySimpleType`

`type`

Fields from interface com.bea.xml.`XmlDecimal`

`type`

Fields from interface com.bea.xml.`XmlInteger`

`type`

Fields from interface com.bea.xml.`XmlNonNegativeInteger`

`type`

Fields from interface com.bea.xml.`XmlObject`

`EQUAL, GREATER_THAN, LESS_THAN, NOT_EQUAL, type`

Fields from interface com.bea.xml.`XmlUnsignedInt`

`type`

Fields from interface com.bea.xml.`XmlUnsignedLong`

`type`

Method Summary

`public int` ***getIntValue()***

Returns this value as an int

`public int` ***intValue***()

Returns this value as an int

`public void` ***set***(*int v*)

Sets this value as an int

`public void` ***setIntValue***(*int v*)

Sets this value as an int

Methods from interface `com.bea.xml.XmlAnySimpleType`

`getStringValue, set, setStringValue, stringValue`

Methods from interface `com.bea.xml.XmlDecimal`

`bigDecimalValue, getBigDecimalValue, set, setBigDecimalValue`

Methods from interface `com.bea.xml.XmlInteger`

`bigIntegerValue, getBigIntegerValue, set, setBigIntegerValue`

Methods from interface `com.bea.xml.XmlObject`

`changeType, compareTo, compareValue, copy, execQuery, execQuery, isImmutable, isNil, schemaType, selectPath, selectPath, set, setNil, toString, validate, validate, valueEquals, valueHashCode`

Methods from interface `com.bea.xml.XmlTokenSource`

`documentProperties, monitor, newCursor, newDomNode, newDomNode, newInputStream, newInputStream, newReader, newReader, newXMLInputStream, newXMLInputStream, save, save, save, save, save, save, save, save, xmlText, xmlText`

Methods from interface `com.bea.xml.XmlUnsignedInt`

`getLongValue, longValue, set, setLongValue`

Field Detail

type

`public static final SchemaType type`

The constant `SchemaType` object representing this schema type.

Method Detail

getIntValue() Method

`public int getIntValue()`

Returns this value as an int

---

intValue() Method

DEPRECATED replaced with `com.bea.xml.XmlUnsignedShort.getIntValue()`

`public int intValue()`

Returns this value as an int

---

set(int) Method

DEPRECATED replaced with `com.bea.xml.XmlUnsignedShort.setIntValue(int)`

`public void set(int v)`

Sets this value as an int

---

setIntValue(int) Method

`public void setIntValue(int v)`

Sets this value as an int

XmlUnsignedShort.Factory Class

public static final class ***XmlUnsignedShort.Factory***

extends `Object`

A class with methods for creating instances of `XmlUnsignedShort`.

Hierarchy

```
Object
 XmlUnsignedShort.Factory
```

Enclosing interface

`XmlUnsignedShort`

Method Summary

public static XmlUnsignedShort	***newInstance***() Creates an empty instance of `XmlUnsignedShort`
public static XmlUnsignedShort	***newInstance***(`XmlOptions` *options*) Creates an empty instance of `XmlUnsignedShort`
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*) Returns a validating XMLInputStream.
public static XMLInputStream	***newValidatingXMLInputStream***(`XMLInputStream` *xis*, `XmlOptions` *options*) Returns a validating XMLInputStream.
public static XmlUnsignedShort	***newValue***(`Object` *obj*) Creates an immutable `XmlUnsignedShort` value
public static XmlUnsignedShort	***parse***(`String` *s*) Parses a `XmlUnsignedShort` fragment from a String.
public static XmlUnsignedShort	***parse***(`String` *s*, `XmlOptions` *options*) Parses a `XmlUnsignedShort` fragment from a String.
public static XmlUnsignedShort	***parse***(`File` *f*) Parses a `XmlUnsignedShort` fragment from a File.

public static XmlUnsignedShort	***parse***(*File f, XmlOptions options*) Parses a XmlUnsignedShort fragment from a File.
public static XmlUnsignedShort	***parse***(*URL u*) Parses a XmlUnsignedShort fragment from a URL.
public static XmlUnsignedShort	***parse***(*URL u, XmlOptions options*) Parses a XmlUnsignedShort fragment from a URL.
public static XmlUnsignedShort	***parse***(*InputStream is*) Parses a XmlUnsignedShort fragment from an InputStream.
public static XmlUnsignedShort	***parse***(*InputStream is, XmlOptions options*) Parses a XmlUnsignedShort fragment from an InputStream.
public static XmlUnsignedShort	***parse***(*Reader r*) Parses a XmlUnsignedShort fragment from a Reader.
public static XmlUnsignedShort	***parse***(*Reader r, XmlOptions options*) Parses a XmlUnsignedShort fragment from a Reader.
public static XmlUnsignedShort	***parse***(*Node node*) Parses a XmlUnsignedShort fragment from a DOM Node.
public static XmlUnsignedShort	***parse***(*Node node, XmlOptions options*) Parses a XmlUnsignedShort fragment from a DOM Node.
public static XmlUnsignedShort	***parse***(*XMLInputStream xis*) Parses a XmlUnsignedShort fragment from an XMLInputStream.
public static XmlUnsignedShort	***parse***(*XMLInputStream xis, XmlOptions options*) Parses a XmlUnsignedShort fragment from an XMLInputStream.

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Method Detail

newInstance() Method

`public static XmlUnsignedShort newInstance()`

Creates an empty instance of XmlUnsignedShort

---

newInstance(XmlOptions) Method

```
public static XmlUnsignedShort newInstance(XmlOptions options)
```

Creates an empty instance of `XmlUnsignedShort`

---

newValidatingXMLInputStream(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValidatingXMLInputStream(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XMLInputStream newValidatingXMLInputStream(XMLInputStream xis,
                                                         XmlOptions options)
     throws XmlException, XMLStreamException
```

Returns a validating XMLInputStream.

Exceptions

XmlException
XMLStreamException

---

newValue(Object) Method

```
public static XmlUnsignedShort newValue(Object obj)
```

Creates an immutable `XmlUnsignedShort` value

---

parse(String) Method

```
public static XmlUnsignedShort parse(String s)
     throws XmlException
```

Parses a `XmlUnsignedShort` fragment from a String. For example: "`<xml-fragment>12345</xml-fragment>`".

Exceptions

XmlException

---

parse(String, XmlOptions) Method

```
public static XmlUnsignedShort parse(String s,
                                     XmlOptions options)
     throws XmlException
```

Parses a `XmlUnsignedShort` fragment from a String. For example: "`<xml-fragment>12345</xml-fragment>`".

Exceptions

XmlException

---

parse(File) Method

```
public static XmlUnsignedShort parse(File f)
     throws XmlException, IOException
```

Parses a `XmlUnsignedShort` fragment from a File.

Exceptions

XmlException
IOException

---

parse(File, XmlOptions) Method

```
public static XmlUnsignedShort parse(File f,
                                     XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedShort` fragment from a File.

Exceptions

XmlException
IOException

---

parse(URL) Method

```
public static XmlUnsignedShort parse(URL u)
     throws XmlException, IOException
```

Parses a `XmlUnsignedShort` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(URL, XmlOptions) Method

```
public static XmlUnsignedShort parse(URL u,
                                     XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedShort` fragment from a URL.

Exceptions

XmlException
IOException

---

parse(InputStream) Method

```
public static XmlUnsignedShort parse(InputStream is)
     throws XmlException, IOException
```

Parses a `XmlUnsignedShort` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(InputStream, XmlOptions) Method

```
public static XmlUnsignedShort parse(InputStream is,
                                     XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedShort` fragment from an InputStream.

Exceptions

XmlException
IOException

---

parse(Reader) Method

```
public static XmlUnsignedShort parse(Reader r)
     throws XmlException, IOException
```

Parses a `XmlUnsignedShort` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Reader, XmlOptions) Method

```
public static XmlUnsignedShort parse(Reader r,
                                     XmlOptions options)
     throws XmlException, IOException
```

Parses a `XmlUnsignedShort` fragment from a Reader.

Exceptions

XmlException
IOException

---

parse(Node) Method

```
public static XmlUnsignedShort parse(Node node)
     throws XmlException
```

Parses a `XmlUnsignedShort` fragment from a DOM Node.

Exceptions

XmlException

---

parse(Node, XmlOptions) Method

```
public static XmlUnsignedShort parse(Node node,
                                     XmlOptions options)
     throws XmlException
```

Parses a `XmlUnsignedShort` fragment from a DOM Node.

Exceptions

`XmlException`

---

parse(XMLInputStream) Method

DEPRECATED Superceded by JSR 173

```
public static XmlUnsignedShort parse(XMLInputStream xis)
     throws XmlException, XMLStreamException
```

Parses a `XmlUnsignedShort` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`

---

parse(XMLInputStream, XmlOptions) Method

DEPRECATED Superceded by JSR 173

```
public static XmlUnsignedShort parse(XMLInputStream xis,
                                     XmlOptions options)
     throws XmlException, XMLStreamException
```

Parses a `XmlUnsignedShort` fragment from an XMLInputStream.

Exceptions

`XmlException`
`XMLStreamException`