| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DatatypeConverterInterface.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/xml/bind/DatatypeConverter.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/Element.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/DatatypeConverterInterface.html)    [**NO FRAMES**](http://docs.google.com/DatatypeConverterInterface.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **javax.xml.bind**

Interface DatatypeConverterInterface

public interface **DatatypeConverterInterface**

The DatatypeConverterInterface is for JAXB provider use only. A JAXB provider must supply a class that implements this interface. JAXB Providers are required to call the [DatatypeConverter.setDatatypeConverter](http://docs.google.com/javax/xml/bind/DatatypeConverter.html#setDatatypeConverter(javax.xml.bind.DatatypeConverterInterface)) api at some point before the first marshal or unmarshal operation (perhaps during the call to JAXBContext.newInstance). This step is necessary to configure the converter that should be used to perform the print and parse functionality. Calling this api repeatedly will have no effect - the DatatypeConverter instance passed into the first invocation is the one that will be used from then on.

This interface defines the parse and print methods. There is one parse and print method for each XML schema datatype specified in the the default binding Table 5-1 in the JAXB specification.

The parse and print methods defined here are invoked by the static parse and print methods defined in the [DatatypeConverter](http://docs.google.com/javax/xml/bind/DatatypeConverter.html) class.

A parse method for a XML schema datatype must be capable of converting any lexical representation of the XML schema datatype ( specified by the  [XML Schema Part2: Datatypes specification](http://www.w3.org/TR/xmlschema-2/) into a value in the value space of the XML schema datatype. If an error is encountered during conversion, then an IllegalArgumentException or a subclass of IllegalArgumentException must be thrown by the method.

A print method for a XML schema datatype can output any lexical representation that is valid with respect to the XML schema datatype. If an error is encountered during conversion, then an IllegalArgumentException, or a subclass of IllegalArgumentException must be thrown by the method.

The prefix xsd: is used to refer to XML schema datatypes  [XML Schema Part2: Datatypes specification.](http://www.w3.org/TR/xmlschema-2/)

**Since:** JAXB1.0 **See Also:**[DatatypeConverter](http://docs.google.com/javax/xml/bind/DatatypeConverter.html), [ParseConversionEvent](http://docs.google.com/javax/xml/bind/ParseConversionEvent.html), [PrintConversionEvent](http://docs.google.com/javax/xml/bind/PrintConversionEvent.html)

| **Method Summary** | |
| --- | --- |
| [String](http://docs.google.com/java/lang/String.html) | [**parseAnySimpleType**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseAnySimpleType(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDAnySimpleType)             Return a string containing the lexical representation of the simple type. |
| byte[] | [**parseBase64Binary**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseBase64Binary(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDBase64Binary)             Converts the string argument into an array of bytes. |
| boolean | [**parseBoolean**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseBoolean(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDBoolean)             Converts the string argument into a boolean value. |
| byte | [**parseByte**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseByte(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDByte)             Converts the string argument into a byte value. |
| [Calendar](http://docs.google.com/java/util/Calendar.html) | [**parseDate**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseDate(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDDate)             Converts the string argument into a Calendar value. |
| [Calendar](http://docs.google.com/java/util/Calendar.html) | [**parseDateTime**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseDateTime(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDDateTime)             Converts the string argument into a Calendar value. |
| [BigDecimal](http://docs.google.com/java/math/BigDecimal.html) | [**parseDecimal**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseDecimal(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDDecimal)             Converts the string argument into a BigDecimal value. |
| double | [**parseDouble**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseDouble(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDDouble)             Converts the string argument into a double value. |
| float | [**parseFloat**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseFloat(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDFloat)             Converts the string argument into a float value. |
| byte[] | [**parseHexBinary**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseHexBinary(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDHexBinary)             Converts the string argument into an array of bytes. |
| int | [**parseInt**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseInt(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDInt)             Convert the string argument into an int value. |
| [BigInteger](http://docs.google.com/java/math/BigInteger.html) | [**parseInteger**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseInteger(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDInteger)             Convert the string argument into a BigInteger value. |
| long | [**parseLong**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseLong(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDLong)             Converts the string argument into a long value. |
| [QName](http://docs.google.com/javax/xml/namespace/QName.html) | [**parseQName**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseQName(java.lang.String,%20javax.xml.namespace.NamespaceContext))([String](http://docs.google.com/java/lang/String.html) lexicalXSDQName, [NamespaceContext](http://docs.google.com/javax/xml/namespace/NamespaceContext.html) nsc)             Converts the string argument into a QName value. |
| short | [**parseShort**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseShort(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDShort)             Converts the string argument into a short value. |
| [String](http://docs.google.com/java/lang/String.html) | [**parseString**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseString(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDString)             Convert the string argument into a string. |
| [Calendar](http://docs.google.com/java/util/Calendar.html) | [**parseTime**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseTime(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDTime)             Converts the string argument into a Calendar value. |
| long | [**parseUnsignedInt**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseUnsignedInt(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDUnsignedInt)             Converts the string argument into a long value. |
| int | [**parseUnsignedShort**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#parseUnsignedShort(java.lang.String))([String](http://docs.google.com/java/lang/String.html) lexicalXSDUnsignedShort)             Converts the string argument into an int value. |
| [String](http://docs.google.com/java/lang/String.html) | [**printAnySimpleType**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printAnySimpleType(java.lang.String))([String](http://docs.google.com/java/lang/String.html) val)             Converts a string value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printBase64Binary**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printBase64Binary(byte%5B%5D))(byte[] val)             Converts an array of bytes into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printBoolean**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printBoolean(boolean))(boolean val)             Converts a boolean value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printByte**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printByte(byte))(byte val)             Converts a byte value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printDate**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printDate(java.util.Calendar))([Calendar](http://docs.google.com/java/util/Calendar.html) val)             Converts a Calendar value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printDateTime**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printDateTime(java.util.Calendar))([Calendar](http://docs.google.com/java/util/Calendar.html) val)             Converts a Calendar value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printDecimal**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printDecimal(java.math.BigDecimal))([BigDecimal](http://docs.google.com/java/math/BigDecimal.html) val)             Converts a BigDecimal value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printDouble**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printDouble(double))(double val)             Converts a double value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printFloat**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printFloat(float))(float val)             Converts a float value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printHexBinary**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printHexBinary(byte%5B%5D))(byte[] val)             Converts an array of bytes into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printInt**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printInt(int))(int val)             Converts an int value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printInteger**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printInteger(java.math.BigInteger))([BigInteger](http://docs.google.com/java/math/BigInteger.html) val)             Converts a BigInteger value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printLong**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printLong(long))(long val)             Converts a long value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printQName**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printQName(javax.xml.namespace.QName,%20javax.xml.namespace.NamespaceContext))([QName](http://docs.google.com/javax/xml/namespace/QName.html) val, [NamespaceContext](http://docs.google.com/javax/xml/namespace/NamespaceContext.html) nsc)             Converts a QName instance into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printShort**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printShort(short))(short val)             Converts a short value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printString**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printString(java.lang.String))([String](http://docs.google.com/java/lang/String.html) val)             Converts the string argument into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printTime**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printTime(java.util.Calendar))([Calendar](http://docs.google.com/java/util/Calendar.html) val)             Converts a Calendar value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printUnsignedInt**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printUnsignedInt(long))(long val)             Converts a long value into a string. |
| [String](http://docs.google.com/java/lang/String.html) | [**printUnsignedShort**](http://docs.google.com/javax/xml/bind/DatatypeConverterInterface.html#printUnsignedShort(int))(int val)             Converts an int value into a string. |

| **Method Detail** |
| --- |

### parseString

[String](http://docs.google.com/java/lang/String.html) **parseString**([String](http://docs.google.com/java/lang/String.html) lexicalXSDString)

Convert the string argument into a string.

**Parameters:**lexicalXSDString - A lexical representation of the XML Schema datatype xsd:string **Returns:**A string that is the same as the input string.

### parseInteger

[BigInteger](http://docs.google.com/java/math/BigInteger.html) **parseInteger**([String](http://docs.google.com/java/lang/String.html) lexicalXSDInteger)

Convert the string argument into a BigInteger value.

**Parameters:**lexicalXSDInteger - A string containing a lexical representation of xsd:integer. **Returns:**A BigInteger value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - lexicalXSDInteger is not a valid string representation of a [BigInteger](http://docs.google.com/java/math/BigInteger.html) value.

### parseInt

int **parseInt**([String](http://docs.google.com/java/lang/String.html) lexicalXSDInt)

Convert the string argument into an int value.

**Parameters:**lexicalXSDInt - A string containing a lexical representation of xsd:int. **Returns:**An int value represented byte the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - lexicalXSDInt is not a valid string representation of an int value.

### parseLong

long **parseLong**([String](http://docs.google.com/java/lang/String.html) lexicalXSDLong)

Converts the string argument into a long value.

**Parameters:**lexicalXSDLong - A string containing lexical representation of xsd:long. **Returns:**A long value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - lexicalXSDLong is not a valid string representation of a long value.

### parseShort

short **parseShort**([String](http://docs.google.com/java/lang/String.html) lexicalXSDShort)

Converts the string argument into a short value.

**Parameters:**lexicalXSDShort - A string containing lexical representation of xsd:short. **Returns:**A short value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - lexicalXSDShort is not a valid string representation of a short value.

### parseDecimal

[BigDecimal](http://docs.google.com/java/math/BigDecimal.html) **parseDecimal**([String](http://docs.google.com/java/lang/String.html) lexicalXSDDecimal)

Converts the string argument into a BigDecimal value.

**Parameters:**lexicalXSDDecimal - A string containing lexical representation of xsd:decimal. **Returns:**A BigDecimal value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - lexicalXSDDecimal is not a valid string representation of [BigDecimal](http://docs.google.com/java/math/BigDecimal.html).

### parseFloat

float **parseFloat**([String](http://docs.google.com/java/lang/String.html) lexicalXSDFloat)

Converts the string argument into a float value.

**Parameters:**lexicalXSDFloat - A string containing lexical representation of xsd:float. **Returns:**A float value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - lexicalXSDFloat is not a valid string representation of a float value.

### parseDouble

double **parseDouble**([String](http://docs.google.com/java/lang/String.html) lexicalXSDDouble)

Converts the string argument into a double value.

**Parameters:**lexicalXSDDouble - A string containing lexical representation of xsd:double. **Returns:**A double value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - lexicalXSDDouble is not a valid string representation of a double value.

### parseBoolean

boolean **parseBoolean**([String](http://docs.google.com/java/lang/String.html) lexicalXSDBoolean)

Converts the string argument into a boolean value.

**Parameters:**lexicalXSDBoolean - A string containing lexical representation of xsd:boolean. **Returns:**A boolean value represented by the string argument. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:boolean.

### parseByte

byte **parseByte**([String](http://docs.google.com/java/lang/String.html) lexicalXSDByte)

Converts the string argument into a byte value.

**Parameters:**lexicalXSDByte - A string containing lexical representation of xsd:byte. **Returns:**A byte value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - lexicalXSDByte does not contain a parseable byte. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:byte.

### parseQName

[QName](http://docs.google.com/javax/xml/namespace/QName.html) **parseQName**([String](http://docs.google.com/java/lang/String.html) lexicalXSDQName,  
 [NamespaceContext](http://docs.google.com/javax/xml/namespace/NamespaceContext.html) nsc)

Converts the string argument into a QName value.

String parameter lexicalXSDQname must conform to lexical value space specifed at [XML Schema Part 2:Datatypes specification:QNames](http://www.w3.org/TR/xmlschema-2/#QName)

**Parameters:**lexicalXSDQName - A string containing lexical representation of xsd:QName.nsc - A namespace context for interpreting a prefix within a QName. **Returns:**A QName value represented by the string argument. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if string parameter does not conform to XML Schema Part 2 specification or if namespace prefix of lexicalXSDQname is not bound to a URI in NamespaceContext nsc.

### parseDateTime

[Calendar](http://docs.google.com/java/util/Calendar.html) **parseDateTime**([String](http://docs.google.com/java/lang/String.html) lexicalXSDDateTime)

Converts the string argument into a Calendar value.

**Parameters:**lexicalXSDDateTime - A string containing lexical representation of xsd:datetime. **Returns:**A Calendar object represented by the string argument. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:dateTime.

### parseBase64Binary

byte[] **parseBase64Binary**([String](http://docs.google.com/java/lang/String.html) lexicalXSDBase64Binary)

Converts the string argument into an array of bytes.

**Parameters:**lexicalXSDBase64Binary - A string containing lexical representation of xsd:base64Binary. **Returns:**An array of bytes represented by the string argument. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:base64Binary

### parseHexBinary

byte[] **parseHexBinary**([String](http://docs.google.com/java/lang/String.html) lexicalXSDHexBinary)

Converts the string argument into an array of bytes.

**Parameters:**lexicalXSDHexBinary - A string containing lexical representation of xsd:hexBinary. **Returns:**An array of bytes represented by the string argument. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:hexBinary.

### parseUnsignedInt

long **parseUnsignedInt**([String](http://docs.google.com/java/lang/String.html) lexicalXSDUnsignedInt)

Converts the string argument into a long value.

**Parameters:**lexicalXSDUnsignedInt - A string containing lexical representation of xsd:unsignedInt. **Returns:**A long value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - if string parameter can not be parsed into a long value.

### parseUnsignedShort

int **parseUnsignedShort**([String](http://docs.google.com/java/lang/String.html) lexicalXSDUnsignedShort)

Converts the string argument into an int value.

**Parameters:**lexicalXSDUnsignedShort - A string containing lexical representation of xsd:unsignedShort. **Returns:**An int value represented by the string argument. **Throws:** [NumberFormatException](http://docs.google.com/java/lang/NumberFormatException.html) - if string parameter can not be parsed into an int value.

### parseTime

[Calendar](http://docs.google.com/java/util/Calendar.html) **parseTime**([String](http://docs.google.com/java/lang/String.html) lexicalXSDTime)

Converts the string argument into a Calendar value.

**Parameters:**lexicalXSDTime - A string containing lexical representation of xsd:Time. **Returns:**A Calendar value represented by the string argument. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:Time.

### parseDate

[Calendar](http://docs.google.com/java/util/Calendar.html) **parseDate**([String](http://docs.google.com/java/lang/String.html) lexicalXSDDate)

Converts the string argument into a Calendar value.

**Parameters:**lexicalXSDDate - A string containing lexical representation of xsd:Date. **Returns:**A Calendar value represented by the string argument. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:Date.

### parseAnySimpleType

[String](http://docs.google.com/java/lang/String.html) **parseAnySimpleType**([String](http://docs.google.com/java/lang/String.html) lexicalXSDAnySimpleType)

Return a string containing the lexical representation of the simple type.

**Parameters:**lexicalXSDAnySimpleType - A string containing lexical representation of the simple type. **Returns:**A string containing the lexical representation of the simple type.

### printString

[String](http://docs.google.com/java/lang/String.html) **printString**([String](http://docs.google.com/java/lang/String.html) val)

Converts the string argument into a string.

**Parameters:**val - A string value. **Returns:**A string containing a lexical representation of xsd:string

### printInteger

[String](http://docs.google.com/java/lang/String.html) **printInteger**([BigInteger](http://docs.google.com/java/math/BigInteger.html) val)

Converts a BigInteger value into a string.

**Parameters:**val - A BigInteger value **Returns:**A string containing a lexical representation of xsd:integer **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - val is null.

### printInt

[String](http://docs.google.com/java/lang/String.html) **printInt**(int val)

Converts an int value into a string.

**Parameters:**val - An int value **Returns:**A string containing a lexical representation of xsd:int

### printLong

[String](http://docs.google.com/java/lang/String.html) **printLong**(long val)

Converts a long value into a string.

**Parameters:**val - A long value **Returns:**A string containing a lexical representation of xsd:long

### printShort

[String](http://docs.google.com/java/lang/String.html) **printShort**(short val)

Converts a short value into a string.

**Parameters:**val - A short value **Returns:**A string containing a lexical representation of xsd:short

### printDecimal

[String](http://docs.google.com/java/lang/String.html) **printDecimal**([BigDecimal](http://docs.google.com/java/math/BigDecimal.html) val)

Converts a BigDecimal value into a string.

**Parameters:**val - A BigDecimal value **Returns:**A string containing a lexical representation of xsd:decimal **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - val is null.

### printFloat

[String](http://docs.google.com/java/lang/String.html) **printFloat**(float val)

Converts a float value into a string.

**Parameters:**val - A float value **Returns:**A string containing a lexical representation of xsd:float

### printDouble

[String](http://docs.google.com/java/lang/String.html) **printDouble**(double val)

Converts a double value into a string.

**Parameters:**val - A double value **Returns:**A string containing a lexical representation of xsd:double

### printBoolean

[String](http://docs.google.com/java/lang/String.html) **printBoolean**(boolean val)

Converts a boolean value into a string.

**Parameters:**val - A boolean value **Returns:**A string containing a lexical representation of xsd:boolean

### printByte

[String](http://docs.google.com/java/lang/String.html) **printByte**(byte val)

Converts a byte value into a string.

**Parameters:**val - A byte value **Returns:**A string containing a lexical representation of xsd:byte

### printQName

[String](http://docs.google.com/java/lang/String.html) **printQName**([QName](http://docs.google.com/javax/xml/namespace/QName.html) val,  
 [NamespaceContext](http://docs.google.com/javax/xml/namespace/NamespaceContext.html) nsc)

Converts a QName instance into a string.

**Parameters:**val - A QName valuensc - A namespace context for interpreting a prefix within a QName. **Returns:**A string containing a lexical representation of QName **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if val is null or if nsc is non-null or nsc.getPrefix(nsprefixFromVal) is null.

### printDateTime

[String](http://docs.google.com/java/lang/String.html) **printDateTime**([Calendar](http://docs.google.com/java/util/Calendar.html) val)

Converts a Calendar value into a string.

**Parameters:**val - A Calendar value **Returns:**A string containing a lexical representation of xsd:dateTime **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if val is null.

### printBase64Binary

[String](http://docs.google.com/java/lang/String.html) **printBase64Binary**(byte[] val)

Converts an array of bytes into a string.

**Parameters:**val - an array of bytes **Returns:**A string containing a lexical representation of xsd:base64Binary **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if val is null.

### printHexBinary

[String](http://docs.google.com/java/lang/String.html) **printHexBinary**(byte[] val)

Converts an array of bytes into a string.

**Parameters:**val - an array of bytes **Returns:**A string containing a lexical representation of xsd:hexBinary **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if val is null.

### printUnsignedInt

[String](http://docs.google.com/java/lang/String.html) **printUnsignedInt**(long val)

Converts a long value into a string.

**Parameters:**val - A long value **Returns:**A string containing a lexical representation of xsd:unsignedInt

### printUnsignedShort

[String](http://docs.google.com/java/lang/String.html) **printUnsignedShort**(int val)

Converts an int value into a string.

**Parameters:**val - An int value **Returns:**A string containing a lexical representation of xsd:unsignedShort

### printTime

[String](http://docs.google.com/java/lang/String.html) **printTime**([Calendar](http://docs.google.com/java/util/Calendar.html) val)

Converts a Calendar value into a string.

**Parameters:**val - A Calendar value **Returns:**A string containing a lexical representation of xsd:time **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if val is null.

### printDate

[String](http://docs.google.com/java/lang/String.html) **printDate**([Calendar](http://docs.google.com/java/util/Calendar.html) val)

Converts a Calendar value into a string.

**Parameters:**val - A Calendar value **Returns:**A string containing a lexical representation of xsd:date **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if val is null.

### printAnySimpleType

[String](http://docs.google.com/java/lang/String.html) **printAnySimpleType**([String](http://docs.google.com/java/lang/String.html) val)

Converts a string value into a string.

**Parameters:**val - A string value **Returns:**A string containing a lexical representation of xsd:AnySimpleType

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DatatypeConverterInterface.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/xml/bind/DatatypeConverter.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/bind/Element.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/bind/DatatypeConverterInterface.html)    [**NO FRAMES**](http://docs.google.com/DatatypeConverterInterface.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).