| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EnumSyntax.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/print/attribute/DocAttributeSet.html)   [**NEXT CLASS**](http://docs.google.com/javax/print/attribute/HashAttributeSet.html) | [**FRAMES**](http://docs.google.com/index.html?javax/print/attribute/EnumSyntax.html)    [**NO FRAMES**](http://docs.google.com/EnumSyntax.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **javax.print.attribute**

Class EnumSyntax

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.print.attribute.EnumSyntax**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [Chromaticity](http://docs.google.com/javax/print/attribute/standard/Chromaticity.html), [ColorSupported](http://docs.google.com/javax/print/attribute/standard/ColorSupported.html), [Compression](http://docs.google.com/javax/print/attribute/standard/Compression.html), [Fidelity](http://docs.google.com/javax/print/attribute/standard/Fidelity.html), [Finishings](http://docs.google.com/javax/print/attribute/standard/Finishings.html), [JobSheets](http://docs.google.com/javax/print/attribute/standard/JobSheets.html), [JobState](http://docs.google.com/javax/print/attribute/standard/JobState.html), [JobStateReason](http://docs.google.com/javax/print/attribute/standard/JobStateReason.html), [Media](http://docs.google.com/javax/print/attribute/standard/Media.html), [MultipleDocumentHandling](http://docs.google.com/javax/print/attribute/standard/MultipleDocumentHandling.html), [OrientationRequested](http://docs.google.com/javax/print/attribute/standard/OrientationRequested.html), [PDLOverrideSupported](http://docs.google.com/javax/print/attribute/standard/PDLOverrideSupported.html), [PresentationDirection](http://docs.google.com/javax/print/attribute/standard/PresentationDirection.html), [PrinterIsAcceptingJobs](http://docs.google.com/javax/print/attribute/standard/PrinterIsAcceptingJobs.html), [PrinterState](http://docs.google.com/javax/print/attribute/standard/PrinterState.html), [PrinterStateReason](http://docs.google.com/javax/print/attribute/standard/PrinterStateReason.html), [PrintQuality](http://docs.google.com/javax/print/attribute/standard/PrintQuality.html), [ReferenceUriSchemesSupported](http://docs.google.com/javax/print/attribute/standard/ReferenceUriSchemesSupported.html), [Severity](http://docs.google.com/javax/print/attribute/standard/Severity.html), [SheetCollate](http://docs.google.com/javax/print/attribute/standard/SheetCollate.html), [Sides](http://docs.google.com/javax/print/attribute/standard/Sides.html)

public abstract class **EnumSyntax**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

Class EnumSyntax is an abstract base class providing the common implementation of all "type safe enumeration" objects. An enumeration class (which extends class EnumSyntax) provides a group of enumeration values (objects) that are singleton instances of the enumeration class; for example:

public class Bach extends EnumSyntax {  
 public static final Bach JOHANN\_SEBASTIAN = new Bach(0);  
 public static final Bach WILHELM\_FRIEDEMANN = new Bach(1);  
 public static final Bach CARL\_PHILIP\_EMMANUEL = new Bach(2);  
 public static final Bach JOHANN\_CHRISTIAN = new Bach(3);  
 public static final Bach P\_D\_Q = new Bach(4);  
  
 private static final String[] stringTable = {  
 "Johann Sebastian Bach",  
 "Wilhelm Friedemann Bach",  
 "Carl Philip Emmanuel Bach",  
 "Johann Christian Bach",  
 "P.D.Q. Bach"  
 };  
  
 protected String[] getStringTable() {  
 return stringTable;  
 }  
  
 private static final Bach[] enumValueTable = {  
 JOHANN\_SEBASTIAN,  
 WILHELM\_FRIEDEMANN,  
 CARL\_PHILIP\_EMMANUEL,  
 JOHANN\_CHRISTIAN,  
 P\_D\_Q  
 };  
  
 protected EnumSyntax[] getEnumValueTable() {  
 return enumValueTable;  
 }  
 }

You can then write code that uses the == and != operators to test enumeration values; for example:

Bach theComposer;  
 . . .  
 if (theComposer == Bach.JOHANN\_SEBASTIAN) {  
 System.out.println ("The greatest composer of all time!");  
 }

The equals() method for an enumeration class just does a test for identical objects (==).

You can convert an enumeration value to a string by calling [toString()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#toString()). The string is obtained from a table supplied by the enumeration class.

Under the hood, an enumeration value is just an integer, a different integer for each enumeration value within an enumeration class. You can get an enumeration value's integer value by calling [getValue()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getValue()). An enumeration value's integer value is established when it is constructed (see [EnumSyntax(int)](http://docs.google.com/javax/print/attribute/EnumSyntax.html#EnumSyntax(int))). Since the constructor is protected, the only possible enumeration values are the singleton objects declared in the enumeration class; additional enumeration values cannot be created at run time.

You can define a subclass of an enumeration class that extends it with additional enumeration values. The subclass's enumeration values' integer values need not be distinct from the superclass's enumeration values' integer values; the ==, !=, equals(), and toString() methods will still work properly even if the subclass uses some of the same integer values as the superclass. However, the application in which the enumeration class and subclass are used may need to have distinct integer values in the superclass and subclass.

**See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#javax.print.attribute.EnumSyntax)

| **Constructor Summary** | |
| --- | --- |
| protected | [**EnumSyntax**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#EnumSyntax(int))(int value)            Construct a new enumeration value with the given integer value. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#clone())()            Returns a clone of this enumeration value, which to preserve the semantics of enumeration values is the same object as this enumeration value. |
| protected  [EnumSyntax](http://docs.google.com/javax/print/attribute/EnumSyntax.html)[] | [**getEnumValueTable**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getEnumValueTable())()            Returns the enumeration value table for this enumeration value's enumeration class. |
| protected  int | [**getOffset**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getOffset())()            Returns the lowest integer value used by this enumeration value's enumeration class. |
| protected  [String](http://docs.google.com/java/lang/String.html)[] | [**getStringTable**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getStringTable())()            Returns the string table for this enumeration value's enumeration class. |
| int | [**getValue**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getValue())()            Returns this enumeration value's integer value. |
| int | [**hashCode**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#hashCode())()            Returns a hash code value for this enumeration value. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**readResolve**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#readResolve())()            During object input, convert this deserialized enumeration instance to the proper enumeration value defined in the enumeration attribute class. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/print/attribute/EnumSyntax.html#toString())()            Returns a string value corresponding to this enumeration value. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### EnumSyntax

protected **EnumSyntax**(int value)

Construct a new enumeration value with the given integer value.

**Parameters:**value - Integer value.

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

### getValue

public int **getValue**()

Returns this enumeration value's integer value.

**Returns:**the value

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()

Returns a clone of this enumeration value, which to preserve the semantics of enumeration values is the same object as this enumeration value.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of this instance.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### hashCode

public int **hashCode**()

Returns a hash code value for this enumeration value. The hash code is just this enumeration value's integer value.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### toString

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

Returns a string value corresponding to this enumeration value.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of the object.

### readResolve

protected [Object](http://docs.google.com/java/lang/Object.html) **readResolve**()  
 throws [ObjectStreamException](http://docs.google.com/java/io/ObjectStreamException.html)

During object input, convert this deserialized enumeration instance to the proper enumeration value defined in the enumeration attribute class.

**Returns:**The enumeration singleton value stored at index *i*-*L* in the enumeration value table returned by [getEnumValueTable()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getEnumValueTable()), where *i* is this enumeration value's integer value and *L* is the value returned by [getOffset()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getOffset()). **Throws:** [ObjectStreamException](http://docs.google.com/java/io/ObjectStreamException.html) - if the stream can't be deserialised [InvalidObjectException](http://docs.google.com/java/io/InvalidObjectException.html) - Thrown if the enumeration value table is null, this enumeration value's integer value does not correspond to an element in the enumeration value table, or the corresponding element in the enumeration value table is null. (Note: [InvalidObjectException](http://docs.google.com/java/io/InvalidObjectException.html) is a subclass of [ObjectStreamException](http://docs.google.com/java/io/ObjectStreamException.html), which readResolve() is declared to throw.)

### getStringTable

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

Returns the string table for this enumeration value's enumeration class. The enumeration class's integer values are assumed to lie in the range *L*..*L*+*N*-1, where *L* is the value returned by [getOffset()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getOffset()) and *N* is the length of the string table. The element in the string table at index *i*-*L* is the value returned by [toString()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#toString()) for the enumeration value whose integer value is *i*. If an integer within the above range is not used by any enumeration value, leave the corresponding table element null.

The default implementation returns null. If the enumeration class (a subclass of class EnumSyntax) does not override this method to return a non-null string table, and the subclass does not override the [toString()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#toString()) method, the base class [toString()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#toString()) method will return just a string representation of this enumeration value's integer value.

**Returns:**the string table

### getEnumValueTable

protected [EnumSyntax](http://docs.google.com/javax/print/attribute/EnumSyntax.html)[] **getEnumValueTable**()

Returns the enumeration value table for this enumeration value's enumeration class. The enumeration class's integer values are assumed to lie in the range *L*..*L*+*N*-1, where *L* is the value returned by [getOffset()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#getOffset()) and *N* is the length of the enumeration value table. The element in the enumeration value table at index *i*-*L* is the enumeration value object whose integer value is *i*; the [readResolve()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#readResolve()) method needs this to preserve singleton semantics during deserialization of an enumeration instance. If an integer within the above range is not used by any enumeration value, leave the corresponding table element null.

The default implementation returns null. If the enumeration class (a subclass of class EnumSyntax) does not override this method to return a non-null enumeration value table, and the subclass does not override the [readResolve()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#readResolve()) method, the base class [readResolve()](http://docs.google.com/javax/print/attribute/EnumSyntax.html#readResolve()) method will throw an exception whenever an enumeration instance is deserialized from an object input stream.

**Returns:**the value table

### getOffset

protected int **getOffset**()

Returns the lowest integer value used by this enumeration value's enumeration class.

The default implementation returns 0. If the enumeration class (a subclass of class EnumSyntax) uses integer values starting at other than 0, override this method in the subclass.

**Returns:**the offset of the lowest enumeration value.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EnumSyntax.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/print/attribute/DocAttributeSet.html)   [**NEXT CLASS**](http://docs.google.com/javax/print/attribute/HashAttributeSet.html) | [**FRAMES**](http://docs.google.com/index.html?javax/print/attribute/EnumSyntax.html)    [**NO FRAMES**](http://docs.google.com/EnumSyntax.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).