| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AWTKeyStroke.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/java/awt/AWTException.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/AWTPermission.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/AWTKeyStroke.html)    [**NO FRAMES**](http://docs.google.com/AWTKeyStroke.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **java.awt**

Class AWTKeyStroke

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html) **Direct Known Subclasses:** [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html)

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

An AWTKeyStroke represents a key action on the keyboard, or equivalent input device. AWTKeyStrokes can correspond to only a press or release of a particular key, just as KEY\_PRESSED and KEY\_RELEASED KeyEvents do; alternately, they can correspond to typing a specific Java character, just as KEY\_TYPED KeyEvents do. In all cases, AWTKeyStrokes can specify modifiers (alt, shift, control, meta, altGraph, or a combination thereof) which must be present during the action for an exact match.

AWTKeyStrokes are immutable, and are intended to be unique. Client code should never create an AWTKeyStroke on its own, but should instead use a variant of getAWTKeyStroke. Client use of these factory methods allows the AWTKeyStroke implementation to cache and share instances efficiently.

**Since:** 1.4 **See Also:**[getAWTKeyStroke(char)](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(char)), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.AWTKeyStroke)

| **Constructor Summary** | |
| --- | --- |
| protected | [**AWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html#AWTKeyStroke())()            Constructs an AWTKeyStroke with default values. |
| protected | [**AWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html#AWTKeyStroke(char,%20int,%20int,%20boolean))(char keyChar, int keyCode, int modifiers, boolean onKeyRelease)            Constructs an AWTKeyStroke with the specified values. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/awt/AWTKeyStroke.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) anObject)            Returns true if this object is identical to the specified object. |
| static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) | [**getAWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(char))(char keyChar)            Returns a shared instance of an AWTKeyStroke that represents a KEY\_TYPED event for the specified character. |
| static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) | [**getAWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(java.lang.Character,%20int))([Character](http://docs.google.com/java/lang/Character.html) keyChar, int modifiers)            Returns a shared instance of an AWTKeyStroke that represents a KEY\_TYPED event for the specified Character object and a set of modifiers. |
| static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) | [**getAWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(int,%20int))(int keyCode, int modifiers)            Returns a shared instance of an AWTKeyStroke, given a numeric key code and a set of modifiers. |
| static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) | [**getAWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(int,%20int,%20boolean))(int keyCode, int modifiers, boolean onKeyRelease)            Returns a shared instance of an AWTKeyStroke, given a numeric key code and a set of modifiers, specifying whether the key is activated when it is pressed or released. |
| static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) | [**getAWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Parses a string and returns an AWTKeyStroke. |
| static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) | [**getAWTKeyStrokeForEvent**](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStrokeForEvent(java.awt.event.KeyEvent))([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) anEvent)            Returns an AWTKeyStroke which represents the stroke which generated a given KeyEvent. |
| char | [**getKeyChar**](http://docs.google.com/java/awt/AWTKeyStroke.html#getKeyChar())()            Returns the character for this AWTKeyStroke. |
| int | [**getKeyCode**](http://docs.google.com/java/awt/AWTKeyStroke.html#getKeyCode())()            Returns the numeric key code for this AWTKeyStroke. |
| int | [**getKeyEventType**](http://docs.google.com/java/awt/AWTKeyStroke.html#getKeyEventType())()            Returns the type of KeyEvent which corresponds to this AWTKeyStroke. |
| int | [**getModifiers**](http://docs.google.com/java/awt/AWTKeyStroke.html#getModifiers())()            Returns the modifier keys for this AWTKeyStroke. |
| int | [**hashCode**](http://docs.google.com/java/awt/AWTKeyStroke.html#hashCode())()            Returns a numeric value for this object that is likely to be unique, making it a good choice as the index value in a hash table. |
| boolean | [**isOnKeyRelease**](http://docs.google.com/java/awt/AWTKeyStroke.html#isOnKeyRelease())()            Returns whether this AWTKeyStroke represents a key release. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**readResolve**](http://docs.google.com/java/awt/AWTKeyStroke.html#readResolve())()            Returns a cached instance of AWTKeyStroke (or a subclass of AWTKeyStroke) which is equal to this instance. |
| protected static void | [**registerSubclass**](http://docs.google.com/java/awt/AWTKeyStroke.html#registerSubclass(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<?> subclass)            Registers a new class which the factory methods in AWTKeyStroke will use when generating new instances of AWTKeyStrokes. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/awt/AWTKeyStroke.html#toString())()            Returns a string that displays and identifies this object's properties. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [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** |
| --- |

### AWTKeyStroke

protected **AWTKeyStroke**()

Constructs an AWTKeyStroke with default values. The default values used are:

| Property | Default Value |
| --- | --- |
| Key Char | KeyEvent.CHAR\_UNDEFINED |
| Key Code | KeyEvent.VK\_UNDEFINED |
| Modifiers | none |
| On key release? | false |

AWTKeyStrokes should not be constructed by client code. Use a variant of getAWTKeyStroke instead.

**See Also:**[getAWTKeyStroke(char)](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(char))

### AWTKeyStroke

protected **AWTKeyStroke**(char keyChar,  
 int keyCode,  
 int modifiers,  
 boolean onKeyRelease)

Constructs an AWTKeyStroke with the specified values. AWTKeyStrokes should not be constructed by client code. Use a variant of getAWTKeyStroke instead.

**Parameters:**keyChar - the character value for a keyboard keykeyCode - the key code for this AWTKeyStrokemodifiers - a bitwise-ored combination of any modifiersonKeyRelease - true if this AWTKeyStroke corresponds to a key release; false otherwise**See Also:**[getAWTKeyStroke(char)](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(char))

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

### registerSubclass

protected static void **registerSubclass**([Class](http://docs.google.com/java/lang/Class.html)<?> subclass)

Registers a new class which the factory methods in AWTKeyStroke will use when generating new instances of AWTKeyStrokes. After invoking this method, the factory methods will return instances of the specified Class. The specified Class must be either AWTKeyStroke or derived from AWTKeyStroke, and it must have a no-arg constructor. The constructor can be of any accessibility, including private. This operation flushes the current AWTKeyStroke cache.

**Parameters:**subclass - the new Class of which the factory methods should create instances **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if subclass is null, or if subclass does not have a no-arg constructor [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if subclass is not AWTKeyStroke, or a class derived from AWTKeyStroke

### getAWTKeyStroke

public static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) **getAWTKeyStroke**(char keyChar)

Returns a shared instance of an AWTKeyStroke that represents a KEY\_TYPED event for the specified character.

**Parameters:**keyChar - the character value for a keyboard key **Returns:**an AWTKeyStroke object for that key

### getAWTKeyStroke

public static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) **getAWTKeyStroke**([Character](http://docs.google.com/java/lang/Character.html) keyChar,  
 int modifiers)

Returns a shared instance of an AWTKeyStroke that represents a KEY\_TYPED event for the specified Character object and a set of modifiers. Note that the first parameter is of type Character rather than char. This is to avoid inadvertent clashes with calls to getAWTKeyStroke(int keyCode, int modifiers). The modifiers consist of any combination of following:

* java.awt.event.InputEvent.SHIFT\_DOWN\_MASK
* java.awt.event.InputEvent.CTRL\_DOWN\_MASK
* java.awt.event.InputEvent.META\_DOWN\_MASK
* java.awt.event.InputEvent.ALT\_DOWN\_MASK
* java.awt.event.InputEvent.ALT\_GRAPH\_DOWN\_MASK

The old modifiers listed below also can be used, but they are mapped to \_DOWN\_ modifiers.

* java.awt.event.InputEvent.SHIFT\_MASK
* java.awt.event.InputEvent.CTRL\_MASK
* java.awt.event.InputEvent.META\_MASK
* java.awt.event.InputEvent.ALT\_MASK
* java.awt.event.InputEvent.ALT\_GRAPH\_MASK

also can be used, but they are mapped to \_DOWN\_ modifiers. Since these numbers are all different powers of two, any combination of them is an integer in which each bit represents a different modifier key. Use 0 to specify no modifiers.

**Parameters:**keyChar - the Character object for a keyboard charactermodifiers - a bitwise-ored combination of any modifiers **Returns:**an AWTKeyStroke object for that key **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if keyChar is null**See Also:**[InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)

### getAWTKeyStroke

public static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) **getAWTKeyStroke**(int keyCode,  
 int modifiers,  
 boolean onKeyRelease)

Returns a shared instance of an AWTKeyStroke, given a numeric key code and a set of modifiers, specifying whether the key is activated when it is pressed or released.

The "virtual key" constants defined in java.awt.event.KeyEvent can be used to specify the key code. For example:

* java.awt.event.KeyEvent.VK\_ENTER
* java.awt.event.KeyEvent.VK\_TAB
* java.awt.event.KeyEvent.VK\_SPACE

The modifiers consist of any combination of:

* java.awt.event.InputEvent.SHIFT\_DOWN\_MASK
* java.awt.event.InputEvent.CTRL\_DOWN\_MASK
* java.awt.event.InputEvent.META\_DOWN\_MASK
* java.awt.event.InputEvent.ALT\_DOWN\_MASK
* java.awt.event.InputEvent.ALT\_GRAPH\_DOWN\_MASK

The old modifiers

* java.awt.event.InputEvent.SHIFT\_MASK
* java.awt.event.InputEvent.CTRL\_MASK
* java.awt.event.InputEvent.META\_MASK
* java.awt.event.InputEvent.ALT\_MASK
* java.awt.event.InputEvent.ALT\_GRAPH\_MASK

also can be used, but they are mapped to \_DOWN\_ modifiers. Since these numbers are all different powers of two, any combination of them is an integer in which each bit represents a different modifier key. Use 0 to specify no modifiers.

**Parameters:**keyCode - an int specifying the numeric code for a keyboard keymodifiers - a bitwise-ored combination of any modifiersonKeyRelease - true if the AWTKeyStroke should represent a key release; false otherwise **Returns:**an AWTKeyStroke object for that key**See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html), [InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)

### getAWTKeyStroke

public static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) **getAWTKeyStroke**(int keyCode,  
 int modifiers)

Returns a shared instance of an AWTKeyStroke, given a numeric key code and a set of modifiers. The returned AWTKeyStroke will correspond to a key press.

The "virtual key" constants defined in java.awt.event.KeyEvent can be used to specify the key code. For example:

* java.awt.event.KeyEvent.VK\_ENTER
* java.awt.event.KeyEvent.VK\_TAB
* java.awt.event.KeyEvent.VK\_SPACE

The modifiers consist of any combination of:

* java.awt.event.InputEvent.SHIFT\_DOWN\_MASK
* java.awt.event.InputEvent.CTRL\_DOWN\_MASK
* java.awt.event.InputEvent.META\_DOWN\_MASK
* java.awt.event.InputEvent.ALT\_DOWN\_MASK
* java.awt.event.InputEvent.ALT\_GRAPH\_DOWN\_MASK

The old modifiers

* java.awt.event.InputEvent.SHIFT\_MASK
* java.awt.event.InputEvent.CTRL\_MASK
* java.awt.event.InputEvent.META\_MASK
* java.awt.event.InputEvent.ALT\_MASK
* java.awt.event.InputEvent.ALT\_GRAPH\_MASK

also can be used, but they are mapped to \_DOWN\_ modifiers. Since these numbers are all different powers of two, any combination of them is an integer in which each bit represents a different modifier key. Use 0 to specify no modifiers.

**Parameters:**keyCode - an int specifying the numeric code for a keyboard keymodifiers - a bitwise-ored combination of any modifiers **Returns:**an AWTKeyStroke object for that key**See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html), [InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)

### getAWTKeyStrokeForEvent

public static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) **getAWTKeyStrokeForEvent**([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) anEvent)

Returns an AWTKeyStroke which represents the stroke which generated a given KeyEvent.

This method obtains the keyChar from a KeyTyped event, and the keyCode from a KeyPressed or KeyReleased event. The KeyEvent modifiers are obtained for all three types of KeyEvent.

**Parameters:**anEvent - the KeyEvent from which to obtain the AWTKeyStroke **Returns:**the AWTKeyStroke that precipitated the event **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if anEvent is null

### getAWTKeyStroke

public static [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html) **getAWTKeyStroke**([String](http://docs.google.com/java/lang/String.html) s)

Parses a string and returns an AWTKeyStroke. The string must have the following syntax:

<modifiers>\* (<typedID> | <pressedReleasedID>)  
  
 modifiers := shift | control | ctrl | meta | alt | altGraph   
 typedID := typed <typedKey>  
 typedKey := string of length 1 giving Unicode character.  
 pressedReleasedID := (pressed | released) key  
 key := KeyEvent key code name, i.e. the name following "VK\_".

If typed, pressed or released is not specified, pressed is assumed. Here are some examples:

"INSERT" => getAWTKeyStroke(KeyEvent.VK\_INSERT, 0);  
 "control DELETE" => getAWTKeyStroke(KeyEvent.VK\_DELETE, InputEvent.CTRL\_MASK);  
 "alt shift X" => getAWTKeyStroke(KeyEvent.VK\_X, InputEvent.ALT\_MASK | InputEvent.SHIFT\_MASK);  
 "alt shift released X" => getAWTKeyStroke(KeyEvent.VK\_X, InputEvent.ALT\_MASK | InputEvent.SHIFT\_MASK, true);  
 "typed a" => getAWTKeyStroke('a');

**Parameters:**s - a String formatted as described above **Returns:**an AWTKeyStroke object for that String **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if s is null, or is formatted incorrectly

### getKeyChar

public final char **getKeyChar**()

Returns the character for this AWTKeyStroke.

**Returns:**a char value**See Also:**[getAWTKeyStroke(char)](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(char)), [KeyEvent.getKeyChar()](http://docs.google.com/java/awt/event/KeyEvent.html#getKeyChar())

### getKeyCode

public final int **getKeyCode**()

Returns the numeric key code for this AWTKeyStroke.

**Returns:**an int containing the key code value**See Also:**[getAWTKeyStroke(int,int)](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(int,%20int)), [KeyEvent.getKeyCode()](http://docs.google.com/java/awt/event/KeyEvent.html#getKeyCode())

### getModifiers

public final int **getModifiers**()

Returns the modifier keys for this AWTKeyStroke.

**Returns:**an int containing the modifiers**See Also:**[getAWTKeyStroke(int,int)](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(int,%20int))

### isOnKeyRelease

public final boolean **isOnKeyRelease**()

Returns whether this AWTKeyStroke represents a key release.

**Returns:**true if this AWTKeyStroke represents a key release; false otherwise**See Also:**[getAWTKeyStroke(int,int,boolean)](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(int,%20int,%20boolean))

### getKeyEventType

public final int **getKeyEventType**()

Returns the type of KeyEvent which corresponds to this AWTKeyStroke.

**Returns:**KeyEvent.KEY\_PRESSED, KeyEvent.KEY\_TYPED, or KeyEvent.KEY\_RELEASED**See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html)

### hashCode

public int **hashCode**()

Returns a numeric value for this object that is likely to be unique, making it a good choice as the index value in a hash table.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**an int that represents 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)

### equals

public final boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) anObject)

Returns true if this object is identical to the specified object.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**anObject - the Object to compare this object to **Returns:**true if the objects are identical**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### toString

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

Returns a string that displays and identifies this object's properties. The String returned by this method can be passed as a parameter to getAWTKeyStroke(String) to produce a key stroke equal to this key stroke.

**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 this object**See Also:**[getAWTKeyStroke(String)](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(java.lang.String))

### readResolve

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

Returns a cached instance of AWTKeyStroke (or a subclass of AWTKeyStroke) which is equal to this instance.

**Returns:**a cached instance which is equal to this instance **Throws:** [ObjectStreamException](http://docs.google.com/java/io/ObjectStreamException.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AWTKeyStroke.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/java/awt/AWTException.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/AWTPermission.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/AWTKeyStroke.html)    [**NO FRAMES**](http://docs.google.com/AWTKeyStroke.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[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).