| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/KeyStroke.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/swing/JWindow.AccessibleJWindow.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/LayoutFocusTraversalPolicy.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/KeyStroke.html)    [**NO FRAMES**](http://docs.google.com/KeyStroke.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#3dy6vkm) |

## **javax.swing**

Class KeyStroke

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html)

public class **KeyStroke**extends [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)

A KeyStroke represents a key action on the keyboard, or equivalent input device. KeyStrokes 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, KeyStrokes can specify modifiers (alt, shift, control, meta, altGraph, or a combination thereof) which must be present during the action for an exact match.

KeyStrokes are used to define high-level (semantic) action events. Instead of trapping every keystroke and throwing away the ones you are not interested in, those keystrokes you care about automatically initiate actions on the Components with which they are registered.

KeyStrokes are immutable, and are intended to be unique. Client code cannot create a KeyStroke; a variant of getKeyStroke must be used instead. These factory methods allow the KeyStroke implementation to cache and share instances efficiently.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

**See Also:**[Keymap](http://docs.google.com/javax/swing/text/Keymap.html), [getKeyStroke(char)](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStroke(char))

| **Method Summary** | |
| --- | --- |
| static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**getKeyStroke**](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStroke(char))(char keyChar)            Returns a shared instance of a KeyStroke that represents a KEY\_TYPED event for the specified character. |
| static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**getKeyStroke**](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStroke(java.lang.Character,%20int))([Character](http://docs.google.com/java/lang/Character.html) keyChar, int modifiers)            Returns a shared instance of a KeyStroke that represents a KEY\_TYPED event for the specified Character object and a set of modifiers. |
| static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**getKeyStroke**](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStroke(char,%20boolean))(char keyChar, boolean onKeyRelease)  **Deprecated.** *use getKeyStroke(char)* |
| static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**getKeyStroke**](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStroke(int,%20int))(int keyCode, int modifiers)            Returns a shared instance of a KeyStroke, given a numeric key code and a set of modifiers. |
| static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**getKeyStroke**](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStroke(int,%20int,%20boolean))(int keyCode, int modifiers, boolean onKeyRelease)            Returns a shared instance of a KeyStroke, given a numeric key code and a set of modifiers, specifying whether the key is activated when it is pressed or released. |
| static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**getKeyStroke**](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStroke(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Parses a string and returns a KeyStroke. |
| static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**getKeyStrokeForEvent**](http://docs.google.com/javax/swing/KeyStroke.html#getKeyStrokeForEvent(java.awt.event.KeyEvent))([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) anEvent)            Returns a KeyStroke which represents the stroke which generated a given KeyEvent. |

| **Methods inherited from class java.awt.**[**AWTKeyStroke**](http://docs.google.com/java/awt/AWTKeyStroke.html) |
| --- |
| [equals](http://docs.google.com/java/awt/AWTKeyStroke.html#equals(java.lang.Object)), [getAWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(char)), [getAWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(java.lang.Character,%20int)), [getAWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(int,%20int)), [getAWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(int,%20int,%20boolean)), [getAWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStroke(java.lang.String)), [getAWTKeyStrokeForEvent](http://docs.google.com/java/awt/AWTKeyStroke.html#getAWTKeyStrokeForEvent(java.awt.event.KeyEvent)), [getKeyChar](http://docs.google.com/java/awt/AWTKeyStroke.html#getKeyChar()), [getKeyCode](http://docs.google.com/java/awt/AWTKeyStroke.html#getKeyCode()), [getKeyEventType](http://docs.google.com/java/awt/AWTKeyStroke.html#getKeyEventType()), [getModifiers](http://docs.google.com/java/awt/AWTKeyStroke.html#getModifiers()), [hashCode](http://docs.google.com/java/awt/AWTKeyStroke.html#hashCode()), [isOnKeyRelease](http://docs.google.com/java/awt/AWTKeyStroke.html#isOnKeyRelease()), [readResolve](http://docs.google.com/java/awt/AWTKeyStroke.html#readResolve()), [registerSubclass](http://docs.google.com/java/awt/AWTKeyStroke.html#registerSubclass(java.lang.Class)), [toString](http://docs.google.com/java/awt/AWTKeyStroke.html#toString()) |

| **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)) |

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

### getKeyStroke

public static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **getKeyStroke**(char keyChar)

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

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

### getKeyStroke

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **getKeyStroke**(char keyChar,  
 boolean onKeyRelease)

**Deprecated.** *use getKeyStroke(char)*

Returns an instance of a KeyStroke, specifying whether the key is considered to be activated when it is pressed or released. Unlike all other factory methods in this class, the instances returned by this method are not necessarily cached or shared.

**Parameters:**keyChar - the character value for a keyboard keyonKeyRelease - true if this KeyStroke corresponds to a key release; false otherwise. **Returns:**a KeyStroke object for that key

### getKeyStroke

public static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **getKeyStroke**([Character](http://docs.google.com/java/lang/Character.html) keyChar,  
 int modifiers)

Returns a shared instance of a KeyStroke 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 getKeyStroke(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 KeyStroke object for that key **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if keyChar is null**Since:** 1.3 **See Also:**[InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)

### getKeyStroke

public static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **getKeyStroke**(int keyCode,  
 int modifiers,  
 boolean onKeyRelease)

Returns a shared instance of a KeyStroke, 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 KeyStroke should represent a key release; false otherwise. **Returns:**a KeyStroke 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)

### getKeyStroke

public static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **getKeyStroke**(int keyCode,  
 int modifiers)

Returns a shared instance of a KeyStroke, given a numeric key code and a set of modifiers. The returned KeyStroke 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:**a KeyStroke 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)

### getKeyStrokeForEvent

public static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **getKeyStrokeForEvent**([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) anEvent)

Returns a KeyStroke 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 KeyStroke **Returns:**the KeyStroke that precipitated the event **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if anEvent is null

### getKeyStroke

public static [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **getKeyStroke**([String](http://docs.google.com/java/lang/String.html) s)

Parses a string and returns a KeyStroke. 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" => getKeyStroke(KeyEvent.VK\_INSERT, 0);  
 "control DELETE" => getKeyStroke(KeyEvent.VK\_DELETE, InputEvent.CTRL\_MASK);  
 "alt shift X" => getKeyStroke(KeyEvent.VK\_X, InputEvent.ALT\_MASK | InputEvent.SHIFT\_MASK);  
 "alt shift released X" => getKeyStroke(KeyEvent.VK\_X, InputEvent.ALT\_MASK | InputEvent.SHIFT\_MASK, true);  
 "typed a" => getKeyStroke('a');

In order to maintain backward-compatibility, specifying a null String, or a String which is formatted incorrectly, returns null.

**Parameters:**s - a String formatted as described above **Returns:**a KeyStroke object for that String, or null if the specified String is null, or is formatted incorrectly

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/KeyStroke.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/swing/JWindow.AccessibleJWindow.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/LayoutFocusTraversalPolicy.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/KeyStroke.html)    [**NO FRAMES**](http://docs.google.com/KeyStroke.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#3dy6vkm) |

[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).