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

## **java.awt**

Interface KeyEventPostProcessor

**All Known Implementing Classes:** [DefaultFocusManager](http://docs.google.com/javax/swing/DefaultFocusManager.html), [DefaultKeyboardFocusManager](http://docs.google.com/java/awt/DefaultKeyboardFocusManager.html), [FocusManager](http://docs.google.com/javax/swing/FocusManager.html), [KeyboardFocusManager](http://docs.google.com/java/awt/KeyboardFocusManager.html)

public interface **KeyEventPostProcessor**

A KeyEventPostProcessor cooperates with the current KeyboardFocusManager in the final resolution of all unconsumed KeyEvents. KeyEventPostProcessors registered with the current KeyboardFocusManager will receive KeyEvents after the KeyEvents have been dispatched to and handled by their targets. KeyEvents that would have been otherwise discarded because no Component in the application currently owns the focus will also be forwarded to registered KeyEventPostProcessors. This will allow applications to implement features that require global KeyEvent post-handling, such as menu shortcuts.

Note that the KeyboardFocusManager itself implements KeyEventPostProcessor. By default, the current KeyboardFocusManager will be the final KeyEventPostProcessor in the chain. The current KeyboardFocusManager cannot be completely deregistered as a KeyEventPostProcessor. However, if a KeyEventPostProcessor reports that no further post-processing of the KeyEvent should take place, the AWT will consider the event fully handled and will take no additional action with regard to the event. (While it is possible for client code to register the current KeyboardFocusManager as a KeyEventPostProcessor one or more times, this is usually unnecessary and not recommended.)

**Since:** 1.4 **See Also:**[KeyboardFocusManager.addKeyEventPostProcessor(java.awt.KeyEventPostProcessor)](http://docs.google.com/java/awt/KeyboardFocusManager.html#addKeyEventPostProcessor(java.awt.KeyEventPostProcessor)), [KeyboardFocusManager.removeKeyEventPostProcessor(java.awt.KeyEventPostProcessor)](http://docs.google.com/java/awt/KeyboardFocusManager.html#removeKeyEventPostProcessor(java.awt.KeyEventPostProcessor))

| **Method Summary** | |
| --- | --- |
| boolean | [**postProcessKeyEvent**](http://docs.google.com/java/awt/KeyEventPostProcessor.html#postProcessKeyEvent(java.awt.event.KeyEvent))([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) e)            This method is called by the current KeyboardFocusManager, requesting that this KeyEventPostProcessor perform any necessary post-processing which should be part of the KeyEvent's final resolution. |

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

### postProcessKeyEvent

boolean **postProcessKeyEvent**([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) e)

This method is called by the current KeyboardFocusManager, requesting that this KeyEventPostProcessor perform any necessary post-processing which should be part of the KeyEvent's final resolution. At the time this method is invoked, typically the KeyEvent has already been dispatched to and handled by its target. However, if no Component in the application currently owns the focus, then the KeyEvent has not been dispatched to any Component. Typically, KeyEvent post-processing will be used to implement features which require global KeyEvent post-handling, such as menu shortcuts. Note that if a KeyEventPostProcessor wishes to dispatch the KeyEvent, it must use redispatchEvent to prevent the AWT from recursively requesting that this KeyEventPostProcessor perform post-processing of the event again.

If an implementation of this method returns false, then the KeyEvent is passed to the next KeyEventPostProcessor in the chain, ending with the current KeyboardFocusManager. If an implementation returns true, the KeyEvent is assumed to have been fully handled (although this need not be the case), and the AWT will take no further action with regard to the KeyEvent. If an implementation consumes the KeyEvent but returns false, the consumed event will still be passed to the next KeyEventPostProcessor in the chain. It is important for developers to check whether the KeyEvent has been consumed before performing any post-processing of the KeyEvent. By default, the current KeyboardFocusManager will perform no post- processing in response to a consumed KeyEvent.

**Parameters:**e - the KeyEvent to post-process **Returns:**true if the AWT should take no further action with regard to the KeyEvent; false otherwise**See Also:**[KeyboardFocusManager.redispatchEvent(java.awt.Component, java.awt.AWTEvent)](http://docs.google.com/java/awt/KeyboardFocusManager.html#redispatchEvent(java.awt.Component,%20java.awt.AWTEvent))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/KeyEventPostProcessor.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/KeyEventDispatcher.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Label.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/KeyEventPostProcessor.html)    [**NO FRAMES**](http://docs.google.com/KeyEventPostProcessor.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).