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

## **java.awt.event**

Interface WindowListener

**All Superinterfaces:** [EventListener](http://docs.google.com/java/util/EventListener.html) **All Known Implementing Classes:** [AWTEventMulticaster](http://docs.google.com/java/awt/AWTEventMulticaster.html), [BasicToolBarUI.FrameListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.FrameListener.html), [JMenu.WinListener](http://docs.google.com/javax/swing/JMenu.WinListener.html), [WindowAdapter](http://docs.google.com/java/awt/event/WindowAdapter.html)

public interface **WindowListener**extends [EventListener](http://docs.google.com/java/util/EventListener.html)

The listener interface for receiving window events. The class that is interested in processing a window event either implements this interface (and all the methods it contains) or extends the abstract WindowAdapter class (overriding only the methods of interest). The listener object created from that class is then registered with a Window using the window's addWindowListener method. When the window's status changes by virtue of being opened, closed, activated or deactivated, iconified or deiconified, the relevant method in the listener object is invoked, and the WindowEvent is passed to it.

**Since:** 1.1 **See Also:**[WindowAdapter](http://docs.google.com/java/awt/event/WindowAdapter.html), [WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html), [Tutorial: How to Write Window Listeners](http://java.sun.com/docs/books/tutorial/uiswing/events/windowlistener.html)

| **Method Summary** | |
| --- | --- |
| void | [**windowActivated**](http://docs.google.com/java/awt/event/WindowListener.html#windowActivated(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Invoked when the Window is set to be the active Window. |
| void | [**windowClosed**](http://docs.google.com/java/awt/event/WindowListener.html#windowClosed(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Invoked when a window has been closed as the result of calling dispose on the window. |
| void | [**windowClosing**](http://docs.google.com/java/awt/event/WindowListener.html#windowClosing(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Invoked when the user attempts to close the window from the window's system menu. |
| void | [**windowDeactivated**](http://docs.google.com/java/awt/event/WindowListener.html#windowDeactivated(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Invoked when a Window is no longer the active Window. |
| void | [**windowDeiconified**](http://docs.google.com/java/awt/event/WindowListener.html#windowDeiconified(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Invoked when a window is changed from a minimized to a normal state. |
| void | [**windowIconified**](http://docs.google.com/java/awt/event/WindowListener.html#windowIconified(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Invoked when a window is changed from a normal to a minimized state. |
| void | [**windowOpened**](http://docs.google.com/java/awt/event/WindowListener.html#windowOpened(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Invoked the first time a window is made visible. |

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

### windowOpened

void **windowOpened**([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)

Invoked the first time a window is made visible.

### windowClosing

void **windowClosing**([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)

Invoked when the user attempts to close the window from the window's system menu.

### windowClosed

void **windowClosed**([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)

Invoked when a window has been closed as the result of calling dispose on the window.

### windowIconified

void **windowIconified**([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)

Invoked when a window is changed from a normal to a minimized state. For many platforms, a minimized window is displayed as the icon specified in the window's iconImage property.

**See Also:**[Frame.setIconImage(java.awt.Image)](http://docs.google.com/java/awt/Frame.html#setIconImage(java.awt.Image))

### windowDeiconified

void **windowDeiconified**([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)

Invoked when a window is changed from a minimized to a normal state.

### windowActivated

void **windowActivated**([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)

Invoked when the Window is set to be the active Window. Only a Frame or a Dialog can be the active Window. The native windowing system may denote the active Window or its children with special decorations, such as a highlighted title bar. The active Window is always either the focused Window, or the first Frame or Dialog that is an owner of the focused Window.

### windowDeactivated

void **windowDeactivated**([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)

Invoked when a Window is no longer the active Window. Only a Frame or a Dialog can be the active Window. The native windowing system may denote the active Window or its children with special decorations, such as a highlighted title bar. The active Window is always either the focused Window, or the first Frame or Dialog that is an owner of the focused Window.

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