| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AWTPermission.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/AWTKeyStroke.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/BasicStroke.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/AWTPermission.html)    [**NO FRAMES**](http://docs.google.com/AWTPermission.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#4d34og8) | METHOD |

## **java.awt**

Class AWTPermission

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.security.Permission](http://docs.google.com/java/security/Permission.html)  
 [java.security.BasicPermission](http://docs.google.com/java/security/BasicPermission.html)  
 **java.awt.AWTPermission**

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

public final class **AWTPermission**extends [BasicPermission](http://docs.google.com/java/security/BasicPermission.html)

This class is for AWT permissions. An AWTPermission contains a target name but no actions list; you either have the named permission or you don't.

The target name is the name of the AWT permission (see below). The naming convention follows the hierarchical property naming convention. Also, an asterisk could be used to represent all AWT permissions.

The following table lists all the possible AWTPermission target names, and for each provides a description of what the permission allows and a discussion of the risks of granting code the permission.

| Permission Target Name | What the Permission Allows | Risks of Allowing this Permission |
| --- | --- | --- |
| accessClipboard | Posting and retrieval of information to and from the AWT clipboard | This would allow malfeasant code to share potentially sensitive or confidential information. |
| accessEventQueue | Access to the AWT event queue | After retrieving the AWT event queue, malicious code may peek at and even remove existing events from its event queue, as well as post bogus events which may purposefully cause the application or applet to misbehave in an insecure manner. |
| accessSystemTray | Access to the AWT SystemTray instance | This would allow malicious code to add tray icons to the system tray. First, such an icon may look like the icon of some known application (such as a firewall or anti-virus) and order a user to do something unsafe (with help of balloon messages). Second, the system tray may be glutted with tray icons so that no one could add a tray icon anymore. |
| createRobot | Create java.awt.Robot objects | The java.awt.Robot object allows code to generate native-level mouse and keyboard events as well as read the screen. It could allow malicious code to control the system, run other programs, read the display, and deny mouse and keyboard access to the user. |
| fullScreenExclusive | Enter full-screen exclusive mode | Entering full-screen exclusive mode allows direct access to low-level graphics card memory. This could be used to spoof the system, since the program is in direct control of rendering. |
| listenToAllAWTEvents | Listen to all AWT events, system-wide | After adding an AWT event listener, malicious code may scan all AWT events dispatched in the system, allowing it to read all user input (such as passwords). Each AWT event listener is called from within the context of that event queue's EventDispatchThread, so if the accessEventQueue permission is also enabled, malicious code could modify the contents of AWT event queues system-wide, causing the application or applet to misbehave in an insecure manner. |
| readDisplayPixels | Readback of pixels from the display screen | Interfaces such as the java.awt.Composite interface or the java.awt.Robot class allow arbitrary code to examine pixels on the display enable malicious code to snoop on the activities of the user. |
| replaceKeyboardFocusManager | Sets the KeyboardFocusManager for a particular thread. | When SecurityManager is installed, the invoking thread must be granted this permission in order to replace the current KeyboardFocusManager. If permission is not granted, a SecurityException will be thrown. |
| setAppletStub | Setting the stub which implements Applet container services | Malicious code could set an applet's stub and result in unexpected behavior or denial of service to an applet. |
| setWindowAlwaysOnTop | Setting always-on-top property of the window: [Window.setAlwaysOnTop(boolean)](http://docs.google.com/java/awt/Window.html#setAlwaysOnTop(boolean)) | The malicious window might make itself look and behave like a real full desktop, so that information entered by the unsuspecting user is captured and subsequently misused |
| showWindowWithoutWarningBanner | Display of a window without also displaying a banner warning that the window was created by an applet | Without this warning, an applet may pop up windows without the user knowing that they belong to an applet. Since users may make security-sensitive decisions based on whether or not the window belongs to an applet (entering a username and password into a dialog box, for example), disabling this warning banner may allow applets to trick the user into entering such information. |
| toolkitModality | Creating [TOOLKIT\_MODAL](http://docs.google.com/java/awt/Dialog.ModalityType.html#TOOLKIT_MODAL) dialogs and setting the [TOOLKIT\_EXCLUDE](http://docs.google.com/java/awt/Dialog.ModalExclusionType.html#TOOLKIT_EXCLUDE) window property. | When a toolkit-modal dialog is shown from an applet, it blocks all other applets in the browser. When launching applications from Java Web Start, its windows (such as the security dialog) may also be blocked by toolkit-modal dialogs, shown from these applications. |
| watchMousePointer | Getting the information about the mouse pointer position at any time | Constantly watching the mouse pointer, an applet can make guesses about what the user is doing, i.e. moving the mouse to the lower left corner of the screen most likely means that the user is about to launch an application. If a virtual keypad is used so that keyboard is emulated using the mouse, an applet may guess what is being typed. |

**See Also:**[BasicPermission](http://docs.google.com/java/security/BasicPermission.html), [Permission](http://docs.google.com/java/security/Permission.html), [Permissions](http://docs.google.com/java/security/Permissions.html), [PermissionCollection](http://docs.google.com/java/security/PermissionCollection.html), [SecurityManager](http://docs.google.com/java/lang/SecurityManager.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.AWTPermission)

| **Constructor Summary** | |
| --- | --- |
| [**AWTPermission**](http://docs.google.com/java/awt/AWTPermission.html#AWTPermission(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Creates a new AWTPermission with the specified name. |
| [**AWTPermission**](http://docs.google.com/java/awt/AWTPermission.html#AWTPermission(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) name, [String](http://docs.google.com/java/lang/String.html) actions)            Creates a new AWTPermission object with the specified name. |

| **Method Summary** | |
| --- | --- |

| **Methods inherited from class java.security.**[**BasicPermission**](http://docs.google.com/java/security/BasicPermission.html) |
| --- |
| [equals](http://docs.google.com/java/security/BasicPermission.html#equals(java.lang.Object)), [getActions](http://docs.google.com/java/security/BasicPermission.html#getActions()), [hashCode](http://docs.google.com/java/security/BasicPermission.html#hashCode()), [implies](http://docs.google.com/java/security/BasicPermission.html#implies(java.security.Permission)), [newPermissionCollection](http://docs.google.com/java/security/BasicPermission.html#newPermissionCollection()) |

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

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

### AWTPermission

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

Creates a new AWTPermission with the specified name. The name is the symbolic name of the AWTPermission, such as "topLevelWindow", "systemClipboard", etc. An asterisk may be used to indicate all AWT permissions.

**Parameters:**name - the name of the AWTPermission **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if name is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if name is empty.

### AWTPermission

public **AWTPermission**([String](http://docs.google.com/java/lang/String.html) name,  
 [String](http://docs.google.com/java/lang/String.html) actions)

Creates a new AWTPermission object with the specified name. The name is the symbolic name of the AWTPermission, and the actions string is currently unused and should be null.

**Parameters:**name - the name of the AWTPermissionactions - should be null **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if name is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if name is empty.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AWTPermission.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/AWTKeyStroke.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/BasicStroke.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/AWTPermission.html)    [**NO FRAMES**](http://docs.google.com/AWTPermission.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#4d34og8) | METHOD |

[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).