| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TrayIcon.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/Transparency.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/TrayIcon.MessageType.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/TrayIcon.html)    [**NO FRAMES**](http://docs.google.com/TrayIcon.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#3rdcrjn) |

## **java.awt**

Class TrayIcon

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

public class **TrayIcon**extends [Object](http://docs.google.com/java/lang/Object.html)

A TrayIcon object represents a tray icon that can be added to the [system tray](http://docs.google.com/java/awt/SystemTray.html). A TrayIcon can have a tooltip (text), an image, a popup menu, and a set of listeners associated with it.

A TrayIcon can generate various [MouseEvents](http://docs.google.com/java/awt/event/MouseEvent.html) and supports adding corresponding listeners to receive notification of these events. TrayIcon processes some of the events by itself. For example, by default, when the right-mouse click is performed on the TrayIcon it displays the specified popup menu. When the mouse hovers over the TrayIcon the tooltip is displayed.

**Note:** When the MouseEvent is dispatched to its registered listeners its component property will be set to null. (See [ComponentEvent.getComponent()](http://docs.google.com/java/awt/event/ComponentEvent.html#getComponent())) The source property will be set to this TrayIcon. (See [EventObject.getSource()](http://docs.google.com/java/util/EventObject.html#getSource()))

**Note:** A well-behaved [TrayIcon](http://docs.google.com/java/awt/TrayIcon.html) implementation will assign different gestures to showing a popup menu and selecting a tray icon.

A TrayIcon can generate an [ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html). On some platforms, this occurs when the user selects the tray icon using either the mouse or keyboard.

If a SecurityManager is installed, the AWTPermission accessSystemTray must be granted in order to create a TrayIcon. Otherwise the constructor will throw a SecurityException.

See the [SystemTray](http://docs.google.com/java/awt/SystemTray.html) class overview for an example on how to use the TrayIcon API.

**Since:** 1.6 **See Also:**[SystemTray.add(java.awt.TrayIcon)](http://docs.google.com/java/awt/SystemTray.html#add(java.awt.TrayIcon)), [ComponentEvent.getComponent()](http://docs.google.com/java/awt/event/ComponentEvent.html#getComponent()), [EventObject.getSource()](http://docs.google.com/java/util/EventObject.html#getSource())

| **Nested Class Summary** | |
| --- | --- |
| static class | [**TrayIcon.MessageType**](http://docs.google.com/java/awt/TrayIcon.MessageType.html)            The message type determines which icon will be displayed in the caption of the message, and a possible system sound a message may generate upon showing. |

| **Constructor Summary** | |
| --- | --- |
| [**TrayIcon**](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image))([Image](http://docs.google.com/java/awt/Image.html) image)            Creates a TrayIcon with the specified image. |
| [**TrayIcon**](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image,%20java.lang.String))([Image](http://docs.google.com/java/awt/Image.html) image, [String](http://docs.google.com/java/lang/String.html) tooltip)            Creates a TrayIcon with the specified image and tooltip text. |
| [**TrayIcon**](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image,%20java.lang.String,%20java.awt.PopupMenu))([Image](http://docs.google.com/java/awt/Image.html) image, [String](http://docs.google.com/java/lang/String.html) tooltip, [PopupMenu](http://docs.google.com/java/awt/PopupMenu.html) popup)            Creates a TrayIcon with the specified image, tooltip and popup menu. |

| **Method Summary** | |
| --- | --- |
| void | [**addActionListener**](http://docs.google.com/java/awt/TrayIcon.html#addActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) listener)            Adds the specified action listener to receive ActionEvents from this TrayIcon. |
| void | [**addMouseListener**](http://docs.google.com/java/awt/TrayIcon.html#addMouseListener(java.awt.event.MouseListener))([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) listener)            Adds the specified mouse listener to receive mouse events from this TrayIcon. |
| void | [**addMouseMotionListener**](http://docs.google.com/java/awt/TrayIcon.html#addMouseMotionListener(java.awt.event.MouseMotionListener))([MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html) listener)            Adds the specified mouse listener to receive mouse-motion events from this TrayIcon. |
| void | [**displayMessage**](http://docs.google.com/java/awt/TrayIcon.html#displayMessage(java.lang.String,%20java.lang.String,%20java.awt.TrayIcon.MessageType))([String](http://docs.google.com/java/lang/String.html) caption, [String](http://docs.google.com/java/lang/String.html) text, [TrayIcon.MessageType](http://docs.google.com/java/awt/TrayIcon.MessageType.html) messageType)            Displays a popup message near the tray icon. |
| [String](http://docs.google.com/java/lang/String.html) | [**getActionCommand**](http://docs.google.com/java/awt/TrayIcon.html#getActionCommand())()            Returns the command name of the action event fired by this tray icon. |
| [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)[] | [**getActionListeners**](http://docs.google.com/java/awt/TrayIcon.html#getActionListeners())()            Returns an array of all the action listeners registered on this TrayIcon. |
| [Image](http://docs.google.com/java/awt/Image.html) | [**getImage**](http://docs.google.com/java/awt/TrayIcon.html#getImage())()            Returns the current image used for this TrayIcon. |
| [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html)[] | [**getMouseListeners**](http://docs.google.com/java/awt/TrayIcon.html#getMouseListeners())()            Returns an array of all the mouse listeners registered on this TrayIcon. |
| [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)[] | [**getMouseMotionListeners**](http://docs.google.com/java/awt/TrayIcon.html#getMouseMotionListeners())()            Returns an array of all the mouse-motion listeners registered on this TrayIcon. |
| [PopupMenu](http://docs.google.com/java/awt/PopupMenu.html) | [**getPopupMenu**](http://docs.google.com/java/awt/TrayIcon.html#getPopupMenu())()            Returns the popup menu associated with this TrayIcon. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getSize**](http://docs.google.com/java/awt/TrayIcon.html#getSize())()            Returns the size, in pixels, of the space that the tray icon occupies in the system tray. |
| [String](http://docs.google.com/java/lang/String.html) | [**getToolTip**](http://docs.google.com/java/awt/TrayIcon.html#getToolTip())()            Returns the tooltip string associated with this TrayIcon. |
| boolean | [**isImageAutoSize**](http://docs.google.com/java/awt/TrayIcon.html#isImageAutoSize())()            Returns the value of the auto-size property. |
| void | [**removeActionListener**](http://docs.google.com/java/awt/TrayIcon.html#removeActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) listener)            Removes the specified action listener. |
| void | [**removeMouseListener**](http://docs.google.com/java/awt/TrayIcon.html#removeMouseListener(java.awt.event.MouseListener))([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) listener)            Removes the specified mouse listener. |
| void | [**removeMouseMotionListener**](http://docs.google.com/java/awt/TrayIcon.html#removeMouseMotionListener(java.awt.event.MouseMotionListener))([MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html) listener)            Removes the specified mouse-motion listener. |
| void | [**setActionCommand**](http://docs.google.com/java/awt/TrayIcon.html#setActionCommand(java.lang.String))([String](http://docs.google.com/java/lang/String.html) command)            Sets the command name for the action event fired by this tray icon. |
| void | [**setImage**](http://docs.google.com/java/awt/TrayIcon.html#setImage(java.awt.Image))([Image](http://docs.google.com/java/awt/Image.html) image)            Sets the image for this TrayIcon. |
| void | [**setImageAutoSize**](http://docs.google.com/java/awt/TrayIcon.html#setImageAutoSize(boolean))(boolean autosize)            Sets the auto-size property. |
| void | [**setPopupMenu**](http://docs.google.com/java/awt/TrayIcon.html#setPopupMenu(java.awt.PopupMenu))([PopupMenu](http://docs.google.com/java/awt/PopupMenu.html) popup)            Sets the popup menu for this TrayIcon. |
| void | [**setToolTip**](http://docs.google.com/java/awt/TrayIcon.html#setToolTip(java.lang.String))([String](http://docs.google.com/java/lang/String.html) tooltip)            Sets the tooltip string for this TrayIcon. |

| **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()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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** |
| --- |

### TrayIcon

public **TrayIcon**([Image](http://docs.google.com/java/awt/Image.html) image)

Creates a TrayIcon with the specified image.

**Parameters:**image - the Image to be used **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if image is null [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the system tray isn't supported by the current platform [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if accessSystemTray permission is not granted**See Also:**[SystemTray.add(TrayIcon)](http://docs.google.com/java/awt/SystemTray.html#add(java.awt.TrayIcon)), [TrayIcon(Image, String, PopupMenu)](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image,%20java.lang.String,%20java.awt.PopupMenu)), [TrayIcon(Image, String)](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image,%20java.lang.String)), [SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [AWTPermission](http://docs.google.com/java/awt/AWTPermission.html)

### TrayIcon

public **TrayIcon**([Image](http://docs.google.com/java/awt/Image.html) image,  
 [String](http://docs.google.com/java/lang/String.html) tooltip)

Creates a TrayIcon with the specified image and tooltip text.

**Parameters:**image - the Image to be usedtooltip - the string to be used as tooltip text; if the value is null no tooltip is shown **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if image is null [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the system tray isn't supported by the current platform [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if accessSystemTray permission is not granted**See Also:**[SystemTray.add(TrayIcon)](http://docs.google.com/java/awt/SystemTray.html#add(java.awt.TrayIcon)), [TrayIcon(Image)](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image)), [TrayIcon(Image, String, PopupMenu)](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image,%20java.lang.String,%20java.awt.PopupMenu)), [SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [AWTPermission](http://docs.google.com/java/awt/AWTPermission.html)

### TrayIcon

public **TrayIcon**([Image](http://docs.google.com/java/awt/Image.html) image,  
 [String](http://docs.google.com/java/lang/String.html) tooltip,  
 [PopupMenu](http://docs.google.com/java/awt/PopupMenu.html) popup)

Creates a TrayIcon with the specified image, tooltip and popup menu.

**Parameters:**image - the Image to be usedtooltip - the string to be used as tooltip text; if the value is null no tooltip is shownpopup - the menu to be used for the tray icon's popup menu; if the value is null no popup menu is shown **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if image is null [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the system tray isn't supported by the current platform [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if accessSystemTray permission is not granted**See Also:**[SystemTray.add(TrayIcon)](http://docs.google.com/java/awt/SystemTray.html#add(java.awt.TrayIcon)), [TrayIcon(Image, String)](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image,%20java.lang.String)), [TrayIcon(Image)](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image)), [PopupMenu](http://docs.google.com/java/awt/PopupMenu.html), [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html), [addMouseListener(MouseListener)](http://docs.google.com/java/awt/TrayIcon.html#addMouseListener(java.awt.event.MouseListener)), [SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [AWTPermission](http://docs.google.com/java/awt/AWTPermission.html)

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

### setImage

public void **setImage**([Image](http://docs.google.com/java/awt/Image.html) image)

Sets the image for this TrayIcon. The previous tray icon image is discarded without calling the [Image.flush()](http://docs.google.com/java/awt/Image.html#flush()) method — you will need to call it manually.

If the image represents an animated image, it will be animated automatically.

See the [setImageAutoSize(boolean)](http://docs.google.com/java/awt/TrayIcon.html#setImageAutoSize(boolean)) property for details on the size of the displayed image.

Calling this method with the same image that is currently being used has no effect.

**Parameters:**image - the non-null Image to be used **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if image is null**See Also:**[getImage()](http://docs.google.com/java/awt/TrayIcon.html#getImage()), [Image](http://docs.google.com/java/awt/Image.html), [SystemTray.add(TrayIcon)](http://docs.google.com/java/awt/SystemTray.html#add(java.awt.TrayIcon)), [TrayIcon(Image, String)](http://docs.google.com/java/awt/TrayIcon.html#TrayIcon(java.awt.Image,%20java.lang.String))

### getImage

public [Image](http://docs.google.com/java/awt/Image.html) **getImage**()

Returns the current image used for this TrayIcon.

**Returns:**the image**See Also:**[setImage(Image)](http://docs.google.com/java/awt/TrayIcon.html#setImage(java.awt.Image)), [Image](http://docs.google.com/java/awt/Image.html)

### setPopupMenu

public void **setPopupMenu**([PopupMenu](http://docs.google.com/java/awt/PopupMenu.html) popup)

Sets the popup menu for this TrayIcon. If popup is null, no popup menu will be associated with this TrayIcon.

Note that this popup must not be added to any parent before or after it is set on the tray icon. If you add it to some parent, the popup may be removed from that parent.

The popup can be set on one TrayIcon only. Setting the same popup on multiple TrayIcons will cause an IllegalArgumentException.

**Note:** Some platforms may not support showing the user-specified popup menu component when the user right-clicks the tray icon. In this situation, either no menu will be displayed or, on some systems, a native version of the menu may be displayed.

**Parameters:**popup - a PopupMenu or null to remove any popup menu **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the popup is already set for another TrayIcon**See Also:**[getPopupMenu()](http://docs.google.com/java/awt/TrayIcon.html#getPopupMenu())

### getPopupMenu

public [PopupMenu](http://docs.google.com/java/awt/PopupMenu.html) **getPopupMenu**()

Returns the popup menu associated with this TrayIcon.

**Returns:**the popup menu or null if none exists**See Also:**[setPopupMenu(PopupMenu)](http://docs.google.com/java/awt/TrayIcon.html#setPopupMenu(java.awt.PopupMenu))

### setToolTip

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

Sets the tooltip string for this TrayIcon. The tooltip is displayed automatically when the mouse hovers over the icon. Setting the tooltip to null removes any tooltip text. When displayed, the tooltip string may be truncated on some platforms; the number of characters that may be displayed is platform-dependent.

**Parameters:**tooltip - the string for the tooltip; if the value is null no tooltip is shown**See Also:**[getToolTip()](http://docs.google.com/java/awt/TrayIcon.html#getToolTip())

### getToolTip

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

Returns the tooltip string associated with this TrayIcon.

**Returns:**the tooltip string or null if none exists**See Also:**[setToolTip(String)](http://docs.google.com/java/awt/TrayIcon.html#setToolTip(java.lang.String))

### setImageAutoSize

public void **setImageAutoSize**(boolean autosize)

Sets the auto-size property. Auto-size determines whether the tray image is automatically sized to fit the space allocated for the image on the tray. By default, the auto-size property is set to false.

If auto-size is false, and the image size doesn't match the tray icon space, the image is painted as-is inside that space — if larger than the allocated space, it will be cropped.

If auto-size is true, the image is stretched or shrunk to fit the tray icon space.

**Parameters:**autosize - true to auto-size the image, false otherwise**See Also:**[isImageAutoSize()](http://docs.google.com/java/awt/TrayIcon.html#isImageAutoSize())

### isImageAutoSize

public boolean **isImageAutoSize**()

Returns the value of the auto-size property.

**Returns:**true if the image will be auto-sized, false otherwise**See Also:**[setImageAutoSize(boolean)](http://docs.google.com/java/awt/TrayIcon.html#setImageAutoSize(boolean))

### addMouseListener

public void **addMouseListener**([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) listener)

Adds the specified mouse listener to receive mouse events from this TrayIcon. Calling this method with a null value has no effect.

**Note**: The MouseEvent's coordinates (received from the TrayIcon) are relative to the screen, not the TrayIcon.

**Note:** The MOUSE\_ENTERED and MOUSE\_EXITED mouse events are not supported.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**listener - the mouse listener**See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html), [removeMouseListener(MouseListener)](http://docs.google.com/java/awt/TrayIcon.html#removeMouseListener(java.awt.event.MouseListener)), [getMouseListeners()](http://docs.google.com/java/awt/TrayIcon.html#getMouseListeners())

### removeMouseListener

public void **removeMouseListener**([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) listener)

Removes the specified mouse listener. Calling this method with null or an invalid value has no effect.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**listener - the mouse listener**See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html), [addMouseListener(MouseListener)](http://docs.google.com/java/awt/TrayIcon.html#addMouseListener(java.awt.event.MouseListener)), [getMouseListeners()](http://docs.google.com/java/awt/TrayIcon.html#getMouseListeners())

### getMouseListeners

public [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html)[] **getMouseListeners**()

Returns an array of all the mouse listeners registered on this TrayIcon.

**Returns:**all of the MouseListeners registered on this TrayIcon or an empty array if no mouse listeners are currently registered**See Also:**[addMouseListener(MouseListener)](http://docs.google.com/java/awt/TrayIcon.html#addMouseListener(java.awt.event.MouseListener)), [removeMouseListener(MouseListener)](http://docs.google.com/java/awt/TrayIcon.html#removeMouseListener(java.awt.event.MouseListener)), [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html)

### addMouseMotionListener

public void **addMouseMotionListener**([MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html) listener)

Adds the specified mouse listener to receive mouse-motion events from this TrayIcon. Calling this method with a null value has no effect.

**Note**: The MouseEvent's coordinates (received from the TrayIcon) are relative to the screen, not the TrayIcon.

**Note:** The MOUSE\_DRAGGED mouse event is not supported.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**listener - the mouse listener**See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html), [removeMouseMotionListener(MouseMotionListener)](http://docs.google.com/java/awt/TrayIcon.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [getMouseMotionListeners()](http://docs.google.com/java/awt/TrayIcon.html#getMouseMotionListeners())

### removeMouseMotionListener

public void **removeMouseMotionListener**([MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html) listener)

Removes the specified mouse-motion listener. Calling this method with null or an invalid value has no effect.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**listener - the mouse listener**See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html), [addMouseMotionListener(MouseMotionListener)](http://docs.google.com/java/awt/TrayIcon.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [getMouseMotionListeners()](http://docs.google.com/java/awt/TrayIcon.html#getMouseMotionListeners())

### getMouseMotionListeners

public [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)[] **getMouseMotionListeners**()

Returns an array of all the mouse-motion listeners registered on this TrayIcon.

**Returns:**all of the MouseInputListeners registered on this TrayIcon or an empty array if no mouse listeners are currently registered**See Also:**[addMouseMotionListener(MouseMotionListener)](http://docs.google.com/java/awt/TrayIcon.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseMotionListener(MouseMotionListener)](http://docs.google.com/java/awt/TrayIcon.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)

### getActionCommand

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

Returns the command name of the action event fired by this tray icon.

**Returns:**the action command name, or null if none exists**See Also:**[addActionListener(ActionListener)](http://docs.google.com/java/awt/TrayIcon.html#addActionListener(java.awt.event.ActionListener)), [setActionCommand(String)](http://docs.google.com/java/awt/TrayIcon.html#setActionCommand(java.lang.String))

### setActionCommand

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

Sets the command name for the action event fired by this tray icon. By default, this action command is set to null.

**Parameters:**command - a string used to set the tray icon's action command.**See Also:**[ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html), [addActionListener(ActionListener)](http://docs.google.com/java/awt/TrayIcon.html#addActionListener(java.awt.event.ActionListener)), [getActionCommand()](http://docs.google.com/java/awt/TrayIcon.html#getActionCommand())

### addActionListener

public void **addActionListener**([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) listener)

Adds the specified action listener to receive ActionEvents from this TrayIcon. Action events usually occur when a user selects the tray icon, using either the mouse or keyboard. The conditions in which action events are generated are platform-dependent.

Calling this method with a null value has no effect.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**listener - the action listener**See Also:**[removeActionListener(java.awt.event.ActionListener)](http://docs.google.com/java/awt/TrayIcon.html#removeActionListener(java.awt.event.ActionListener)), [getActionListeners()](http://docs.google.com/java/awt/TrayIcon.html#getActionListeners()), [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html), [setActionCommand(String)](http://docs.google.com/java/awt/TrayIcon.html#setActionCommand(java.lang.String))

### removeActionListener

public void **removeActionListener**([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) listener)

Removes the specified action listener. Calling this method with null or an invalid value has no effect.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**listener - the action listener**See Also:**[ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html), [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html), [addActionListener(ActionListener)](http://docs.google.com/java/awt/TrayIcon.html#addActionListener(java.awt.event.ActionListener)), [getActionListeners()](http://docs.google.com/java/awt/TrayIcon.html#getActionListeners()), [setActionCommand(String)](http://docs.google.com/java/awt/TrayIcon.html#setActionCommand(java.lang.String))

### getActionListeners

public [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)[] **getActionListeners**()

Returns an array of all the action listeners registered on this TrayIcon.

**Returns:**all of the ActionListeners registered on this TrayIcon or an empty array if no action listeners are currently registered**See Also:**[addActionListener(ActionListener)](http://docs.google.com/java/awt/TrayIcon.html#addActionListener(java.awt.event.ActionListener)), [removeActionListener(ActionListener)](http://docs.google.com/java/awt/TrayIcon.html#removeActionListener(java.awt.event.ActionListener)), [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)

### displayMessage

public void **displayMessage**([String](http://docs.google.com/java/lang/String.html) caption,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [TrayIcon.MessageType](http://docs.google.com/java/awt/TrayIcon.MessageType.html) messageType)

Displays a popup message near the tray icon. The message will disappear after a time or if the user clicks on it. Clicking on the message may trigger an ActionEvent.

Either the caption or the text may be null, but an NullPointerException is thrown if both are null. When displayed, the caption or text strings may be truncated on some platforms; the number of characters that may be displayed is platform-dependent.

**Note:** Some platforms may not support showing a message.

**Parameters:**caption - the caption displayed above the text, usually in bold; may be nulltext - the text displayed for the particular message; may be nullmessageType - an enum indicating the message type **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if both caption and text are null

### getSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getSize**()

Returns the size, in pixels, of the space that the tray icon occupies in the system tray. For the tray icon that is not yet added to the system tray, the returned size is equal to the result of the [SystemTray.getTrayIconSize()](http://docs.google.com/java/awt/SystemTray.html#getTrayIconSize()).

**Returns:**the size of the tray icon, in pixels**See Also:**[setImageAutoSize(boolean)](http://docs.google.com/java/awt/TrayIcon.html#setImageAutoSize(boolean)), [Image](http://docs.google.com/java/awt/Image.html), [getSize()](http://docs.google.com/java/awt/TrayIcon.html#getSize())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TrayIcon.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/Transparency.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/TrayIcon.MessageType.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/TrayIcon.html)    [**NO FRAMES**](http://docs.google.com/TrayIcon.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#3rdcrjn) |

[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).