| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ToolTipManager.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/Timer.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/ToolTipManager.insideTimerAction.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/ToolTipManager.html)    [**NO FRAMES**](http://docs.google.com/ToolTipManager.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | CONSTR | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | CONSTR | [METHOD](#3rdcrjn) |

## **javax.swing**

Class ToolTipManager

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.event.MouseAdapter](http://docs.google.com/java/awt/event/MouseAdapter.html)  
 **javax.swing.ToolTipManager**

**All Implemented Interfaces:** [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html), [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html), [EventListener](http://docs.google.com/java/util/EventListener.html)

public class **ToolTipManager**extends [MouseAdapter](http://docs.google.com/java/awt/event/MouseAdapter.html)implements [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)

Manages all the ToolTips in the system.

ToolTipManager contains numerous properties for configuring how long it will take for the tooltips to become visible, and how long till they hide. Consider a component that has a different tooltip based on where the mouse is, such as JTree. When the mouse moves into the JTree and over a region that has a valid tooltip, the tooltip will become visibile after initialDelay milliseconds. After dismissDelay milliseconds the tooltip will be hidden. If the mouse is over a region that has a valid tooltip, and the tooltip is currently visible, when the mouse moves to a region that doesn't have a valid tooltip the tooltip will be hidden. If the mouse then moves back into a region that has a valid tooltip within reshowDelay milliseconds, the tooltip will immediately be shown, otherwise the tooltip will be shown again after initialDelay milliseconds.

**See Also:**[JComponent.createToolTip()](http://docs.google.com/javax/swing/JComponent.html#createToolTip())

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**ToolTipManager.insideTimerAction**](http://docs.google.com/javax/swing/ToolTipManager.insideTimerAction.html) |
| protected  class | [**ToolTipManager.outsideTimerAction**](http://docs.google.com/javax/swing/ToolTipManager.outsideTimerAction.html) |
| protected  class | [**ToolTipManager.stillInsideTimerAction**](http://docs.google.com/javax/swing/ToolTipManager.stillInsideTimerAction.html) |

| **Field Summary** | |
| --- | --- |
| protected  boolean | [**heavyWeightPopupEnabled**](http://docs.google.com/javax/swing/ToolTipManager.html#heavyWeightPopupEnabled) |
| protected  boolean | [**lightWeightPopupEnabled**](http://docs.google.com/javax/swing/ToolTipManager.html#lightWeightPopupEnabled) |

| **Method Summary** | |
| --- | --- |
| int | [**getDismissDelay**](http://docs.google.com/javax/swing/ToolTipManager.html#getDismissDelay())()            Returns the dismissal delay value. |
| int | [**getInitialDelay**](http://docs.google.com/javax/swing/ToolTipManager.html#getInitialDelay())()            Returns the initial delay value. |
| int | [**getReshowDelay**](http://docs.google.com/javax/swing/ToolTipManager.html#getReshowDelay())()            Returns the reshow delay property. |
| boolean | [**isEnabled**](http://docs.google.com/javax/swing/ToolTipManager.html#isEnabled())()            Returns true if this object is enabled. |
| boolean | [**isLightWeightPopupEnabled**](http://docs.google.com/javax/swing/ToolTipManager.html#isLightWeightPopupEnabled())()            Returns true if lightweight (all-Java) Tooltips are in use, or false if heavyweight (native peer) Tooltips are being used. |
| void | [**mouseDragged**](http://docs.google.com/javax/swing/ToolTipManager.html#mouseDragged(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Called when the mouse is pressed and dragged. |
| void | [**mouseEntered**](http://docs.google.com/javax/swing/ToolTipManager.html#mouseEntered(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Called when the mouse enters the region of a component. |
| void | [**mouseExited**](http://docs.google.com/javax/swing/ToolTipManager.html#mouseExited(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Called when the mouse exits the region of a component. |
| void | [**mouseMoved**](http://docs.google.com/javax/swing/ToolTipManager.html#mouseMoved(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Called when the mouse is moved. |
| void | [**mousePressed**](http://docs.google.com/javax/swing/ToolTipManager.html#mousePressed(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Called when the mouse is pressed. |
| void | [**registerComponent**](http://docs.google.com/javax/swing/ToolTipManager.html#registerComponent(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) component)            Registers a component for tooltip management. |
| void | [**setDismissDelay**](http://docs.google.com/javax/swing/ToolTipManager.html#setDismissDelay(int))(int milliseconds)            Specifies the dismissal delay value. |
| void | [**setEnabled**](http://docs.google.com/javax/swing/ToolTipManager.html#setEnabled(boolean))(boolean flag)            Enables or disables the tooltip. |
| void | [**setInitialDelay**](http://docs.google.com/javax/swing/ToolTipManager.html#setInitialDelay(int))(int milliseconds)            Specifies the initial delay value. |
| void | [**setLightWeightPopupEnabled**](http://docs.google.com/javax/swing/ToolTipManager.html#setLightWeightPopupEnabled(boolean))(boolean aFlag)            When displaying the JToolTip, the ToolTipManager chooses to use a lightweight JPanel if it fits. |
| void | [**setReshowDelay**](http://docs.google.com/javax/swing/ToolTipManager.html#setReshowDelay(int))(int milliseconds)            Used to specify the amount of time before the user has to wait initialDelay milliseconds before a tooltip will be shown. |
| static [ToolTipManager](http://docs.google.com/javax/swing/ToolTipManager.html) | [**sharedInstance**](http://docs.google.com/javax/swing/ToolTipManager.html#sharedInstance())()            Returns a shared ToolTipManager instance. |
| void | [**unregisterComponent**](http://docs.google.com/javax/swing/ToolTipManager.html#unregisterComponent(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) component)            Removes a component from tooltip control. |

| **Methods inherited from class java.awt.event.**[**MouseAdapter**](http://docs.google.com/java/awt/event/MouseAdapter.html) |
| --- |
| [mouseClicked](http://docs.google.com/java/awt/event/MouseAdapter.html#mouseClicked(java.awt.event.MouseEvent)), [mouseReleased](http://docs.google.com/java/awt/event/MouseAdapter.html#mouseReleased(java.awt.event.MouseEvent)), [mouseWheelMoved](http://docs.google.com/java/awt/event/MouseAdapter.html#mouseWheelMoved(java.awt.event.MouseWheelEvent)) |

| **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)) |

| **Field Detail** |
| --- |

### lightWeightPopupEnabled

protected boolean **lightWeightPopupEnabled**

### heavyWeightPopupEnabled

protected boolean **heavyWeightPopupEnabled**

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

### setEnabled

public void **setEnabled**(boolean flag)

Enables or disables the tooltip.

**Parameters:**flag - true to enable the tip, false otherwise

### isEnabled

public boolean **isEnabled**()

Returns true if this object is enabled.

**Returns:**true if this object is enabled, false otherwise

### setLightWeightPopupEnabled

public void **setLightWeightPopupEnabled**(boolean aFlag)

When displaying the JToolTip, the ToolTipManager chooses to use a lightweight JPanel if it fits. This method allows you to disable this feature. You have to do disable it if your application mixes light weight and heavy weights components.

**Parameters:**aFlag - true if a lightweight panel is desired, false otherwise

### isLightWeightPopupEnabled

public boolean **isLightWeightPopupEnabled**()

Returns true if lightweight (all-Java) Tooltips are in use, or false if heavyweight (native peer) Tooltips are being used.

**Returns:**true if lightweight ToolTips are in use

### setInitialDelay

public void **setInitialDelay**(int milliseconds)

Specifies the initial delay value.

**Parameters:**milliseconds - the number of milliseconds to delay (after the cursor has paused) before displaying the tooltip**See Also:**[getInitialDelay()](http://docs.google.com/javax/swing/ToolTipManager.html#getInitialDelay())

### getInitialDelay

public int **getInitialDelay**()

Returns the initial delay value.

**Returns:**an integer representing the initial delay value, in milliseconds**See Also:**[setInitialDelay(int)](http://docs.google.com/javax/swing/ToolTipManager.html#setInitialDelay(int))

### setDismissDelay

public void **setDismissDelay**(int milliseconds)

Specifies the dismissal delay value.

**Parameters:**milliseconds - the number of milliseconds to delay before taking away the tooltip**See Also:**[getDismissDelay()](http://docs.google.com/javax/swing/ToolTipManager.html#getDismissDelay())

### getDismissDelay

public int **getDismissDelay**()

Returns the dismissal delay value.

**Returns:**an integer representing the dismissal delay value, in milliseconds**See Also:**[setDismissDelay(int)](http://docs.google.com/javax/swing/ToolTipManager.html#setDismissDelay(int))

### setReshowDelay

public void **setReshowDelay**(int milliseconds)

Used to specify the amount of time before the user has to wait initialDelay milliseconds before a tooltip will be shown. That is, if the tooltip is hidden, and the user moves into a region of the same Component that has a valid tooltip within milliseconds milliseconds the tooltip will immediately be shown. Otherwise, if the user moves into a region with a valid tooltip after milliseconds milliseconds, the user will have to wait an additional initialDelay milliseconds before the tooltip is shown again.

**Parameters:**milliseconds - time in milliseconds**See Also:**[getReshowDelay()](http://docs.google.com/javax/swing/ToolTipManager.html#getReshowDelay())

### getReshowDelay

public int **getReshowDelay**()

Returns the reshow delay property.

**Returns:**reshown delay property**See Also:**[setReshowDelay(int)](http://docs.google.com/javax/swing/ToolTipManager.html#setReshowDelay(int))

### sharedInstance

public static [ToolTipManager](http://docs.google.com/javax/swing/ToolTipManager.html) **sharedInstance**()

Returns a shared ToolTipManager instance.

**Returns:**a shared ToolTipManager object

### registerComponent

public void **registerComponent**([JComponent](http://docs.google.com/javax/swing/JComponent.html) component)

Registers a component for tooltip management.

This will register key bindings to show and hide the tooltip text only if component has focus bindings. This is done so that components that are not normally focus traversable, such as JLabel, are not made focus traversable as a result of invoking this method.

**Parameters:**component - a JComponent object to add**See Also:**[Component.isFocusTraversable()](http://docs.google.com/java/awt/Component.html#isFocusTraversable())

### unregisterComponent

public void **unregisterComponent**([JComponent](http://docs.google.com/javax/swing/JComponent.html) component)

Removes a component from tooltip control.

**Parameters:**component - a JComponent object to remove

### mouseEntered

public void **mouseEntered**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Called when the mouse enters the region of a component. This determines whether the tool tip should be shown.

**Specified by:**[mouseEntered](http://docs.google.com/java/awt/event/MouseListener.html#mouseEntered(java.awt.event.MouseEvent)) in interface [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html)**Overrides:**[mouseEntered](http://docs.google.com/java/awt/event/MouseAdapter.html#mouseEntered(java.awt.event.MouseEvent)) in class [MouseAdapter](http://docs.google.com/java/awt/event/MouseAdapter.html) **Parameters:**event - the event in question

### mouseExited

public void **mouseExited**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Called when the mouse exits the region of a component. Any tool tip showing should be hidden.

**Specified by:**[mouseExited](http://docs.google.com/java/awt/event/MouseListener.html#mouseExited(java.awt.event.MouseEvent)) in interface [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html)**Overrides:**[mouseExited](http://docs.google.com/java/awt/event/MouseAdapter.html#mouseExited(java.awt.event.MouseEvent)) in class [MouseAdapter](http://docs.google.com/java/awt/event/MouseAdapter.html) **Parameters:**event - the event in question

### mousePressed

public void **mousePressed**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Called when the mouse is pressed. Any tool tip showing should be hidden.

**Specified by:**[mousePressed](http://docs.google.com/java/awt/event/MouseListener.html#mousePressed(java.awt.event.MouseEvent)) in interface [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html)**Overrides:**[mousePressed](http://docs.google.com/java/awt/event/MouseAdapter.html#mousePressed(java.awt.event.MouseEvent)) in class [MouseAdapter](http://docs.google.com/java/awt/event/MouseAdapter.html) **Parameters:**event - the event in question

### mouseDragged

public void **mouseDragged**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Called when the mouse is pressed and dragged. Does nothing.

**Specified by:**[mouseDragged](http://docs.google.com/java/awt/event/MouseMotionListener.html#mouseDragged(java.awt.event.MouseEvent)) in interface [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)**Overrides:**[mouseDragged](http://docs.google.com/java/awt/event/MouseAdapter.html#mouseDragged(java.awt.event.MouseEvent)) in class [MouseAdapter](http://docs.google.com/java/awt/event/MouseAdapter.html) **Parameters:**event - the event in question

### mouseMoved

public void **mouseMoved**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Called when the mouse is moved. Determines whether the tool tip should be displayed.

**Specified by:**[mouseMoved](http://docs.google.com/java/awt/event/MouseMotionListener.html#mouseMoved(java.awt.event.MouseEvent)) in interface [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)**Overrides:**[mouseMoved](http://docs.google.com/java/awt/event/MouseAdapter.html#mouseMoved(java.awt.event.MouseEvent)) in class [MouseAdapter](http://docs.google.com/java/awt/event/MouseAdapter.html) **Parameters:**event - the event in question

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ToolTipManager.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/Timer.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/ToolTipManager.insideTimerAction.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/ToolTipManager.html)    [**NO FRAMES**](http://docs.google.com/ToolTipManager.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | CONSTR | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | CONSTR | [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).