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

## **javax.swing.text**

Interface Caret

**All Known Implementing Classes:** [BasicTextUI.BasicCaret](http://docs.google.com/javax/swing/plaf/basic/BasicTextUI.BasicCaret.html), [DefaultCaret](http://docs.google.com/javax/swing/text/DefaultCaret.html)

public interface **Caret**

A place within a document view that represents where things can be inserted into the document model. A caret has a position in the document referred to as a dot. The dot is where the caret is currently located in the model. There is a second position maintained by the caret that represents the other end of a selection called mark. If there is no selection the dot and mark will be equal. If a selection exists, the two values will be different.

The dot can be placed by either calling setDot or moveDot. Setting the dot has the effect of removing any selection that may have previously existed. The dot and mark will be equal. Moving the dot has the effect of creating a selection as the mark is left at whatever position it previously had.

| **Method Summary** | |
| --- | --- |
| void | [**addChangeListener**](http://docs.google.com/javax/swing/text/Caret.html#addChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Adds a listener to track whenever the caret position has been changed. |
| void | [**deinstall**](http://docs.google.com/javax/swing/text/Caret.html#deinstall(javax.swing.text.JTextComponent))([JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) c)            Called when the UI is being removed from the interface of a JTextComponent. |
| int | [**getBlinkRate**](http://docs.google.com/javax/swing/text/Caret.html#getBlinkRate())()            Gets the blink rate of the caret. |
| int | [**getDot**](http://docs.google.com/javax/swing/text/Caret.html#getDot())()            Fetches the current position of the caret. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getMagicCaretPosition**](http://docs.google.com/javax/swing/text/Caret.html#getMagicCaretPosition())()            Gets the current caret visual location. |
| int | [**getMark**](http://docs.google.com/javax/swing/text/Caret.html#getMark())()            Fetches the current position of the mark. |
| void | [**install**](http://docs.google.com/javax/swing/text/Caret.html#install(javax.swing.text.JTextComponent))([JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) c)            Called when the UI is being installed into the interface of a JTextComponent. |
| boolean | [**isSelectionVisible**](http://docs.google.com/javax/swing/text/Caret.html#isSelectionVisible())()            Determines if the selection is currently visible. |
| boolean | [**isVisible**](http://docs.google.com/javax/swing/text/Caret.html#isVisible())()            Determines if the caret is currently visible. |
| void | [**moveDot**](http://docs.google.com/javax/swing/text/Caret.html#moveDot(int))(int dot)            Moves the caret position (dot) to some other position, leaving behind the mark. |
| void | [**paint**](http://docs.google.com/javax/swing/text/Caret.html#paint(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Renders the caret. |
| void | [**removeChangeListener**](http://docs.google.com/javax/swing/text/Caret.html#removeChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Removes a listener that was tracking caret position changes. |
| void | [**setBlinkRate**](http://docs.google.com/javax/swing/text/Caret.html#setBlinkRate(int))(int rate)            Sets the blink rate of the caret. |
| void | [**setDot**](http://docs.google.com/javax/swing/text/Caret.html#setDot(int))(int dot)            Sets the caret position to some position. |
| void | [**setMagicCaretPosition**](http://docs.google.com/javax/swing/text/Caret.html#setMagicCaretPosition(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Set the current caret visual location. |
| void | [**setSelectionVisible**](http://docs.google.com/javax/swing/text/Caret.html#setSelectionVisible(boolean))(boolean v)            Sets the visibility of the selection |
| void | [**setVisible**](http://docs.google.com/javax/swing/text/Caret.html#setVisible(boolean))(boolean v)            Sets the visibility of the caret. |

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

### install

void **install**([JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) c)

Called when the UI is being installed into the interface of a JTextComponent. This can be used to gain access to the model that is being navigated by the implementation of this interface.

**Parameters:**c - the JTextComponent

### deinstall

void **deinstall**([JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) c)

Called when the UI is being removed from the interface of a JTextComponent. This is used to unregister any listeners that were attached.

**Parameters:**c - the JTextComponent

### paint

void **paint**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Renders the caret. This method is called by UI classes.

**Parameters:**g - the graphics context

### addChangeListener

void **addChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Adds a listener to track whenever the caret position has been changed.

**Parameters:**l - the change listener

### removeChangeListener

void **removeChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Removes a listener that was tracking caret position changes.

**Parameters:**l - the change listener

### isVisible

boolean **isVisible**()

Determines if the caret is currently visible.

**Returns:**true if the caret is visible else false

### setVisible

void **setVisible**(boolean v)

Sets the visibility of the caret.

**Parameters:**v - true if the caret should be shown, and false if the caret should be hidden

### isSelectionVisible

boolean **isSelectionVisible**()

Determines if the selection is currently visible.

**Returns:**true if the caret is visible else false

### setSelectionVisible

void **setSelectionVisible**(boolean v)

Sets the visibility of the selection

**Parameters:**v - true if the caret should be shown, and false if the caret should be hidden

### setMagicCaretPosition

void **setMagicCaretPosition**([Point](http://docs.google.com/java/awt/Point.html) p)

Set the current caret visual location. This can be used when moving between lines that have uneven end positions (such as when caret up or down actions occur). If text flows left-to-right or right-to-left the x-coordinate will indicate the desired navigation location for vertical movement. If the text flow is top-to-bottom, the y-coordinate will indicate the desired navigation location for horizontal movement.

**Parameters:**p - the Point to use for the saved position. This can be null to indicate there is no visual location.

### getMagicCaretPosition

[Point](http://docs.google.com/java/awt/Point.html) **getMagicCaretPosition**()

Gets the current caret visual location.

**Returns:**the visual position.**See Also:**[setMagicCaretPosition(java.awt.Point)](http://docs.google.com/javax/swing/text/Caret.html#setMagicCaretPosition(java.awt.Point))

### setBlinkRate

void **setBlinkRate**(int rate)

Sets the blink rate of the caret. This determines if and how fast the caret blinks, commonly used as one way to attract attention to the caret.

**Parameters:**rate - the delay in milliseconds >= 0. If this is zero the caret will not blink.

### getBlinkRate

int **getBlinkRate**()

Gets the blink rate of the caret. This determines if and how fast the caret blinks, commonly used as one way to attract attention to the caret.

**Returns:**the delay in milliseconds >= 0. If this is zero the caret will not blink.

### getDot

int **getDot**()

Fetches the current position of the caret.

**Returns:**the position >= 0

### getMark

int **getMark**()

Fetches the current position of the mark. If there is a selection, the mark will not be the same as the dot.

**Returns:**the position >= 0

### setDot

void **setDot**(int dot)

Sets the caret position to some position. This causes the mark to become the same as the dot, effectively setting the selection range to zero.

If the parameter is negative or beyond the length of the document, the caret is placed at the beginning or at the end, respectively.

**Parameters:**dot - the new position to set the caret to

### moveDot

void **moveDot**(int dot)

Moves the caret position (dot) to some other position, leaving behind the mark. This is useful for making selections.

**Parameters:**dot - the new position to move the caret to >= 0

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Caret.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/text/BoxView.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/ChangedCharSetException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/Caret.html)    [**NO FRAMES**](http://docs.google.com/Caret.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).