| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ComponentView.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/ChangedCharSetException.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/CompositeView.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/ComponentView.html)    [**NO FRAMES**](http://docs.google.com/ComponentView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **javax.swing.text**

Class ComponentView

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.text.View](http://docs.google.com/javax/swing/text/View.html)  
 **javax.swing.text.ComponentView**

**All Implemented Interfaces:** [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html) **Direct Known Subclasses:** [FormView](http://docs.google.com/javax/swing/text/html/FormView.html), [ObjectView](http://docs.google.com/javax/swing/text/html/ObjectView.html)

public class **ComponentView**extends [View](http://docs.google.com/javax/swing/text/View.html)

Component decorator that implements the view interface. The entire element is used to represent the component. This acts as a gateway from the display-only View implementations to interactive lightweight components (ie it allows components to be embedded into the View hierarchy).

The component is placed relative to the text baseline according to the value returned by Component.getAlignmentY. For Swing components this value can be conveniently set using the method JComponent.setAlignmentY. For example, setting a value of 0.75 will cause 75 percent of the component to be above the baseline, and 25 percent of the component to be below the baseline.

This class is implemented to do the extra work necessary to work properly in the presence of multiple threads (i.e. from asynchronous notification of model changes for example) by ensuring that all component access is done on the event thread.

The component used is determined by the return value of the createComponent method. The default implementation of this method is to return the component held as an attribute of the element (by calling StyleConstants.getComponent). A limitation of this behavior is that the component cannot be used by more than one text component (i.e. with a shared model). Subclasses can remove this constraint by implementing the createComponent to actually create a component based upon some kind of specification contained in the attributes. The ObjectView class in the html package is an example of a ComponentView implementation that supports multiple component views of a shared model.

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class javax.swing.text.**[**View**](http://docs.google.com/javax/swing/text/View.html) |
| --- |
| [BadBreakWeight](http://docs.google.com/javax/swing/text/View.html#BadBreakWeight), [ExcellentBreakWeight](http://docs.google.com/javax/swing/text/View.html#ExcellentBreakWeight), [ForcedBreakWeight](http://docs.google.com/javax/swing/text/View.html#ForcedBreakWeight), [GoodBreakWeight](http://docs.google.com/javax/swing/text/View.html#GoodBreakWeight), [X\_AXIS](http://docs.google.com/javax/swing/text/View.html#X_AXIS), [Y\_AXIS](http://docs.google.com/javax/swing/text/View.html#Y_AXIS) |

| **Fields inherited from interface javax.swing.**[**SwingConstants**](http://docs.google.com/javax/swing/SwingConstants.html) |
| --- |
| [BOTTOM](http://docs.google.com/javax/swing/SwingConstants.html#BOTTOM), [CENTER](http://docs.google.com/javax/swing/SwingConstants.html#CENTER), [EAST](http://docs.google.com/javax/swing/SwingConstants.html#EAST), [HORIZONTAL](http://docs.google.com/javax/swing/SwingConstants.html#HORIZONTAL), [LEADING](http://docs.google.com/javax/swing/SwingConstants.html#LEADING), [LEFT](http://docs.google.com/javax/swing/SwingConstants.html#LEFT), [NEXT](http://docs.google.com/javax/swing/SwingConstants.html#NEXT), [NORTH](http://docs.google.com/javax/swing/SwingConstants.html#NORTH), [NORTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_EAST), [NORTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_WEST), [PREVIOUS](http://docs.google.com/javax/swing/SwingConstants.html#PREVIOUS), [RIGHT](http://docs.google.com/javax/swing/SwingConstants.html#RIGHT), [SOUTH](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH), [SOUTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_EAST), [SOUTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_WEST), [TOP](http://docs.google.com/javax/swing/SwingConstants.html#TOP), [TRAILING](http://docs.google.com/javax/swing/SwingConstants.html#TRAILING), [VERTICAL](http://docs.google.com/javax/swing/SwingConstants.html#VERTICAL), [WEST](http://docs.google.com/javax/swing/SwingConstants.html#WEST) |

| **Constructor Summary** | |
| --- | --- |
| [**ComponentView**](http://docs.google.com/javax/swing/text/ComponentView.html#ComponentView(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Creates a new ComponentView object. |

| **Method Summary** | |
| --- | --- |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**createComponent**](http://docs.google.com/javax/swing/text/ComponentView.html#createComponent())()            Create the component that is associated with this view. |
| float | [**getAlignment**](http://docs.google.com/javax/swing/text/ComponentView.html#getAlignment(int))(int axis)            Determines the desired alignment for this view along an axis. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponent**](http://docs.google.com/javax/swing/text/ComponentView.html#getComponent())()            Fetch the component associated with the view. |
| float | [**getMaximumSpan**](http://docs.google.com/javax/swing/text/ComponentView.html#getMaximumSpan(int))(int axis)            Determines the maximum span for this view along an axis. |
| float | [**getMinimumSpan**](http://docs.google.com/javax/swing/text/ComponentView.html#getMinimumSpan(int))(int axis)            Determines the minimum span for this view along an axis. |
| float | [**getPreferredSpan**](http://docs.google.com/javax/swing/text/ComponentView.html#getPreferredSpan(int))(int axis)            Determines the preferred span for this view along an axis. |
| [Shape](http://docs.google.com/java/awt/Shape.html) | [**modelToView**](http://docs.google.com/javax/swing/text/ComponentView.html#modelToView(int,%20java.awt.Shape,%20javax.swing.text.Position.Bias))(int pos, [Shape](http://docs.google.com/java/awt/Shape.html) a, [Position.Bias](http://docs.google.com/javax/swing/text/Position.Bias.html) b)            Provides a mapping from the coordinate space of the model to that of the view. |
| void | [**paint**](http://docs.google.com/javax/swing/text/ComponentView.html#paint(java.awt.Graphics,%20java.awt.Shape))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Shape](http://docs.google.com/java/awt/Shape.html) a)            The real paint behavior occurs naturally from the association that the component has with its parent container (the same container hosting this view). |
| void | [**setParent**](http://docs.google.com/javax/swing/text/ComponentView.html#setParent(javax.swing.text.View))([View](http://docs.google.com/javax/swing/text/View.html) p)            Sets the parent for a child view. |
| int | [**viewToModel**](http://docs.google.com/javax/swing/text/ComponentView.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D))(float x, float y, [Shape](http://docs.google.com/java/awt/Shape.html) a, [Position.Bias](http://docs.google.com/javax/swing/text/Position.Bias.html)[] bias)            Provides a mapping from the view coordinate space to the logical coordinate space of the model. |

| **Methods inherited from class javax.swing.text.**[**View**](http://docs.google.com/javax/swing/text/View.html) |
| --- |
| [append](http://docs.google.com/javax/swing/text/View.html#append(javax.swing.text.View)), [breakView](http://docs.google.com/javax/swing/text/View.html#breakView(int,%20int,%20float,%20float)), [changedUpdate](http://docs.google.com/javax/swing/text/View.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [createFragment](http://docs.google.com/javax/swing/text/View.html#createFragment(int,%20int)), [forwardUpdate](http://docs.google.com/javax/swing/text/View.html#forwardUpdate(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [forwardUpdateToView](http://docs.google.com/javax/swing/text/View.html#forwardUpdateToView(javax.swing.text.View,%20javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [getAttributes](http://docs.google.com/javax/swing/text/View.html#getAttributes()), [getBreakWeight](http://docs.google.com/javax/swing/text/View.html#getBreakWeight(int,%20float,%20float)), [getChildAllocation](http://docs.google.com/javax/swing/text/View.html#getChildAllocation(int,%20java.awt.Shape)), [getContainer](http://docs.google.com/javax/swing/text/View.html#getContainer()), [getDocument](http://docs.google.com/javax/swing/text/View.html#getDocument()), [getElement](http://docs.google.com/javax/swing/text/View.html#getElement()), [getEndOffset](http://docs.google.com/javax/swing/text/View.html#getEndOffset()), [getGraphics](http://docs.google.com/javax/swing/text/View.html#getGraphics()), [getNextVisualPositionFrom](http://docs.google.com/javax/swing/text/View.html#getNextVisualPositionFrom(int,%20javax.swing.text.Position.Bias,%20java.awt.Shape,%20int,%20javax.swing.text.Position.Bias%5B%5D)), [getParent](http://docs.google.com/javax/swing/text/View.html#getParent()), [getResizeWeight](http://docs.google.com/javax/swing/text/View.html#getResizeWeight(int)), [getStartOffset](http://docs.google.com/javax/swing/text/View.html#getStartOffset()), [getToolTipText](http://docs.google.com/javax/swing/text/View.html#getToolTipText(float,%20float,%20java.awt.Shape)), [getView](http://docs.google.com/javax/swing/text/View.html#getView(int)), [getViewCount](http://docs.google.com/javax/swing/text/View.html#getViewCount()), [getViewFactory](http://docs.google.com/javax/swing/text/View.html#getViewFactory()), [getViewIndex](http://docs.google.com/javax/swing/text/View.html#getViewIndex(float,%20float,%20java.awt.Shape)), [getViewIndex](http://docs.google.com/javax/swing/text/View.html#getViewIndex(int,%20javax.swing.text.Position.Bias)), [insert](http://docs.google.com/javax/swing/text/View.html#insert(int,%20javax.swing.text.View)), [insertUpdate](http://docs.google.com/javax/swing/text/View.html#insertUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [isVisible](http://docs.google.com/javax/swing/text/View.html#isVisible()), [modelToView](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20javax.swing.text.Position.Bias,%20int,%20javax.swing.text.Position.Bias,%20java.awt.Shape)), [modelToView](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20java.awt.Shape)), [preferenceChanged](http://docs.google.com/javax/swing/text/View.html#preferenceChanged(javax.swing.text.View,%20boolean,%20boolean)), [remove](http://docs.google.com/javax/swing/text/View.html#remove(int)), [removeAll](http://docs.google.com/javax/swing/text/View.html#removeAll()), [removeUpdate](http://docs.google.com/javax/swing/text/View.html#removeUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [replace](http://docs.google.com/javax/swing/text/View.html#replace(int,%20int,%20javax.swing.text.View%5B%5D)), [setSize](http://docs.google.com/javax/swing/text/View.html#setSize(float,%20float)), [updateChildren](http://docs.google.com/javax/swing/text/View.html#updateChildren(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20javax.swing.text.ViewFactory)), [updateLayout](http://docs.google.com/javax/swing/text/View.html#updateLayout(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20java.awt.Shape)), [viewToModel](http://docs.google.com/javax/swing/text/View.html#viewToModel(float,%20float,%20java.awt.Shape)) |

| **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** |
| --- |

### ComponentView

public **ComponentView**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)

Creates a new ComponentView object.

**Parameters:**elem - the element to decorate

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

### createComponent

protected [Component](http://docs.google.com/java/awt/Component.html) **createComponent**()

Create the component that is associated with this view. This will be called when it has been determined that a new component is needed. This would result from a call to setParent or as a result of being notified that attributes have changed.

### getComponent

public final [Component](http://docs.google.com/java/awt/Component.html) **getComponent**()

Fetch the component associated with the view.

### paint

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

The real paint behavior occurs naturally from the association that the component has with its parent container (the same container hosting this view). This is implemented to do nothing.

**Specified by:**[paint](http://docs.google.com/javax/swing/text/View.html#paint(java.awt.Graphics,%20java.awt.Shape)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**g - the graphics contexta - the shape**See Also:**[View.paint(java.awt.Graphics, java.awt.Shape)](http://docs.google.com/javax/swing/text/View.html#paint(java.awt.Graphics,%20java.awt.Shape))

### getPreferredSpan

public float **getPreferredSpan**(int axis)

Determines the preferred span for this view along an axis. This is implemented to return the value returned by Component.getPreferredSize along the axis of interest.

**Specified by:**[getPreferredSpan](http://docs.google.com/javax/swing/text/View.html#getPreferredSpan(int)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**axis - may be either View.X\_AXIS or View.Y\_AXIS **Returns:**the span the view would like to be rendered into >= 0. Typically the view is told to render into the span that is returned, although there is no guarantee. The parent may choose to resize or break the view. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - for an invalid axis**See Also:**[View.getPreferredSpan(int)](http://docs.google.com/javax/swing/text/View.html#getPreferredSpan(int))

### getMinimumSpan

public float **getMinimumSpan**(int axis)

Determines the minimum span for this view along an axis. This is implemented to return the value returned by Component.getMinimumSize along the axis of interest.

**Overrides:**[getMinimumSpan](http://docs.google.com/javax/swing/text/View.html#getMinimumSpan(int)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**axis - may be either View.X\_AXIS or View.Y\_AXIS **Returns:**the span the view would like to be rendered into >= 0. Typically the view is told to render into the span that is returned, although there is no guarantee. The parent may choose to resize or break the view. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - for an invalid axis**See Also:**[View.getPreferredSpan(int)](http://docs.google.com/javax/swing/text/View.html#getPreferredSpan(int))

### getMaximumSpan

public float **getMaximumSpan**(int axis)

Determines the maximum span for this view along an axis. This is implemented to return the value returned by Component.getMaximumSize along the axis of interest.

**Overrides:**[getMaximumSpan](http://docs.google.com/javax/swing/text/View.html#getMaximumSpan(int)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**axis - may be either View.X\_AXIS or View.Y\_AXIS **Returns:**the span the view would like to be rendered into >= 0. Typically the view is told to render into the span that is returned, although there is no guarantee. The parent may choose to resize or break the view. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - for an invalid axis**See Also:**[View.getPreferredSpan(int)](http://docs.google.com/javax/swing/text/View.html#getPreferredSpan(int))

### getAlignment

public float **getAlignment**(int axis)

Determines the desired alignment for this view along an axis. This is implemented to give the alignment of the embedded component.

**Overrides:**[getAlignment](http://docs.google.com/javax/swing/text/View.html#getAlignment(int)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**axis - may be either View.X\_AXIS or View.Y\_AXIS **Returns:**the desired alignment. This should be a value between 0.0 and 1.0 where 0 indicates alignment at the origin and 1.0 indicates alignment to the full span away from the origin. An alignment of 0.5 would be the center of the view.

### setParent

public void **setParent**([View](http://docs.google.com/javax/swing/text/View.html) p)

Sets the parent for a child view. The parent calls this on the child to tell it who its parent is, giving the view access to things like the hosting Container. The superclass behavior is executed, followed by a call to createComponent if the parent view parameter is non-null and a component has not yet been created. The embedded components parent is then set to the value returned by getContainer. If the parent view parameter is null, this view is being cleaned up, thus the component is removed from its parent.

The changing of the component hierarchy will touch the component lock, which is the one thing that is not safe from the View hierarchy. Therefore, this functionality is executed immediately if on the event thread, or is queued on the event queue if called from another thread (notification of change from an asynchronous update).

**Overrides:**[setParent](http://docs.google.com/javax/swing/text/View.html#setParent(javax.swing.text.View)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**p - the parent

### modelToView

public [Shape](http://docs.google.com/java/awt/Shape.html) **modelToView**(int pos,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [Position.Bias](http://docs.google.com/javax/swing/text/Position.Bias.html) b)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Provides a mapping from the coordinate space of the model to that of the view.

**Specified by:**[modelToView](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20java.awt.Shape,%20javax.swing.text.Position.Bias)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**pos - the position to convert >= 0a - the allocated region to render intob - the bias toward the previous character or the next character represented by the offset, in case the position is a boundary of two views; b will have one of these values:

* Position.Bias.Forward
* Position.Bias.Backward

**Returns:**the bounding box of the given position is returned **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if the given position does not represent a valid location in the associated document**See Also:**[View.modelToView(int, java.awt.Shape, javax.swing.text.Position.Bias)](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20java.awt.Shape,%20javax.swing.text.Position.Bias))

### viewToModel

public int **viewToModel**(float x,  
 float y,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [Position.Bias](http://docs.google.com/javax/swing/text/Position.Bias.html)[] bias)

Provides a mapping from the view coordinate space to the logical coordinate space of the model.

**Specified by:**[viewToModel](http://docs.google.com/javax/swing/text/View.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**x - the X coordinate >= 0y - the Y coordinate >= 0a - the allocated region to render into **Returns:**the location within the model that best represents the given point in the view**See Also:**[View.viewToModel(float, float, java.awt.Shape, javax.swing.text.Position.Bias[])](http://docs.google.com/javax/swing/text/View.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ComponentView.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/ChangedCharSetException.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/CompositeView.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/ComponentView.html)    [**NO FRAMES**](http://docs.google.com/ComponentView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).