| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PlainView.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/PlainDocument.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/Position.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/PlainView.html)    [**NO FRAMES**](http://docs.google.com/PlainView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **javax.swing.text**

Class PlainView

[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.PlainView**

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

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

Implements View interface for a simple multi-line text view that has text in one font and color. The view represents each child element as a line of text.

**See Also:**[View](http://docs.google.com/javax/swing/text/View.html)

| **Field Summary** | |
| --- | --- |
| protected  [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) | [**metrics**](http://docs.google.com/javax/swing/text/PlainView.html#metrics)            Font metrics for the current font. |

| **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** | |
| --- | --- |
| [**PlainView**](http://docs.google.com/javax/swing/text/PlainView.html#PlainView(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Constructs a new PlainView wrapped on an element. |

| **Method Summary** | |
| --- | --- |
| void | [**changedUpdate**](http://docs.google.com/javax/swing/text/PlainView.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) changes, [Shape](http://docs.google.com/java/awt/Shape.html) a, [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)            Gives notification from the document that attributes were changed in a location that this view is responsible for. |
| protected  void | [**damageLineRange**](http://docs.google.com/javax/swing/text/PlainView.html#damageLineRange(int,%20int,%20java.awt.Shape,%20java.awt.Component))(int line0, int line1, [Shape](http://docs.google.com/java/awt/Shape.html) a, [Component](http://docs.google.com/java/awt/Component.html) host)            Repaint the given line range. |
| protected  void | [**drawLine**](http://docs.google.com/javax/swing/text/PlainView.html#drawLine(int,%20java.awt.Graphics,%20int,%20int))(int lineIndex, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x, int y)            Renders a line of text, suppressing whitespace at the end and expanding any tabs. |
| protected  int | [**drawSelectedText**](http://docs.google.com/javax/swing/text/PlainView.html#drawSelectedText(java.awt.Graphics,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x, int y, int p0, int p1)            Renders the given range in the model as selected text. |
| protected  int | [**drawUnselectedText**](http://docs.google.com/javax/swing/text/PlainView.html#drawUnselectedText(java.awt.Graphics,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x, int y, int p0, int p1)            Renders the given range in the model as normal unselected text. |
| protected  [Segment](http://docs.google.com/javax/swing/text/Segment.html) | [**getLineBuffer**](http://docs.google.com/javax/swing/text/PlainView.html#getLineBuffer())()            Gives access to a buffer that can be used to fetch text from the associated document. |
| float | [**getPreferredSpan**](http://docs.google.com/javax/swing/text/PlainView.html#getPreferredSpan(int))(int axis)            Determines the preferred span for this view along an axis. |
| protected  int | [**getTabSize**](http://docs.google.com/javax/swing/text/PlainView.html#getTabSize())()            Returns the tab size set for the document, defaulting to 8. |
| void | [**insertUpdate**](http://docs.google.com/javax/swing/text/PlainView.html#insertUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) changes, [Shape](http://docs.google.com/java/awt/Shape.html) a, [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)            Gives notification that something was inserted into the document in a location that this view is responsible for. |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**lineToRect**](http://docs.google.com/javax/swing/text/PlainView.html#lineToRect(java.awt.Shape,%20int))([Shape](http://docs.google.com/java/awt/Shape.html) a, int line)            Determine the rectangle that represents the given line. |
| [Shape](http://docs.google.com/java/awt/Shape.html) | [**modelToView**](http://docs.google.com/javax/swing/text/PlainView.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 document model coordinate space to the coordinate space of the view mapped to it. |
| float | [**nextTabStop**](http://docs.google.com/javax/swing/text/PlainView.html#nextTabStop(float,%20int))(float x, int tabOffset)            Returns the next tab stop position after a given reference position. |
| void | [**paint**](http://docs.google.com/javax/swing/text/PlainView.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)            Renders using the given rendering surface and area on that surface. |
| void | [**removeUpdate**](http://docs.google.com/javax/swing/text/PlainView.html#removeUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) changes, [Shape](http://docs.google.com/java/awt/Shape.html) a, [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)            Gives notification that something was removed from the document in a location that this view is responsible for. |
| void | [**setSize**](http://docs.google.com/javax/swing/text/PlainView.html#setSize(float,%20float))(float width, float height)            Sets the size of the view. |
| protected  void | [**updateDamage**](http://docs.google.com/javax/swing/text/PlainView.html#updateDamage(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) changes, [Shape](http://docs.google.com/java/awt/Shape.html) a, [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)            Repaint the region of change covered by the given document event. |
| protected  void | [**updateMetrics**](http://docs.google.com/javax/swing/text/PlainView.html#updateMetrics())()            Checks to see if the font metrics and longest line are up-to-date. |
| int | [**viewToModel**](http://docs.google.com/javax/swing/text/PlainView.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D))(float fx, float fy, [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)), [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)), [getAlignment](http://docs.google.com/javax/swing/text/View.html#getAlignment(int)), [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()), [getMaximumSpan](http://docs.google.com/javax/swing/text/View.html#getMaximumSpan(int)), [getMinimumSpan](http://docs.google.com/javax/swing/text/View.html#getMinimumSpan(int)), [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)), [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()), [replace](http://docs.google.com/javax/swing/text/View.html#replace(int,%20int,%20javax.swing.text.View%5B%5D)), [setParent](http://docs.google.com/javax/swing/text/View.html#setParent(javax.swing.text.View)), [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)) |

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

### metrics

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

Font metrics for the current font.

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

### PlainView

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

Constructs a new PlainView wrapped on an element.

**Parameters:**elem - the element

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

### getTabSize

protected int **getTabSize**()

Returns the tab size set for the document, defaulting to 8.

**Returns:**the tab size

### drawLine

protected void **drawLine**(int lineIndex,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x,  
 int y)

Renders a line of text, suppressing whitespace at the end and expanding any tabs. This is implemented to make calls to the methods drawUnselectedText and drawSelectedText so that the way selected and unselected text are rendered can be customized.

**Parameters:**lineIndex - the line to draw >= 0g - the Graphics contextx - the starting X position >= 0y - the starting Y position >= 0**See Also:**[drawUnselectedText(java.awt.Graphics, int, int, int, int)](http://docs.google.com/javax/swing/text/PlainView.html#drawUnselectedText(java.awt.Graphics,%20int,%20int,%20int,%20int)), [drawSelectedText(java.awt.Graphics, int, int, int, int)](http://docs.google.com/javax/swing/text/PlainView.html#drawSelectedText(java.awt.Graphics,%20int,%20int,%20int,%20int))

### drawUnselectedText

protected int **drawUnselectedText**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x,  
 int y,  
 int p0,  
 int p1)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Renders the given range in the model as normal unselected text. Uses the foreground or disabled color to render the text.

**Parameters:**g - the graphics contextx - the starting X coordinate >= 0y - the starting Y coordinate >= 0p0 - the beginning position in the model >= 0p1 - the ending position in the model >= 0 **Returns:**the X location of the end of the range >= 0 **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if the range is invalid

### drawSelectedText

protected int **drawSelectedText**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x,  
 int y,  
 int p0,  
 int p1)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Renders the given range in the model as selected text. This is implemented to render the text in the color specified in the hosting component. It assumes the highlighter will render the selected background.

**Parameters:**g - the graphics contextx - the starting X coordinate >= 0y - the starting Y coordinate >= 0p0 - the beginning position in the model >= 0p1 - the ending position in the model >= 0 **Returns:**the location of the end of the range **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if the range is invalid

### getLineBuffer

protected final [Segment](http://docs.google.com/javax/swing/text/Segment.html) **getLineBuffer**()

Gives access to a buffer that can be used to fetch text from the associated document.

**Returns:**the buffer

### updateMetrics

protected void **updateMetrics**()

Checks to see if the font metrics and longest line are up-to-date.

**Since:** 1.4

### getPreferredSpan

public float **getPreferredSpan**(int axis)

Determines the preferred span for this view along an axis.

**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))

### paint

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

Renders using the given rendering surface and area on that surface. The view may need to do layout and create child views to enable itself to render into the given allocation.

**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 rendering surface to usea - the allocated region to render into**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))

### 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 document model coordinate space to the coordinate space of the view mapped to it.

**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 **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 fx,  
 float fy,  
 [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:**fx - the X coordinate >= 0fy - 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 >= 0**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))

### insertUpdate

public void **insertUpdate**([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) changes,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)

Gives notification that something was inserted into the document in a location that this view is responsible for.

**Overrides:**[insertUpdate](http://docs.google.com/javax/swing/text/View.html#insertUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**changes - the change information from the associated documenta - the current allocation of the viewf - the factory to use to rebuild if the view has children**See Also:**[View.insertUpdate(javax.swing.event.DocumentEvent, java.awt.Shape, javax.swing.text.ViewFactory)](http://docs.google.com/javax/swing/text/View.html#insertUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))

### removeUpdate

public void **removeUpdate**([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) changes,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)

Gives notification that something was removed from the document in a location that this view is responsible for.

**Overrides:**[removeUpdate](http://docs.google.com/javax/swing/text/View.html#removeUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**changes - the change information from the associated documenta - the current allocation of the viewf - the factory to use to rebuild if the view has children**See Also:**[View.removeUpdate(javax.swing.event.DocumentEvent, java.awt.Shape, javax.swing.text.ViewFactory)](http://docs.google.com/javax/swing/text/View.html#removeUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))

### changedUpdate

public void **changedUpdate**([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) changes,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)

Gives notification from the document that attributes were changed in a location that this view is responsible for.

**Overrides:**[changedUpdate](http://docs.google.com/javax/swing/text/View.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**changes - the change information from the associated documenta - the current allocation of the viewf - the factory to use to rebuild if the view has children**See Also:**[View.changedUpdate(javax.swing.event.DocumentEvent, java.awt.Shape, javax.swing.text.ViewFactory)](http://docs.google.com/javax/swing/text/View.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))

### setSize

public void **setSize**(float width,  
 float height)

Sets the size of the view. This should cause layout of the view along the given axis, if it has any layout duties.

**Overrides:**[setSize](http://docs.google.com/javax/swing/text/View.html#setSize(float,%20float)) in class [View](http://docs.google.com/javax/swing/text/View.html) **Parameters:**width - the width >= 0height - the height >= 0

### nextTabStop

public float **nextTabStop**(float x,  
 int tabOffset)

Returns the next tab stop position after a given reference position. This implementation does not support things like centering so it ignores the tabOffset argument.

**Specified by:**[nextTabStop](http://docs.google.com/javax/swing/text/TabExpander.html#nextTabStop(float,%20int)) in interface [TabExpander](http://docs.google.com/javax/swing/text/TabExpander.html) **Parameters:**x - the current position >= 0tabOffset - the position within the text stream that the tab occurred at >= 0. **Returns:**the tab stop, measured in points >= 0

### updateDamage

protected void **updateDamage**([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) changes,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)

Repaint the region of change covered by the given document event. Damages the line that begins the range to cover the case when the insert/remove is only on one line. If lines are added or removed, damages the whole view. The longest line is checked to see if it has changed.

**Since:** 1.4

### damageLineRange

protected void **damageLineRange**(int line0,  
 int line1,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [Component](http://docs.google.com/java/awt/Component.html) host)

Repaint the given line range.

**Parameters:**host - the component hosting the view (used to call repaint)a - the region allocated for the view to render intoline0 - the starting line number to repaint. This must be a valid line number in the model.line1 - the ending line number to repaint. This must be a valid line number in the model.**Since:** 1.4

### lineToRect

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **lineToRect**([Shape](http://docs.google.com/java/awt/Shape.html) a,  
 int line)

Determine the rectangle that represents the given line.

**Parameters:**a - the region allocated for the view to render intoline - the line number to find the region of. This must be a valid line number in the model.**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PlainView.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/PlainDocument.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/Position.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/PlainView.html)    [**NO FRAMES**](http://docs.google.com/PlainView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

[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).