| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ParagraphView.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/html/Option.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/StyleSheet.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/ParagraphView.html)    [**NO FRAMES**](http://docs.google.com/ParagraphView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | [FIELD](#3dy6vkm) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: FIELD | [CONSTR](#z337ya) | [METHOD](#1y810tw) |

## **javax.swing.text.html**

Class ParagraphView

[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.CompositeView](http://docs.google.com/javax/swing/text/CompositeView.html)  
 [javax.swing.text.BoxView](http://docs.google.com/javax/swing/text/BoxView.html)  
 [javax.swing.text.FlowView](http://docs.google.com/javax/swing/text/FlowView.html)  
 [javax.swing.text.ParagraphView](http://docs.google.com/javax/swing/text/ParagraphView.html)  
 **javax.swing.text.html.ParagraphView**

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

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

Displays the a paragraph, and uses css attributes for its configuration.

| **Nested Class Summary** | |
| --- | --- |

| **Nested classes/interfaces inherited from class javax.swing.text.**[**FlowView**](http://docs.google.com/javax/swing/text/FlowView.html) |
| --- |
| [FlowView.FlowStrategy](http://docs.google.com/javax/swing/text/FlowView.FlowStrategy.html) |

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

| **Fields inherited from class javax.swing.text.**[**ParagraphView**](http://docs.google.com/javax/swing/text/ParagraphView.html) |
| --- |
| [firstLineIndent](http://docs.google.com/javax/swing/text/ParagraphView.html#firstLineIndent) |

| **Fields inherited from class javax.swing.text.**[**FlowView**](http://docs.google.com/javax/swing/text/FlowView.html) |
| --- |
| [layoutPool](http://docs.google.com/javax/swing/text/FlowView.html#layoutPool), [layoutSpan](http://docs.google.com/javax/swing/text/FlowView.html#layoutSpan), [strategy](http://docs.google.com/javax/swing/text/FlowView.html#strategy) |

| **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** | |
| --- | --- |
| [**ParagraphView**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#ParagraphView(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Constructs a ParagraphView for the given element. |

| **Method Summary** | |
| --- | --- |
| protected  [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) | [**calculateMinorAxisRequirements**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#calculateMinorAxisRequirements(int,%20javax.swing.SizeRequirements))(int axis, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) r)            Calculate the needs for the paragraph along the minor axis. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**getAttributes**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#getAttributes())()            Fetches the attributes to use when rendering. |
| float | [**getMaximumSpan**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#getMaximumSpan(int))(int axis)            Determines the maximum span for this view along an axis. |
| float | [**getMinimumSpan**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#getMinimumSpan(int))(int axis)            Determines the minimum span for this view along an axis. |
| float | [**getPreferredSpan**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#getPreferredSpan(int))(int axis)            Determines the preferred span for this view. |
| protected  [StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) | [**getStyleSheet**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#getStyleSheet())() |
| boolean | [**isVisible**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#isVisible())()            Indicates whether or not this view should be displayed. |
| void | [**paint**](http://docs.google.com/javax/swing/text/html/ParagraphView.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 | [**setParent**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#setParent(javax.swing.text.View))([View](http://docs.google.com/javax/swing/text/View.html) parent)            Establishes the parent view for this view. |
| protected  void | [**setPropertiesFromAttributes**](http://docs.google.com/javax/swing/text/html/ParagraphView.html#setPropertiesFromAttributes())()            Sets up the paragraph from css attributes instead of the values found in StyleConstants (i.e. |

| **Methods inherited from class javax.swing.text.**[**ParagraphView**](http://docs.google.com/javax/swing/text/ParagraphView.html) |
| --- |
| [adjustRow](http://docs.google.com/javax/swing/text/ParagraphView.html#adjustRow(javax.swing.text.ParagraphView.Row,%20int,%20int)), [breakView](http://docs.google.com/javax/swing/text/ParagraphView.html#breakView(int,%20float,%20java.awt.Shape)), [changedUpdate](http://docs.google.com/javax/swing/text/ParagraphView.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [createRow](http://docs.google.com/javax/swing/text/ParagraphView.html#createRow()), [findOffsetToCharactersInString](http://docs.google.com/javax/swing/text/ParagraphView.html#findOffsetToCharactersInString(char%5B%5D,%20int)), [flipEastAndWestAtEnds](http://docs.google.com/javax/swing/text/ParagraphView.html#flipEastAndWestAtEnds(int,%20javax.swing.text.Position.Bias)), [getAlignment](http://docs.google.com/javax/swing/text/ParagraphView.html#getAlignment(int)), [getBreakWeight](http://docs.google.com/javax/swing/text/ParagraphView.html#getBreakWeight(int,%20float)), [getClosestPositionTo](http://docs.google.com/javax/swing/text/ParagraphView.html#getClosestPositionTo(int,%20javax.swing.text.Position.Bias,%20java.awt.Shape,%20int,%20javax.swing.text.Position.Bias%5B%5D,%20int,%20int)), [getFlowSpan](http://docs.google.com/javax/swing/text/ParagraphView.html#getFlowSpan(int)), [getFlowStart](http://docs.google.com/javax/swing/text/ParagraphView.html#getFlowStart(int)), [getLayoutView](http://docs.google.com/javax/swing/text/ParagraphView.html#getLayoutView(int)), [getLayoutViewCount](http://docs.google.com/javax/swing/text/ParagraphView.html#getLayoutViewCount()), [getNextNorthSouthVisualPositionFrom](http://docs.google.com/javax/swing/text/ParagraphView.html#getNextNorthSouthVisualPositionFrom(int,%20javax.swing.text.Position.Bias,%20java.awt.Shape,%20int,%20javax.swing.text.Position.Bias%5B%5D)), [getPartialSize](http://docs.google.com/javax/swing/text/ParagraphView.html#getPartialSize(int,%20int)), [getTabBase](http://docs.google.com/javax/swing/text/ParagraphView.html#getTabBase()), [getTabSet](http://docs.google.com/javax/swing/text/ParagraphView.html#getTabSet()), [nextTabStop](http://docs.google.com/javax/swing/text/ParagraphView.html#nextTabStop(float,%20int)), [setFirstLineIndent](http://docs.google.com/javax/swing/text/ParagraphView.html#setFirstLineIndent(float)), [setJustification](http://docs.google.com/javax/swing/text/ParagraphView.html#setJustification(int)), [setLineSpacing](http://docs.google.com/javax/swing/text/ParagraphView.html#setLineSpacing(float)) |

| **Methods inherited from class javax.swing.text.**[**FlowView**](http://docs.google.com/javax/swing/text/FlowView.html) |
| --- |
| [getFlowAxis](http://docs.google.com/javax/swing/text/FlowView.html#getFlowAxis()), [getViewIndexAtPosition](http://docs.google.com/javax/swing/text/FlowView.html#getViewIndexAtPosition(int)), [insertUpdate](http://docs.google.com/javax/swing/text/FlowView.html#insertUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [layout](http://docs.google.com/javax/swing/text/FlowView.html#layout(int,%20int)), [loadChildren](http://docs.google.com/javax/swing/text/FlowView.html#loadChildren(javax.swing.text.ViewFactory)), [removeUpdate](http://docs.google.com/javax/swing/text/FlowView.html#removeUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)) |

| **Methods inherited from class javax.swing.text.**[**BoxView**](http://docs.google.com/javax/swing/text/BoxView.html) |
| --- |
| [baselineLayout](http://docs.google.com/javax/swing/text/BoxView.html#baselineLayout(int,%20int,%20int%5B%5D,%20int%5B%5D)), [baselineRequirements](http://docs.google.com/javax/swing/text/BoxView.html#baselineRequirements(int,%20javax.swing.SizeRequirements)), [calculateMajorAxisRequirements](http://docs.google.com/javax/swing/text/BoxView.html#calculateMajorAxisRequirements(int,%20javax.swing.SizeRequirements)), [childAllocation](http://docs.google.com/javax/swing/text/BoxView.html#childAllocation(int,%20java.awt.Rectangle)), [forwardUpdate](http://docs.google.com/javax/swing/text/BoxView.html#forwardUpdate(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [getAxis](http://docs.google.com/javax/swing/text/BoxView.html#getAxis()), [getChildAllocation](http://docs.google.com/javax/swing/text/BoxView.html#getChildAllocation(int,%20java.awt.Shape)), [getHeight](http://docs.google.com/javax/swing/text/BoxView.html#getHeight()), [getOffset](http://docs.google.com/javax/swing/text/BoxView.html#getOffset(int,%20int)), [getResizeWeight](http://docs.google.com/javax/swing/text/BoxView.html#getResizeWeight(int)), [getSpan](http://docs.google.com/javax/swing/text/BoxView.html#getSpan(int,%20int)), [getViewAtPoint](http://docs.google.com/javax/swing/text/BoxView.html#getViewAtPoint(int,%20int,%20java.awt.Rectangle)), [getWidth](http://docs.google.com/javax/swing/text/BoxView.html#getWidth()), [isAfter](http://docs.google.com/javax/swing/text/BoxView.html#isAfter(int,%20int,%20java.awt.Rectangle)), [isAllocationValid](http://docs.google.com/javax/swing/text/BoxView.html#isAllocationValid()), [isBefore](http://docs.google.com/javax/swing/text/BoxView.html#isBefore(int,%20int,%20java.awt.Rectangle)), [isLayoutValid](http://docs.google.com/javax/swing/text/BoxView.html#isLayoutValid(int)), [layoutChanged](http://docs.google.com/javax/swing/text/BoxView.html#layoutChanged(int)), [layoutMajorAxis](http://docs.google.com/javax/swing/text/BoxView.html#layoutMajorAxis(int,%20int,%20int%5B%5D,%20int%5B%5D)), [layoutMinorAxis](http://docs.google.com/javax/swing/text/BoxView.html#layoutMinorAxis(int,%20int,%20int%5B%5D,%20int%5B%5D)), [modelToView](http://docs.google.com/javax/swing/text/BoxView.html#modelToView(int,%20java.awt.Shape,%20javax.swing.text.Position.Bias)), [paintChild](http://docs.google.com/javax/swing/text/BoxView.html#paintChild(java.awt.Graphics,%20java.awt.Rectangle,%20int)), [preferenceChanged](http://docs.google.com/javax/swing/text/BoxView.html#preferenceChanged(javax.swing.text.View,%20boolean,%20boolean)), [replace](http://docs.google.com/javax/swing/text/BoxView.html#replace(int,%20int,%20javax.swing.text.View%5B%5D)), [setAxis](http://docs.google.com/javax/swing/text/BoxView.html#setAxis(int)), [setSize](http://docs.google.com/javax/swing/text/BoxView.html#setSize(float,%20float)), [viewToModel](http://docs.google.com/javax/swing/text/BoxView.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D)) |

| **Methods inherited from class javax.swing.text.**[**CompositeView**](http://docs.google.com/javax/swing/text/CompositeView.html) |
| --- |
| [getBottomInset](http://docs.google.com/javax/swing/text/CompositeView.html#getBottomInset()), [getInsideAllocation](http://docs.google.com/javax/swing/text/CompositeView.html#getInsideAllocation(java.awt.Shape)), [getLeftInset](http://docs.google.com/javax/swing/text/CompositeView.html#getLeftInset()), [getNextEastWestVisualPositionFrom](http://docs.google.com/javax/swing/text/CompositeView.html#getNextEastWestVisualPositionFrom(int,%20javax.swing.text.Position.Bias,%20java.awt.Shape,%20int,%20javax.swing.text.Position.Bias%5B%5D)), [getNextVisualPositionFrom](http://docs.google.com/javax/swing/text/CompositeView.html#getNextVisualPositionFrom(int,%20javax.swing.text.Position.Bias,%20java.awt.Shape,%20int,%20javax.swing.text.Position.Bias%5B%5D)), [getRightInset](http://docs.google.com/javax/swing/text/CompositeView.html#getRightInset()), [getTopInset](http://docs.google.com/javax/swing/text/CompositeView.html#getTopInset()), [getView](http://docs.google.com/javax/swing/text/CompositeView.html#getView(int)), [getViewAtPosition](http://docs.google.com/javax/swing/text/CompositeView.html#getViewAtPosition(int,%20java.awt.Rectangle)), [getViewCount](http://docs.google.com/javax/swing/text/CompositeView.html#getViewCount()), [getViewIndex](http://docs.google.com/javax/swing/text/CompositeView.html#getViewIndex(int,%20javax.swing.text.Position.Bias)), [modelToView](http://docs.google.com/javax/swing/text/CompositeView.html#modelToView(int,%20javax.swing.text.Position.Bias,%20int,%20javax.swing.text.Position.Bias,%20java.awt.Shape)), [setInsets](http://docs.google.com/javax/swing/text/CompositeView.html#setInsets(short,%20short,%20short,%20short)), [setParagraphInsets](http://docs.google.com/javax/swing/text/CompositeView.html#setParagraphInsets(javax.swing.text.AttributeSet)) |

| **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)), [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)), [getBreakWeight](http://docs.google.com/javax/swing/text/View.html#getBreakWeight(int,%20float,%20float)), [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()), [getParent](http://docs.google.com/javax/swing/text/View.html#getParent()), [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)), [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)), [insert](http://docs.google.com/javax/swing/text/View.html#insert(int,%20javax.swing.text.View)), [modelToView](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20java.awt.Shape)), [remove](http://docs.google.com/javax/swing/text/View.html#remove(int)), [removeAll](http://docs.google.com/javax/swing/text/View.html#removeAll()), [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** |
| --- |

### ParagraphView

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

Constructs a ParagraphView for the given element.

**Parameters:**elem - the element that this view is responsible for

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

### setParent

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

Establishes the parent view for this view. This is guaranteed to be called before any other methods if the parent view is functioning properly.

This is implemented to forward to the superclass as well as call the [setPropertiesFromAttributes](#ihv636) method to set the paragraph properties from the css attributes. The call is made at this time to ensure the ability to resolve upward through the parents view attributes.

**Overrides:**[setParent](http://docs.google.com/javax/swing/text/FlowView.html#setParent(javax.swing.text.View)) in class [FlowView](http://docs.google.com/javax/swing/text/FlowView.html) **Parameters:**parent - the new parent, or null if the view is being removed from a parent it was previously added to

### getAttributes

public [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) **getAttributes**()

Fetches the attributes to use when rendering. This is implemented to multiplex the attributes specified in the model with a StyleSheet.

**Overrides:**[getAttributes](http://docs.google.com/javax/swing/text/View.html#getAttributes()) in class [View](http://docs.google.com/javax/swing/text/View.html)

### setPropertiesFromAttributes

protected void **setPropertiesFromAttributes**()

Sets up the paragraph from css attributes instead of the values found in StyleConstants (i.e. which are used by the superclass). Since

**Overrides:**[setPropertiesFromAttributes](http://docs.google.com/javax/swing/text/ParagraphView.html#setPropertiesFromAttributes()) in class [ParagraphView](http://docs.google.com/javax/swing/text/ParagraphView.html)

### getStyleSheet

protected [StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) **getStyleSheet**()

### calculateMinorAxisRequirements

protected [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) **calculateMinorAxisRequirements**(int axis,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) r)

Calculate the needs for the paragraph along the minor axis. This implemented to use the requirements of the superclass, modified slightly to set a minimum span allowed. Typical html rendering doesn't let the view size shrink smaller than the length of the longest word.

**Overrides:**[calculateMinorAxisRequirements](http://docs.google.com/javax/swing/text/FlowView.html#calculateMinorAxisRequirements(int,%20javax.swing.SizeRequirements)) in class [FlowView](http://docs.google.com/javax/swing/text/FlowView.html) **Parameters:**axis - the axis being studiedr - the SizeRequirements object; if null one will be created **Returns:**the newly initialized SizeRequirements object**See Also:**[SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)

### isVisible

public boolean **isVisible**()

Indicates whether or not this view should be displayed. If none of the children wish to be displayed and the only visible child is the break that ends the paragraph, the paragraph will not be considered visible. Otherwise, it will be considered visible and return true.

**Overrides:**[isVisible](http://docs.google.com/javax/swing/text/View.html#isVisible()) in class [View](http://docs.google.com/javax/swing/text/View.html) **Returns:**true if the paragraph should be displayed

### 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. This is implemented to delgate to the superclass after stashing the base coordinate for tab calculations.

**Overrides:**[paint](http://docs.google.com/javax/swing/text/ParagraphView.html#paint(java.awt.Graphics,%20java.awt.Shape)) in class [ParagraphView](http://docs.google.com/javax/swing/text/ParagraphView.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))

### getPreferredSpan

public float **getPreferredSpan**(int axis)

Determines the preferred span for this view. Returns 0 if the view is not visible, otherwise it calls the superclass method to get the preferred span. axis.

**Overrides:**[getPreferredSpan](http://docs.google.com/javax/swing/text/BoxView.html#getPreferredSpan(int)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**axis - may be either View.X\_AXIS or View.Y\_AXIS **Returns:**the span the view would like to be rendered into; 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**See Also:**[BoxView.getPreferredSpan(int)](http://docs.google.com/javax/swing/text/BoxView.html#getPreferredSpan(int))

### getMinimumSpan

public float **getMinimumSpan**(int axis)

Determines the minimum span for this view along an axis. Returns 0 if the view is not visible, otherwise it calls the superclass method to get the minimum span.

**Overrides:**[getMinimumSpan](http://docs.google.com/javax/swing/text/BoxView.html#getMinimumSpan(int)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**axis - may be either View.X\_AXIS or View.Y\_AXIS **Returns:**the minimum span the view can be rendered into**See Also:**[BoxView.getMinimumSpan(int)](http://docs.google.com/javax/swing/text/BoxView.html#getMinimumSpan(int))

### getMaximumSpan

public float **getMaximumSpan**(int axis)

Determines the maximum span for this view along an axis. Returns 0 if the view is not visible, otherwise it calls the superclass method ot get the maximum span.

**Overrides:**[getMaximumSpan](http://docs.google.com/javax/swing/text/BoxView.html#getMaximumSpan(int)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**axis - may be either View.X\_AXIS or View.Y\_AXIS **Returns:**the maximum span the view can be rendered into**See Also:**[BoxView.getMaximumSpan(int)](http://docs.google.com/javax/swing/text/BoxView.html#getMaximumSpan(int))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ParagraphView.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/html/Option.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/StyleSheet.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/ParagraphView.html)    [**NO FRAMES**](http://docs.google.com/ParagraphView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | [FIELD](#3dy6vkm) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: FIELD | [CONSTR](#z337ya) | [METHOD](#1y810tw) |

[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).