| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BlockView.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   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/html/CSS.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/html/BlockView.html)    [**NO FRAMES**](http://docs.google.com/BlockView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: FIELD | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

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

Class BlockView

[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.html.BlockView**

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

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

A view implementation to display a block (as a box) with CSS specifications.

| **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** | |
| --- | --- |
| [**BlockView**](http://docs.google.com/javax/swing/text/html/BlockView.html#BlockView(javax.swing.text.Element,%20int))([Element](http://docs.google.com/javax/swing/text/Element.html) elem, int axis)            Creates a new view that represents an html box. |

| **Method Summary** | |
| --- | --- |
| protected  [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) | [**calculateMajorAxisRequirements**](http://docs.google.com/javax/swing/text/html/BlockView.html#calculateMajorAxisRequirements(int,%20javax.swing.SizeRequirements))(int axis, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) r)            Calculate the requirements of the block along the major axis (i.e. |
| protected  [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) | [**calculateMinorAxisRequirements**](http://docs.google.com/javax/swing/text/html/BlockView.html#calculateMinorAxisRequirements(int,%20javax.swing.SizeRequirements))(int axis, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) r)            Calculate the requirements of the block along the minor axis (i.e. |
| void | [**changedUpdate**](http://docs.google.com/javax/swing/text/html/BlockView.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. |
| float | [**getAlignment**](http://docs.google.com/javax/swing/text/html/BlockView.html#getAlignment(int))(int axis)            Gets the alignment. |
| [AttributeSet](http://docs.google.com/javax/swing/text/AttributeSet.html) | [**getAttributes**](http://docs.google.com/javax/swing/text/html/BlockView.html#getAttributes())()            Fetches the attributes to use when rendering. |
| float | [**getMaximumSpan**](http://docs.google.com/javax/swing/text/html/BlockView.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/BlockView.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/BlockView.html#getPreferredSpan(int))(int axis)            Determines the preferred span for this view along an axis. |
| int | [**getResizeWeight**](http://docs.google.com/javax/swing/text/html/BlockView.html#getResizeWeight(int))(int axis)            Gets the resize weight. |
| protected  [StyleSheet](http://docs.google.com/javax/swing/text/html/StyleSheet.html) | [**getStyleSheet**](http://docs.google.com/javax/swing/text/html/BlockView.html#getStyleSheet())() |
| protected  void | [**layoutMinorAxis**](http://docs.google.com/javax/swing/text/html/BlockView.html#layoutMinorAxis(int,%20int,%20int%5B%5D,%20int%5B%5D))(int targetSpan, int axis, int[] offsets, int[] spans)            Performs layout for the minor axis of the box (i.e. |
| void | [**paint**](http://docs.google.com/javax/swing/text/html/BlockView.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) allocation)            Renders using the given rendering surface and area on that surface. |
| void | [**setParent**](http://docs.google.com/javax/swing/text/html/BlockView.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/BlockView.html#setPropertiesFromAttributes())()            Update any cached values that come from attributes. |

| **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)), [childAllocation](http://docs.google.com/javax/swing/text/BoxView.html#childAllocation(int,%20java.awt.Rectangle)), [flipEastAndWestAtEnds](http://docs.google.com/javax/swing/text/BoxView.html#flipEastAndWestAtEnds(int,%20javax.swing.text.Position.Bias)), [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)), [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)), [layout](http://docs.google.com/javax/swing/text/BoxView.html#layout(int,%20int)), [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)), [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)), [getNextNorthSouthVisualPositionFrom](http://docs.google.com/javax/swing/text/CompositeView.html#getNextNorthSouthVisualPositionFrom(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)), [getViewIndexAtPosition](http://docs.google.com/javax/swing/text/CompositeView.html#getViewIndexAtPosition(int)), [loadChildren](http://docs.google.com/javax/swing/text/CompositeView.html#loadChildren(javax.swing.text.ViewFactory)), [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)), [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,%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()), [removeUpdate](http://docs.google.com/javax/swing/text/View.html#removeUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [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** |
| --- |

### BlockView

public **BlockView**([Element](http://docs.google.com/javax/swing/text/Element.html) elem,  
 int axis)

Creates a new view that represents an html box. This can be used for a number of elements.

**Parameters:**elem - the element to create a view foraxis - either View.X\_AXIS or View.Y\_AXIS

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

### calculateMajorAxisRequirements

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

Calculate the requirements of the block along the major axis (i.e. the axis along with it tiles). This is implemented to provide the superclass behavior and then adjust it if the CSS width or height attribute is specified and applicable to the axis.

**Overrides:**[calculateMajorAxisRequirements](http://docs.google.com/javax/swing/text/BoxView.html#calculateMajorAxisRequirements(int,%20javax.swing.SizeRequirements)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.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)

### 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 requirements of the block along the minor axis (i.e. the axis orthoginal to the axis along with it tiles). This is implemented to provide the superclass behavior and then adjust it if the CSS width or height attribute is specified and applicable to the axis.

**Overrides:**[calculateMinorAxisRequirements](http://docs.google.com/javax/swing/text/BoxView.html#calculateMinorAxisRequirements(int,%20javax.swing.SizeRequirements)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.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)

### layoutMinorAxis

protected void **layoutMinorAxis**(int targetSpan,  
 int axis,  
 int[] offsets,  
 int[] spans)

Performs layout for the minor axis of the box (i.e. the axis orthoginal to the axis that it represents). The results of the layout (the offset and span for each children) are placed in the given arrays which represent the allocations to the children along the minor axis.

**Overrides:**[layoutMinorAxis](http://docs.google.com/javax/swing/text/BoxView.html#layoutMinorAxis(int,%20int,%20int%5B%5D,%20int%5B%5D)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**targetSpan - the total span given to the view, which whould be used to layout the childre.axis - the axis being layed outoffsets - the offsets from the origin of the view for each of the child views; this is a return value and is filled in by the implementation of this methodspans - the span of each child view; this is a return value and is filled in by the implementation of this method

### paint

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

Renders using the given rendering surface and area on that surface. This is implemented to delegate to the css box painter to paint the border and background prior to the interior.

**Overrides:**[paint](http://docs.google.com/javax/swing/text/BoxView.html#paint(java.awt.Graphics,%20java.awt.Shape)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**g - the rendering surface to useallocation - 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))

### 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)

### getResizeWeight

public int **getResizeWeight**(int axis)

Gets the resize weight.

**Overrides:**[getResizeWeight](http://docs.google.com/javax/swing/text/BoxView.html#getResizeWeight(int)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**axis - may be either X\_AXIS or Y\_AXIS **Returns:**the weight **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - for an invalid axis

### getAlignment

public float **getAlignment**(int axis)

Gets the alignment.

**Overrides:**[getAlignment](http://docs.google.com/javax/swing/text/BoxView.html#getAlignment(int)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**axis - may be either X\_AXIS or Y\_AXIS **Returns:**the alignment

### 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)

**Description copied from class:** [**View**](http://docs.google.com/javax/swing/text/View.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)) Gives notification from the document that attributes were changed in a location that this view is responsible for. To reduce the burden to subclasses, this functionality is spread out into the following calls that subclasses can reimplement:

1. [updateChildren](#3o7alnk) is called if there were any changes to the element this view is responsible for. If this view has child views that are represent the child elements, then this method should do whatever is necessary to make sure the child views correctly represent the model.
2. [forwardUpdate](#23ckvvd) is called to forward the DocumentEvent to the appropriate child views.
3. [updateLayout](#ihv636) is called to give the view a chance to either repair its layout, to reschedule layout, or do nothing.

**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))

### getPreferredSpan

public float **getPreferredSpan**(int axis)

Determines the preferred span for this view along an 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 >= 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 type**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.

**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 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 type**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.

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

### setPropertiesFromAttributes

protected void **setPropertiesFromAttributes**()

Update any cached values that come from attributes.

### getStyleSheet

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

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