| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TableView.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/TabExpander.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/TableView.TableCell.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/TableView.html)    [**NO FRAMES**](http://docs.google.com/TableView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

## **javax.swing.text**

Class TableView

[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.TableView**

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

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

Implements View interface for a table, that is composed of an element structure where the child elements of the element this view is responsible for represent rows and the child elements of the row elements are cells. The cell elements can have an arbitrary element structure under them, which will be built with the ViewFactory returned by the getViewFactory method.

  TABLE  
   ROW  
   CELL  
   CELL  
   ROW  
   CELL  
   CELL

This is implemented as a hierarchy of boxes, the table itself is a vertical box, the rows are horizontal boxes, and the cells are vertical boxes. The cells are allowed to span multiple columns and rows. By default, the table can be thought of as being formed over a grid (i.e. somewhat like one would find in gridbag layout), where table cells can request to span more than one grid cell. The default horizontal span of table cells will be based upon this grid, but can be changed by reimplementing the requested span of the cell (i.e. table cells can have independant spans if desired).

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

| **Nested Class Summary** | |
| --- | --- |
| class | [**TableView.TableCell**](http://docs.google.com/javax/swing/text/TableView.TableCell.html)  **Deprecated.** *A table cell can now be any View implementation.* |
| class | [**TableView.TableRow**](http://docs.google.com/javax/swing/text/TableView.TableRow.html)            View of a row in a row-centric table. |

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

| **Method Summary** | |
| --- | --- |
| protected  [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) | [**calculateMinorAxisRequirements**](http://docs.google.com/javax/swing/text/TableView.html#calculateMinorAxisRequirements(int,%20javax.swing.SizeRequirements))(int axis, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) r)            Calculate the requirements for the minor axis. |
| protected  [TableView.TableCell](http://docs.google.com/javax/swing/text/TableView.TableCell.html) | [**createTableCell**](http://docs.google.com/javax/swing/text/TableView.html#createTableCell(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)  **Deprecated.** *Table cells can now be any arbitrary View implementation and should be produced by the ViewFactory rather than the table.* |
| protected  [TableView.TableRow](http://docs.google.com/javax/swing/text/TableView.TableRow.html) | [**createTableRow**](http://docs.google.com/javax/swing/text/TableView.html#createTableRow(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Creates a new table row. |
| protected  void | [**forwardUpdate**](http://docs.google.com/javax/swing/text/TableView.html#forwardUpdate(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))([DocumentEvent.ElementChange](http://docs.google.com/javax/swing/event/DocumentEvent.ElementChange.html) ec, [DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e, [Shape](http://docs.google.com/java/awt/Shape.html) a, [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)            Forwards the given DocumentEvent to the child views that need to be notified of the change to the model. |
| protected  [View](http://docs.google.com/javax/swing/text/View.html) | [**getViewAtPosition**](http://docs.google.com/javax/swing/text/TableView.html#getViewAtPosition(int,%20java.awt.Rectangle))(int pos, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) a)            Fetches the child view that represents the given position in the model. |
| protected  void | [**layoutColumns**](http://docs.google.com/javax/swing/text/TableView.html#layoutColumns(int,%20int%5B%5D,%20int%5B%5D,%20javax.swing.SizeRequirements%5B%5D))(int targetSpan, int[] offsets, int[] spans, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] reqs)            Lays out the columns to fit within the given target span. |
| protected  void | [**layoutMinorAxis**](http://docs.google.com/javax/swing/text/TableView.html#layoutMinorAxis(int,%20int,%20int%5B%5D,%20int%5B%5D))(int targetSpan, int axis, int[] offsets, int[] spans)            Perform layout for the minor axis of the box (i.e. |
| void | [**replace**](http://docs.google.com/javax/swing/text/TableView.html#replace(int,%20int,%20javax.swing.text.View%5B%5D))(int offset, int length, [View](http://docs.google.com/javax/swing/text/View.html)[] views)            Change the child views. |

| **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)), [flipEastAndWestAtEnds](http://docs.google.com/javax/swing/text/BoxView.html#flipEastAndWestAtEnds(int,%20javax.swing.text.Position.Bias)), [getAlignment](http://docs.google.com/javax/swing/text/BoxView.html#getAlignment(int)), [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()), [getMaximumSpan](http://docs.google.com/javax/swing/text/BoxView.html#getMaximumSpan(int)), [getMinimumSpan](http://docs.google.com/javax/swing/text/BoxView.html#getMinimumSpan(int)), [getOffset](http://docs.google.com/javax/swing/text/BoxView.html#getOffset(int,%20int)), [getPreferredSpan](http://docs.google.com/javax/swing/text/BoxView.html#getPreferredSpan(int)), [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)), [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)), [paint](http://docs.google.com/javax/swing/text/BoxView.html#paint(java.awt.Graphics,%20java.awt.Shape)), [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)), [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)), [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)), [setParent](http://docs.google.com/javax/swing/text/CompositeView.html#setParent(javax.swing.text.View)) |

| **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)), [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)), [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** |
| --- |

### TableView

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

Constructs a TableView for the given element.

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

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

### createTableRow

protected [TableView.TableRow](http://docs.google.com/javax/swing/text/TableView.TableRow.html) **createTableRow**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)

Creates a new table row.

**Parameters:**elem - an element **Returns:**the row

### createTableCell

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected [TableView.TableCell](http://docs.google.com/javax/swing/text/TableView.TableCell.html) **createTableCell**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)

**Deprecated.** *Table cells can now be any arbitrary View implementation and should be produced by the ViewFactory rather than the table.*

**Parameters:**elem - an element **Returns:**the cell

### forwardUpdate

protected void **forwardUpdate**([DocumentEvent.ElementChange](http://docs.google.com/javax/swing/event/DocumentEvent.ElementChange.html) ec,  
 [DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e,  
 [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:** [**BoxView**](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)) Forwards the given DocumentEvent to the child views that need to be notified of the change to the model. If a child changed its requirements and the allocation was valid prior to forwarding the portion of the box from the starting child to the end of the box will be repainted.

**Overrides:**[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)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**ec - changes to the element this view is responsible for (may be null if there were no changes)e - 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)), [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)), [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))

### replace

public void **replace**(int offset,  
 int length,  
 [View](http://docs.google.com/javax/swing/text/View.html)[] views)

Change the child views. This is implemented to provide the superclass behavior and invalidate the grid so that rows and columns will be recalculated.

**Overrides:**[replace](http://docs.google.com/javax/swing/text/BoxView.html#replace(int,%20int,%20javax.swing.text.View%5B%5D)) in class [BoxView](http://docs.google.com/javax/swing/text/BoxView.html) **Parameters:**offset - the starting index into the child views to insert the new views; this should be a value >= 0 and <= getViewCountlength - the number of existing child views to remove; This should be a value >= 0 and <= (getViewCount() - offset)views - the child views to add; this value can be nullto indicate no children are being added (useful to remove)

### layoutColumns

protected void **layoutColumns**(int targetSpan,  
 int[] offsets,  
 int[] spans,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] reqs)

Lays out the columns to fit within the given target span. Returns the results through offsets and spans.

**Parameters:**targetSpan - the given span for total of all the table columnsreqs - the requirements desired for each column. This is the column maximum of the cells minimum, preferred, and maximum requested spanspans - the return value of how much to allocated to each columnoffsets - the return value of the offset from the origin for each column

### layoutMinorAxis

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

Perform layout for the minor axis of the box (i.e. the axis orthoginal to the axis that it represents). The results of the layout should be placed in the given arrays which represent the allocations to the children along the minor axis. This is called by the superclass whenever the layout needs to be updated along the minor axis.

This is implemented to call the [layoutColumns](#3as4poj) method, and then forward to the superclass to actually carry out the layout of the tables rows.

**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 children.axis - the axis being layed out.offsets - 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 method.spans - the span of each child view. This is a return value and is filled in by the implementation of this method.

### 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 for the minor axis. This is called by the superclass whenever the requirements need to be updated (i.e. a preferenceChanged was messaged through this view).

This is implemented to calculate the requirements as the sum of the requirements of the columns.

**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)

### getViewAtPosition

protected [View](http://docs.google.com/javax/swing/text/View.html) **getViewAtPosition**(int pos,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) a)

Fetches the child view that represents the given position in the model. This is implemented to walk through the children looking for a range that contains the given position. In this view the children do not necessarily have a one to one mapping with the child elements.

**Overrides:**[getViewAtPosition](http://docs.google.com/javax/swing/text/CompositeView.html#getViewAtPosition(int,%20java.awt.Rectangle)) in class [CompositeView](http://docs.google.com/javax/swing/text/CompositeView.html) **Parameters:**pos - the search position >= 0a - the allocation to the table on entry, and the allocation of the view containing the position on exit **Returns:**the view representing the given position, or null if there isn't one

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TableView.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/TabExpander.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/TableView.TableCell.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/TableView.html)    [**NO FRAMES**](http://docs.google.com/TableView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

[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).