| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JTableHeader.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/table/DefaultTableModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/table/JTableHeader.AccessibleJTableHeader.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/table/JTableHeader.html)    [**NO FRAMES**](http://docs.google.com/JTableHeader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#qsh70q) | [METHOD](#49x2ik5) |

## **javax.swing.table**

Class JTableHeader

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.Component](http://docs.google.com/java/awt/Component.html)  
 [java.awt.Container](http://docs.google.com/java/awt/Container.html)  
 [javax.swing.JComponent](http://docs.google.com/javax/swing/JComponent.html)  
 **javax.swing.table.JTableHeader**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [EventListener](http://docs.google.com/java/util/EventListener.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html), [TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html)

public class **JTableHeader**extends [JComponent](http://docs.google.com/javax/swing/JComponent.html)implements [TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

This is the object which manages the header of the JTable.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

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

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JTableHeader.AccessibleJTableHeader**](http://docs.google.com/javax/swing/table/JTableHeader.AccessibleJTableHeader.html)            This class implements accessibility support for the JTableHeader class. |

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

| **Nested classes/interfaces inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [Container.AccessibleAWTContainer](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [Component.AccessibleAWTComponent](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html), [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html), [Component.BltBufferStrategy](http://docs.google.com/java/awt/Component.BltBufferStrategy.html), [Component.FlipBufferStrategy](http://docs.google.com/java/awt/Component.FlipBufferStrategy.html) |

| **Field Summary** | |
| --- | --- |
| protected  [TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) | [**columnModel**](http://docs.google.com/javax/swing/table/JTableHeader.html#columnModel)            The TableColumnModel of the table header. |
| protected  [TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) | [**draggedColumn**](http://docs.google.com/javax/swing/table/JTableHeader.html#draggedColumn)            The index of the column being dragged. |
| protected  int | [**draggedDistance**](http://docs.google.com/javax/swing/table/JTableHeader.html#draggedDistance)            The distance from its original position the column has been dragged. |
| protected  boolean | [**reorderingAllowed**](http://docs.google.com/javax/swing/table/JTableHeader.html#reorderingAllowed)            If true, reordering of columns are allowed by the user; the default is true. |
| protected  boolean | [**resizingAllowed**](http://docs.google.com/javax/swing/table/JTableHeader.html#resizingAllowed)            If true, resizing of columns are allowed by the user; the default is true. |
| protected  [TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) | [**resizingColumn**](http://docs.google.com/javax/swing/table/JTableHeader.html#resizingColumn)            The index of the column being resized. |
| protected  [JTable](http://docs.google.com/javax/swing/JTable.html) | [**table**](http://docs.google.com/javax/swing/table/JTableHeader.html#table)            The table for which this object is the header; the default is null. |
| protected  boolean | [**updateTableInRealTime**](http://docs.google.com/javax/swing/table/JTableHeader.html#updateTableInRealTime)            Obsolete as of Java 2 platform v1.3. |

| **Fields inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [accessibleContext](http://docs.google.com/javax/swing/JComponent.html#accessibleContext), [listenerList](http://docs.google.com/javax/swing/JComponent.html#listenerList), [TOOL\_TIP\_TEXT\_KEY](http://docs.google.com/javax/swing/JComponent.html#TOOL_TIP_TEXT_KEY), [ui](http://docs.google.com/javax/swing/JComponent.html#ui), [UNDEFINED\_CONDITION](http://docs.google.com/javax/swing/JComponent.html#UNDEFINED_CONDITION), [WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT](http://docs.google.com/javax/swing/JComponent.html#WHEN_ANCESTOR_OF_FOCUSED_COMPONENT), [WHEN\_FOCUSED](http://docs.google.com/javax/swing/JComponent.html#WHEN_FOCUSED), [WHEN\_IN\_FOCUSED\_WINDOW](http://docs.google.com/javax/swing/JComponent.html#WHEN_IN_FOCUSED_WINDOW) |

| **Fields inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [BOTTOM\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#BOTTOM_ALIGNMENT), [CENTER\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#CENTER_ALIGNMENT), [LEFT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#LEFT_ALIGNMENT), [RIGHT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#RIGHT_ALIGNMENT), [TOP\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#TOP_ALIGNMENT) |

| **Fields inherited from interface java.awt.image.**[**ImageObserver**](http://docs.google.com/java/awt/image/ImageObserver.html) |
| --- |
| [ABORT](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT), [ALLBITS](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS), [ERROR](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR), [FRAMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS), [HEIGHT](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT), [PROPERTIES](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES), [SOMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS), [WIDTH](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH) |

| **Constructor Summary** | |
| --- | --- |
| [**JTableHeader**](http://docs.google.com/javax/swing/table/JTableHeader.html#JTableHeader())()            Constructs a JTableHeader with a default TableColumnModel. |
| [**JTableHeader**](http://docs.google.com/javax/swing/table/JTableHeader.html#JTableHeader(javax.swing.table.TableColumnModel))([TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) cm)            Constructs a JTableHeader which is initialized with cm as the column model. |

| **Method Summary** | |
| --- | --- |
| void | [**columnAdded**](http://docs.google.com/javax/swing/table/JTableHeader.html#columnAdded(javax.swing.event.TableColumnModelEvent))([TableColumnModelEvent](http://docs.google.com/javax/swing/event/TableColumnModelEvent.html) e)            Invoked when a column is added to the table column model. |
| int | [**columnAtPoint**](http://docs.google.com/javax/swing/table/JTableHeader.html#columnAtPoint(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) point)            Returns the index of the column that point lies in, or -1 if it lies out of bounds. |
| void | [**columnMarginChanged**](http://docs.google.com/javax/swing/table/JTableHeader.html#columnMarginChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when a column is moved due to a margin change. |
| void | [**columnMoved**](http://docs.google.com/javax/swing/table/JTableHeader.html#columnMoved(javax.swing.event.TableColumnModelEvent))([TableColumnModelEvent](http://docs.google.com/javax/swing/event/TableColumnModelEvent.html) e)            Invoked when a column is repositioned. |
| void | [**columnRemoved**](http://docs.google.com/javax/swing/table/JTableHeader.html#columnRemoved(javax.swing.event.TableColumnModelEvent))([TableColumnModelEvent](http://docs.google.com/javax/swing/event/TableColumnModelEvent.html) e)            Invoked when a column is removed from the table column model. |
| void | [**columnSelectionChanged**](http://docs.google.com/javax/swing/table/JTableHeader.html#columnSelectionChanged(javax.swing.event.ListSelectionEvent))([ListSelectionEvent](http://docs.google.com/javax/swing/event/ListSelectionEvent.html) e)            Invoked when the selection model of the TableColumnModel is changed. |
| protected  [TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) | [**createDefaultColumnModel**](http://docs.google.com/javax/swing/table/JTableHeader.html#createDefaultColumnModel())()            Returns the default column model object which is a DefaultTableColumnModel. |
| protected  [TableCellRenderer](http://docs.google.com/javax/swing/table/TableCellRenderer.html) | [**createDefaultRenderer**](http://docs.google.com/javax/swing/table/JTableHeader.html#createDefaultRenderer())()            Returns a default renderer to be used when no header renderer is defined by a TableColumn. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/table/JTableHeader.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JTableHeader. |
| [TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) | [**getColumnModel**](http://docs.google.com/javax/swing/table/JTableHeader.html#getColumnModel())()            Returns the TableColumnModel that contains all column information of this table header. |
| [TableCellRenderer](http://docs.google.com/javax/swing/table/TableCellRenderer.html) | [**getDefaultRenderer**](http://docs.google.com/javax/swing/table/JTableHeader.html#getDefaultRenderer())()            Returns the default renderer used when no headerRenderer is defined by a TableColumn. |
| [TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) | [**getDraggedColumn**](http://docs.google.com/javax/swing/table/JTableHeader.html#getDraggedColumn())()            Returns the the dragged column, if and only if, a drag is in process, otherwise returns null. |
| int | [**getDraggedDistance**](http://docs.google.com/javax/swing/table/JTableHeader.html#getDraggedDistance())()            Returns the column's horizontal distance from its original position, if and only if, a drag is in process. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getHeaderRect**](http://docs.google.com/javax/swing/table/JTableHeader.html#getHeaderRect(int))(int column)            Returns the rectangle containing the header tile at column. |
| boolean | [**getReorderingAllowed**](http://docs.google.com/javax/swing/table/JTableHeader.html#getReorderingAllowed())()            Returns true if the user is allowed to rearrange columns by dragging their headers, false otherwise. |
| boolean | [**getResizingAllowed**](http://docs.google.com/javax/swing/table/JTableHeader.html#getResizingAllowed())()            Returns true if the user is allowed to resize columns by dragging between their headers, false otherwise. |
| [TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) | [**getResizingColumn**](http://docs.google.com/javax/swing/table/JTableHeader.html#getResizingColumn())()            Returns the resizing column. |
| [JTable](http://docs.google.com/javax/swing/JTable.html) | [**getTable**](http://docs.google.com/javax/swing/table/JTableHeader.html#getTable())()            Returns the table associated with this header. |
| [String](http://docs.google.com/java/lang/String.html) | [**getToolTipText**](http://docs.google.com/javax/swing/table/JTableHeader.html#getToolTipText(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)            Allows the renderer's tips to be used if there is text set. |
| [TableHeaderUI](http://docs.google.com/javax/swing/plaf/TableHeaderUI.html) | [**getUI**](http://docs.google.com/javax/swing/table/JTableHeader.html#getUI())()            Returns the look and feel (L&F) object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/table/JTableHeader.html#getUIClassID())()            Returns the suffix used to construct the name of the look and feel (L&F) class used to render this component. |
| boolean | [**getUpdateTableInRealTime**](http://docs.google.com/javax/swing/table/JTableHeader.html#getUpdateTableInRealTime())()            Obsolete as of Java 2 platform v1.3. |
| protected  void | [**initializeLocalVars**](http://docs.google.com/javax/swing/table/JTableHeader.html#initializeLocalVars())()            Initializes the local variables and properties with default values. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/table/JTableHeader.html#paramString())()            Returns a string representation of this JTableHeader. |
| void | [**resizeAndRepaint**](http://docs.google.com/javax/swing/table/JTableHeader.html#resizeAndRepaint())()            Sizes the header and marks it as needing display. |
| void | [**setColumnModel**](http://docs.google.com/javax/swing/table/JTableHeader.html#setColumnModel(javax.swing.table.TableColumnModel))([TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) columnModel)            Sets the column model for this table to newModel and registers for listener notifications from the new column model. |
| void | [**setDefaultRenderer**](http://docs.google.com/javax/swing/table/JTableHeader.html#setDefaultRenderer(javax.swing.table.TableCellRenderer))([TableCellRenderer](http://docs.google.com/javax/swing/table/TableCellRenderer.html) defaultRenderer)            Sets the default renderer to be used when no headerRenderer is defined by a TableColumn. |
| void | [**setDraggedColumn**](http://docs.google.com/javax/swing/table/JTableHeader.html#setDraggedColumn(javax.swing.table.TableColumn))([TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) aColumn)            Sets the header's draggedColumn to aColumn. |
| void | [**setDraggedDistance**](http://docs.google.com/javax/swing/table/JTableHeader.html#setDraggedDistance(int))(int distance)            Sets the header's draggedDistance to distance. |
| void | [**setReorderingAllowed**](http://docs.google.com/javax/swing/table/JTableHeader.html#setReorderingAllowed(boolean))(boolean reorderingAllowed)            Sets whether the user can drag column headers to reorder columns. |
| void | [**setResizingAllowed**](http://docs.google.com/javax/swing/table/JTableHeader.html#setResizingAllowed(boolean))(boolean resizingAllowed)            Sets whether the user can resize columns by dragging between headers. |
| void | [**setResizingColumn**](http://docs.google.com/javax/swing/table/JTableHeader.html#setResizingColumn(javax.swing.table.TableColumn))([TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) aColumn)            Sets the header's resizingColumn to aColumn. |
| void | [**setTable**](http://docs.google.com/javax/swing/table/JTableHeader.html#setTable(javax.swing.JTable))([JTable](http://docs.google.com/javax/swing/JTable.html) table)            Sets the table associated with this header. |
| void | [**setUI**](http://docs.google.com/javax/swing/table/JTableHeader.html#setUI(javax.swing.plaf.TableHeaderUI))([TableHeaderUI](http://docs.google.com/javax/swing/plaf/TableHeaderUI.html) ui)            Sets the look and feel (L&F) object that renders this component. |
| void | [**setUpdateTableInRealTime**](http://docs.google.com/javax/swing/table/JTableHeader.html#setUpdateTableInRealTime(boolean))(boolean flag)            Obsolete as of Java 2 platform v1.3. |
| void | [**updateUI**](http://docs.google.com/javax/swing/table/JTableHeader.html#updateUI())()            Notification from the UIManager that the look and feel (L&F) has changed. |

| **Methods inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [addAncestorListener](http://docs.google.com/javax/swing/JComponent.html#addAncestorListener(javax.swing.event.AncestorListener)), [addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()), [addVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#addVetoableChangeListener(java.beans.VetoableChangeListener)), [computeVisibleRect](http://docs.google.com/javax/swing/JComponent.html#computeVisibleRect(java.awt.Rectangle)), [contains](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [createToolTip](http://docs.google.com/javax/swing/JComponent.html#createToolTip()), [disable](http://docs.google.com/javax/swing/JComponent.html#disable()), [enable](http://docs.google.com/javax/swing/JComponent.html#enable()), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20int,%20int)), [fireVetoableChange](http://docs.google.com/javax/swing/JComponent.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [getActionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getActionForKeyStroke(javax.swing.KeyStroke)), [getActionMap](http://docs.google.com/javax/swing/JComponent.html#getActionMap()), [getAlignmentX](http://docs.google.com/javax/swing/JComponent.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/javax/swing/JComponent.html#getAlignmentY()), [getAncestorListeners](http://docs.google.com/javax/swing/JComponent.html#getAncestorListeners()), [getAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#getAutoscrolls()), [getBaseline](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/JComponent.html#getBaselineResizeBehavior()), [getBorder](http://docs.google.com/javax/swing/JComponent.html#getBorder()), [getBounds](http://docs.google.com/javax/swing/JComponent.html#getBounds(java.awt.Rectangle)), [getClientProperty](http://docs.google.com/javax/swing/JComponent.html#getClientProperty(java.lang.Object)), [getComponentGraphics](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [getComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getComponentPopupMenu()), [getConditionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getConditionForKeyStroke(javax.swing.KeyStroke)), [getDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#getDebugGraphicsOptions()), [getDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale()), [getFontMetrics](http://docs.google.com/javax/swing/JComponent.html#getFontMetrics(java.awt.Font)), [getGraphics](http://docs.google.com/javax/swing/JComponent.html#getGraphics()), [getHeight](http://docs.google.com/javax/swing/JComponent.html#getHeight()), [getInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getInheritsPopupMenu()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [getInputVerifier](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)), [getListeners](http://docs.google.com/javax/swing/JComponent.html#getListeners(java.lang.Class)), [getLocation](http://docs.google.com/javax/swing/JComponent.html#getLocation(java.awt.Point)), [getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [getNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent()), [getPopupLocation](http://docs.google.com/javax/swing/JComponent.html#getPopupLocation(java.awt.event.MouseEvent)), [getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [getRegisteredKeyStrokes](http://docs.google.com/javax/swing/JComponent.html#getRegisteredKeyStrokes()), [getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()), [getSize](http://docs.google.com/javax/swing/JComponent.html#getSize(java.awt.Dimension)), [getToolTipLocation](http://docs.google.com/javax/swing/JComponent.html#getToolTipLocation(java.awt.event.MouseEvent)), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText()), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()), [paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)), [paintChildren](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(int,%20int,%20int,%20int)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(java.awt.Rectangle)), [print](http://docs.google.com/javax/swing/JComponent.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/javax/swing/JComponent.html#printAll(java.awt.Graphics)), [printBorder](http://docs.google.com/javax/swing/JComponent.html#printBorder(java.awt.Graphics)), [printChildren](http://docs.google.com/javax/swing/JComponent.html#printChildren(java.awt.Graphics)), [printComponent](http://docs.google.com/javax/swing/JComponent.html#printComponent(java.awt.Graphics)), [processComponentKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processComponentKeyEvent(java.awt.event.KeyEvent)), [processKeyBinding](http://docs.google.com/javax/swing/JComponent.html#processKeyBinding(javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20int,%20boolean)), [processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [putClientProperty](http://docs.google.com/javax/swing/JComponent.html#putClientProperty(java.lang.Object,%20java.lang.Object)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20javax.swing.KeyStroke,%20int)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int)), [removeAncestorListener](http://docs.google.com/javax/swing/JComponent.html#removeAncestorListener(javax.swing.event.AncestorListener)), [removeNotify](http://docs.google.com/javax/swing/JComponent.html#removeNotify()), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

| **Methods inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [addContainerListener](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int)), [countComponents](http://docs.google.com/java/awt/Container.html#countComponents()), [deliverEvent](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event)), [doLayout](http://docs.google.com/java/awt/Container.html#doLayout()), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(int,%20int)), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(java.awt.Point)), [getComponent](http://docs.google.com/java/awt/Container.html#getComponent(int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(java.awt.Point)), [getComponentCount](http://docs.google.com/java/awt/Container.html#getComponentCount()), [getComponents](http://docs.google.com/java/awt/Container.html#getComponents()), [getComponentZOrder](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component)), [getContainerListeners](http://docs.google.com/java/awt/Container.html#getContainerListeners()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int)), [getFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [insets](http://docs.google.com/java/awt/Container.html#insets()), [invalidate](http://docs.google.com/java/awt/Container.html#invalidate()), [isAncestorOf](http://docs.google.com/java/awt/Container.html#isAncestorOf(java.awt.Component)), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [isFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider()), [isFocusTraversalPolicySet](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicySet()), [layout](http://docs.google.com/java/awt/Container.html#layout()), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Container.html#locate(int,%20int)), [minimumSize](http://docs.google.com/java/awt/Container.html#minimumSize()), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [printComponents](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics)), [processContainerEvent](http://docs.google.com/java/awt/Container.html#processContainerEvent(java.awt.event.ContainerEvent)), [processEvent](http://docs.google.com/java/awt/Container.html#processEvent(java.awt.AWTEvent)), [remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [remove](http://docs.google.com/java/awt/Container.html#remove(int)), [removeAll](http://docs.google.com/java/awt/Container.html#removeAll()), [removeContainerListener](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [setComponentZOrder](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int)), [setFocusCycleRoot](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [validate](http://docs.google.com/java/awt/Container.html#validate()), [validateTree](http://docs.google.com/java/awt/Container.html#validateTree()) |

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processInputMethodEvent](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)), [remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)), [removeComponentListener](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [removeFocusListener](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [removeHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyListener](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [removeInputMethodListener](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [removeKeyListener](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [removeMouseListener](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [removeMouseMotionListener](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseWheelListener](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [repaint](http://docs.google.com/java/awt/Component.html#repaint()), [repaint](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long)), [resize](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension)), [resize](http://docs.google.com/java/awt/Component.html#resize(int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setIgnoreRepaint](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean)), [setLocale](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [show](http://docs.google.com/java/awt/Component.html#show()), [show](http://docs.google.com/java/awt/Component.html#show(boolean)), [size](http://docs.google.com/java/awt/Component.html#size()), [toString](http://docs.google.com/java/awt/Component.html#toString()), [transferFocus](http://docs.google.com/java/awt/Component.html#transferFocus()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()) |

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

### table

protected [JTable](http://docs.google.com/javax/swing/JTable.html) **table**

The table for which this object is the header; the default is null.

### columnModel

protected [TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) **columnModel**

The TableColumnModel of the table header.

### reorderingAllowed

protected boolean **reorderingAllowed**

If true, reordering of columns are allowed by the user; the default is true.

### resizingAllowed

protected boolean **resizingAllowed**

If true, resizing of columns are allowed by the user; the default is true.

### updateTableInRealTime

protected boolean **updateTableInRealTime**

Obsolete as of Java 2 platform v1.3. Real time repaints, in response to column dragging or resizing, are now unconditional.

### resizingColumn

protected transient [TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) **resizingColumn**

The index of the column being resized. null if not resizing.

### draggedColumn

protected transient [TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) **draggedColumn**

The index of the column being dragged. null if not dragging.

### draggedDistance

protected transient int **draggedDistance**

The distance from its original position the column has been dragged.

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

### JTableHeader

public **JTableHeader**()

Constructs a JTableHeader with a default TableColumnModel.

**See Also:**[createDefaultColumnModel()](http://docs.google.com/javax/swing/table/JTableHeader.html#createDefaultColumnModel())

### JTableHeader

public **JTableHeader**([TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) cm)

Constructs a JTableHeader which is initialized with cm as the column model. If cm is null this method will initialize the table header with a default TableColumnModel.

**Parameters:**cm - the column model for the table**See Also:**[createDefaultColumnModel()](http://docs.google.com/javax/swing/table/JTableHeader.html#createDefaultColumnModel())

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

### setTable

public void **setTable**([JTable](http://docs.google.com/javax/swing/JTable.html) table)

Sets the table associated with this header.

**Parameters:**table - the new table

### getTable

public [JTable](http://docs.google.com/javax/swing/JTable.html) **getTable**()

Returns the table associated with this header.

**Returns:**the table property

### setReorderingAllowed

public void **setReorderingAllowed**(boolean reorderingAllowed)

Sets whether the user can drag column headers to reorder columns.

**Parameters:**reorderingAllowed - true if the table view should allow reordering; otherwise false**See Also:**[getReorderingAllowed()](http://docs.google.com/javax/swing/table/JTableHeader.html#getReorderingAllowed())

### getReorderingAllowed

public boolean **getReorderingAllowed**()

Returns true if the user is allowed to rearrange columns by dragging their headers, false otherwise. The default is true. You can rearrange columns programmatically regardless of this setting.

**Returns:**the reorderingAllowed property**See Also:**[setReorderingAllowed(boolean)](http://docs.google.com/javax/swing/table/JTableHeader.html#setReorderingAllowed(boolean))

### setResizingAllowed

public void **setResizingAllowed**(boolean resizingAllowed)

Sets whether the user can resize columns by dragging between headers.

**Parameters:**resizingAllowed - true if table view should allow resizing**See Also:**[getResizingAllowed()](http://docs.google.com/javax/swing/table/JTableHeader.html#getResizingAllowed())

### getResizingAllowed

public boolean **getResizingAllowed**()

Returns true if the user is allowed to resize columns by dragging between their headers, false otherwise. The default is true. You can resize columns programmatically regardless of this setting.

**Returns:**the resizingAllowed property**See Also:**[setResizingAllowed(boolean)](http://docs.google.com/javax/swing/table/JTableHeader.html#setResizingAllowed(boolean))

### getDraggedColumn

public [TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) **getDraggedColumn**()

Returns the the dragged column, if and only if, a drag is in process, otherwise returns null.

**Returns:**the dragged column, if a drag is in process, otherwise returns null**See Also:**[getDraggedDistance()](http://docs.google.com/javax/swing/table/JTableHeader.html#getDraggedDistance())

### getDraggedDistance

public int **getDraggedDistance**()

Returns the column's horizontal distance from its original position, if and only if, a drag is in process. Otherwise, the the return value is meaningless.

**Returns:**the column's horizontal distance from its original position, if a drag is in process, otherwise the return value is meaningless**See Also:**[getDraggedColumn()](http://docs.google.com/javax/swing/table/JTableHeader.html#getDraggedColumn())

### getResizingColumn

public [TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) **getResizingColumn**()

Returns the resizing column. If no column is being resized this method returns null.

**Returns:**the resizing column, if a resize is in process, otherwise returns null

### setUpdateTableInRealTime

public void **setUpdateTableInRealTime**(boolean flag)

Obsolete as of Java 2 platform v1.3. Real time repaints, in response to column dragging or resizing, are now unconditional.

### getUpdateTableInRealTime

public boolean **getUpdateTableInRealTime**()

Obsolete as of Java 2 platform v1.3. Real time repaints, in response to column dragging or resizing, are now unconditional.

### setDefaultRenderer

public void **setDefaultRenderer**([TableCellRenderer](http://docs.google.com/javax/swing/table/TableCellRenderer.html) defaultRenderer)

Sets the default renderer to be used when no headerRenderer is defined by a TableColumn.

**Parameters:**defaultRenderer - the default renderer**Since:** 1.3

### getDefaultRenderer

public [TableCellRenderer](http://docs.google.com/javax/swing/table/TableCellRenderer.html) **getDefaultRenderer**()

Returns the default renderer used when no headerRenderer is defined by a TableColumn.

**Returns:**the default renderer**Since:** 1.3

### columnAtPoint

public int **columnAtPoint**([Point](http://docs.google.com/java/awt/Point.html) point)

Returns the index of the column that point lies in, or -1 if it lies out of bounds.

**Returns:**the index of the column that point lies in, or -1 if it lies out of bounds

### getHeaderRect

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getHeaderRect**(int column)

Returns the rectangle containing the header tile at column. When the column parameter is out of bounds this method uses the same conventions as the JTable method getCellRect.

**Returns:**the rectangle containing the header tile at column**See Also:**[JTable.getCellRect(int, int, boolean)](http://docs.google.com/javax/swing/JTable.html#getCellRect(int,%20int,%20boolean))

### getToolTipText

public [String](http://docs.google.com/java/lang/String.html) **getToolTipText**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event)

Allows the renderer's tips to be used if there is text set.

**Overrides:**[getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText(java.awt.event.MouseEvent)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**event - the location of the event identifies the proper renderer and, therefore, the proper tip **Returns:**the tool tip for this component

### getUI

public [TableHeaderUI](http://docs.google.com/javax/swing/plaf/TableHeaderUI.html) **getUI**()

Returns the look and feel (L&F) object that renders this component.

**Returns:**the TableHeaderUI object that renders this component

### setUI

public void **setUI**([TableHeaderUI](http://docs.google.com/javax/swing/plaf/TableHeaderUI.html) ui)

Sets the look and feel (L&F) object that renders this component.

**Parameters:**ui - the TableHeaderUI L&F object**See Also:**[UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### updateUI

public void **updateUI**()

Notification from the UIManager that the look and feel (L&F) has changed. Replaces the current UI object with the latest version from the UIManager.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/JComponent.html#updateUI()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **See Also:**[JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### getUIClassID

public [String](http://docs.google.com/java/lang/String.html) **getUIClassID**()

Returns the suffix used to construct the name of the look and feel (L&F) class used to render this component.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the string "TableHeaderUI"**See Also:**[JComponent.getUIClassID()](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()), [UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### setColumnModel

public void **setColumnModel**([TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) columnModel)

Sets the column model for this table to newModel and registers for listener notifications from the new column model.

**Parameters:**columnModel - the new data source for this table **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if newModel is null**See Also:**[getColumnModel()](http://docs.google.com/javax/swing/table/JTableHeader.html#getColumnModel())

### getColumnModel

public [TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) **getColumnModel**()

Returns the TableColumnModel that contains all column information of this table header.

**Returns:**the columnModel property**See Also:**[setColumnModel(javax.swing.table.TableColumnModel)](http://docs.google.com/javax/swing/table/JTableHeader.html#setColumnModel(javax.swing.table.TableColumnModel))

### columnAdded

public void **columnAdded**([TableColumnModelEvent](http://docs.google.com/javax/swing/event/TableColumnModelEvent.html) e)

Invoked when a column is added to the table column model.

Application code will not use these methods explicitly, they are used internally by JTable.

**Specified by:**[columnAdded](http://docs.google.com/javax/swing/event/TableColumnModelListener.html#columnAdded(javax.swing.event.TableColumnModelEvent)) in interface [TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html) **Parameters:**e - the event received**See Also:**[TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html)

### columnRemoved

public void **columnRemoved**([TableColumnModelEvent](http://docs.google.com/javax/swing/event/TableColumnModelEvent.html) e)

Invoked when a column is removed from the table column model.

Application code will not use these methods explicitly, they are used internally by JTable.

**Specified by:**[columnRemoved](http://docs.google.com/javax/swing/event/TableColumnModelListener.html#columnRemoved(javax.swing.event.TableColumnModelEvent)) in interface [TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html) **Parameters:**e - the event received**See Also:**[TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html)

### columnMoved

public void **columnMoved**([TableColumnModelEvent](http://docs.google.com/javax/swing/event/TableColumnModelEvent.html) e)

Invoked when a column is repositioned.

Application code will not use these methods explicitly, they are used internally by JTable.

**Specified by:**[columnMoved](http://docs.google.com/javax/swing/event/TableColumnModelListener.html#columnMoved(javax.swing.event.TableColumnModelEvent)) in interface [TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html) **Parameters:**e - the event received**See Also:**[TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html)

### columnMarginChanged

public void **columnMarginChanged**([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)

Invoked when a column is moved due to a margin change.

Application code will not use these methods explicitly, they are used internally by JTable.

**Specified by:**[columnMarginChanged](http://docs.google.com/javax/swing/event/TableColumnModelListener.html#columnMarginChanged(javax.swing.event.ChangeEvent)) in interface [TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html) **Parameters:**e - the event received**See Also:**[TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html)

### columnSelectionChanged

public void **columnSelectionChanged**([ListSelectionEvent](http://docs.google.com/javax/swing/event/ListSelectionEvent.html) e)

Invoked when the selection model of the TableColumnModel is changed. This method currently has no effect (the header is not redrawn).

Application code will not use these methods explicitly, they are used internally by JTable.

**Specified by:**[columnSelectionChanged](http://docs.google.com/javax/swing/event/TableColumnModelListener.html#columnSelectionChanged(javax.swing.event.ListSelectionEvent)) in interface [TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html) **Parameters:**e - the event received**See Also:**[TableColumnModelListener](http://docs.google.com/javax/swing/event/TableColumnModelListener.html)

### createDefaultColumnModel

protected [TableColumnModel](http://docs.google.com/javax/swing/table/TableColumnModel.html) **createDefaultColumnModel**()

Returns the default column model object which is a DefaultTableColumnModel. A subclass can override this method to return a different column model object

**Returns:**the default column model object

### createDefaultRenderer

protected [TableCellRenderer](http://docs.google.com/javax/swing/table/TableCellRenderer.html) **createDefaultRenderer**()

Returns a default renderer to be used when no header renderer is defined by a TableColumn.

**Returns:**the default table column renderer**Since:** 1.3

### initializeLocalVars

protected void **initializeLocalVars**()

Initializes the local variables and properties with default values. Used by the constructor methods.

### resizeAndRepaint

public void **resizeAndRepaint**()

Sizes the header and marks it as needing display. Equivalent to revalidate followed by repaint.

### setDraggedColumn

public void **setDraggedColumn**([TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) aColumn)

Sets the header's draggedColumn to aColumn.

Application code will not use this method explicitly, it is used internally by the column dragging mechanism.

**Parameters:**aColumn - the column being dragged, or null if no column is being dragged

### setDraggedDistance

public void **setDraggedDistance**(int distance)

Sets the header's draggedDistance to distance.

**Parameters:**distance - the distance dragged

### setResizingColumn

public void **setResizingColumn**([TableColumn](http://docs.google.com/javax/swing/table/TableColumn.html) aColumn)

Sets the header's resizingColumn to aColumn.

Application code will not use this method explicitly, it is used internally by the column sizing mechanism.

**Parameters:**aColumn - the column being resized, or null if no column is being resized

### paramString

protected [String](http://docs.google.com/java/lang/String.html) **paramString**()

Returns a string representation of this JTableHeader. This method is intended to be used only for debugging purposes, and the content and format of the returned string may vary between implementations. The returned string may be empty but may not be null.

Overriding paramString to provide information about the specific new aspects of the JFC components.

**Overrides:**[paramString](http://docs.google.com/javax/swing/JComponent.html#paramString()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**a string representation of this JTableHeader

### getAccessibleContext

public [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) **getAccessibleContext**()

Gets the AccessibleContext associated with this JTableHeader. For JTableHeaders, the AccessibleContext takes the form of an AccessibleJTableHeader. A new AccessibleJTableHeader instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/javax/swing/JComponent.html#getAccessibleContext()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**an AccessibleJTableHeader that serves as the AccessibleContext of this JTableHeader

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JTableHeader.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/table/DefaultTableModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/table/JTableHeader.AccessibleJTableHeader.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/table/JTableHeader.html)    [**NO FRAMES**](http://docs.google.com/JTableHeader.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#qsh70q) | [METHOD](#49x2ik5) |

[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).