| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicTableHeaderUI.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/plaf/basic/BasicTabbedPaneUI.TabSelectionHandler.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.MouseInputHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicTableHeaderUI.html)    [**NO FRAMES**](http://docs.google.com/BasicTableHeaderUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

## **javax.swing.plaf.basic**

Class BasicTableHeaderUI

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.plaf.ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)  
 [javax.swing.plaf.TableHeaderUI](http://docs.google.com/javax/swing/plaf/TableHeaderUI.html)  
 **javax.swing.plaf.basic.BasicTableHeaderUI**

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

BasicTableHeaderUI implementation

| **Nested Class Summary** | |
| --- | --- |
| class | [**BasicTableHeaderUI.MouseInputHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.MouseInputHandler.html)            This inner class is marked "public" due to a compiler bug. |

| **Field Summary** | |
| --- | --- |
| protected  [JTableHeader](http://docs.google.com/javax/swing/table/JTableHeader.html) | [**header**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#header)            The JTableHeader that is delegating the painting to this UI. |
| protected  [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) | [**mouseInputListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#mouseInputListener) |
| protected  [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html) | [**rendererPane**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#rendererPane) |

| **Constructor Summary** | |
| --- | --- |
| [**BasicTableHeaderUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#BasicTableHeaderUI())() |

| **Method Summary** | |
| --- | --- |
| protected  [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) | [**createMouseInputListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#createMouseInputListener())()            Creates the mouse listener for the JTableHeader. |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) h) |
| int | [**getBaseline**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#getBaseline(javax.swing.JComponent,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c, int width, int height)            Returns the baseline. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#getMaximumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Return the maximum size of the header. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#getMinimumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Return the minimum size of the header. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#getPreferredSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Return the preferred size of the header. |
| protected  int | [**getRolloverColumn**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#getRolloverColumn())()            Returns the index of the column header over which the mouse currently is. |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#installDefaults())()            Initialize JTableHeader properties, e.g. |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#installKeyboardActions())()            Register all keyboard actions on the JTableHeader. |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#installListeners())()            Attaches listeners to the JTableHeader. |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#installUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Configures the specified component appropriate for the look and feel. |
| void | [**paint**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Paints the specified component appropriate for the look and feel. |
| protected  void | [**rolloverColumnUpdated**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#rolloverColumnUpdated(int,%20int))(int oldColumn, int newColumn)            This method gets called every time the rollover column in the table header is updated. |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#uninstallDefaults())() |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#uninstallKeyboardActions())()            Unregisters default key actions. |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#uninstallListeners())() |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#uninstallUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Reverses configuration which was done on the specified component during installUI. |

| **Methods inherited from class javax.swing.plaf.**[**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html) |
| --- |
| [contains](http://docs.google.com/javax/swing/plaf/ComponentUI.html#contains(javax.swing.JComponent,%20int,%20int)), [getAccessibleChild](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChild(javax.swing.JComponent,%20int)), [getAccessibleChildrenCount](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChildrenCount(javax.swing.JComponent)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaselineResizeBehavior(javax.swing.JComponent)), [update](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

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

### header

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

The JTableHeader that is delegating the painting to this UI.

### rendererPane

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

### mouseInputListener

protected [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) **mouseInputListener**

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

### BasicTableHeaderUI

public **BasicTableHeaderUI**()

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

### createMouseInputListener

protected [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) **createMouseInputListener**()

Creates the mouse listener for the JTableHeader.

### createUI

public static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **createUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) h)

### installUI

public void **installUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)) Configures the specified component appropriate for the look and feel. This method is invoked when the ComponentUI instance is being installed as the UI delegate on the specified component. This method should completely configure the component for the look and feel, including the following:

1. Install any default property values for color, fonts, borders, icons, opacity, etc. on the component. Whenever possible, property values initialized by the client program should *not* be overridden.
2. Install a LayoutManager on the component if necessary.
3. Create/add any required sub-components to the component.
4. Create/install event listeners on the component.
5. Create/install a PropertyChangeListener on the component in order to detect and respond to component property changes appropriately.
6. Install keyboard UI (mnemonics, traversal, etc.) on the component.
7. Initialize any appropriate instance data.

**Overrides:**[installUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component where this UI delegate is being installed**See Also:**[ComponentUI.uninstallUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)), [JComponent.setUI(javax.swing.plaf.ComponentUI)](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### installDefaults

protected void **installDefaults**()

Initialize JTableHeader properties, e.g. font, foreground, and background. The font, foreground, and background properties are only set if their current value is either null or a UIResource, other properties are set if the current value is null.

**See Also:**[installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#installUI(javax.swing.JComponent))

### installListeners

protected void **installListeners**()

Attaches listeners to the JTableHeader.

### installKeyboardActions

protected void **installKeyboardActions**()

Register all keyboard actions on the JTableHeader.

### uninstallUI

public void **uninstallUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)) Reverses configuration which was done on the specified component during installUI. This method is invoked when this UIComponent instance is being removed as the UI delegate for the specified component. This method should undo the configuration performed in installUI, being careful to leave the JComponent instance in a clean state (no extraneous listeners, look-and-feel-specific property objects, etc.). This should include the following:

1. Remove any UI-set borders from the component.
2. Remove any UI-set layout managers on the component.
3. Remove any UI-added sub-components from the component.
4. Remove any UI-added event/property listeners from the component.
5. Remove any UI-installed keyboard UI from the component.
6. Nullify any allocated instance data objects to allow for GC.

**Overrides:**[uninstallUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component from which this UI delegate is being removed; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### uninstallDefaults

protected void **uninstallDefaults**()

### uninstallListeners

protected void **uninstallListeners**()

### uninstallKeyboardActions

protected void **uninstallKeyboardActions**()

Unregisters default key actions.

### getRolloverColumn

protected int **getRolloverColumn**()

Returns the index of the column header over which the mouse currently is. When the mouse is not over the table header, -1 is returned.

**Returns:**the index of the current rollover column**Since:** 1.6 **See Also:**[rolloverColumnUpdated(int, int)](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#rolloverColumnUpdated(int,%20int))

### rolloverColumnUpdated

protected void **rolloverColumnUpdated**(int oldColumn,  
 int newColumn)

This method gets called every time the rollover column in the table header is updated. Every look and feel supporting rollover effect in table header should override this method and repaint the header.

**Parameters:**oldColumn - the index of the previous rollover column or -1 if the mouse was not over a columnnewColumn - the index of the new rollover column or -1 if the mouse is not over a column**Since:** 1.6 **See Also:**[getRolloverColumn()](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.html#getRolloverColumn()), [JTableHeader.getHeaderRect(int)](http://docs.google.com/javax/swing/table/JTableHeader.html#getHeaderRect(int))

### getBaseline

public int **getBaseline**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 int width,  
 int height)

Returns the baseline.

**Overrides:**[getBaseline](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaseline(javax.swing.JComponent,%20int,%20int)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - JComponent baseline is being requested forwidth - the width to get the baseline forheight - the height to get the baseline for **Returns:**baseline or a value < 0 indicating there is no reasonable baseline **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if c is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if width or height is < 0**Since:** 1.6 **See Also:**[JComponent.getBaseline(int, int)](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int))

### paint

public void **paint**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) Paints the specified component appropriate for the look and feel. This method is invoked from the ComponentUI.update method when the specified component is being painted. Subclasses should override this method and use the specified Graphics object to render the content of the component.

**Overrides:**[paint](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**g - the Graphics context in which to paintc - the component being painted; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.update(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent))

### getMinimumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMinimumSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Return the minimum size of the header. The minimum width is the sum of the minimum widths of each column (plus inter-cell spacing).

**Overrides:**[getMinimumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMinimumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose minimum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**a Dimension object or null**See Also:**[JComponent.getMinimumSize()](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [LayoutManager.minimumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)), [ComponentUI.getPreferredSize(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent))

### getPreferredSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getPreferredSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Return the preferred size of the header. The preferred height is the maximum of the preferred heights of all of the components provided by the header renderers. The preferred width is the sum of the preferred widths of each column (plus inter-cell spacing).

**Overrides:**[getPreferredSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose preferred size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[JComponent.getPreferredSize()](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [LayoutManager.preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container))

### getMaximumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMaximumSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Return the maximum size of the header. The maximum width is the sum of the maximum widths of each column (plus inter-cell spacing).

**Overrides:**[getMaximumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose maximum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**a Dimension object or null**See Also:**[JComponent.getMaximumSize()](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [LayoutManager2.maximumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager2.html#maximumLayoutSize(java.awt.Container))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicTableHeaderUI.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/plaf/basic/BasicTabbedPaneUI.TabSelectionHandler.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicTableHeaderUI.MouseInputHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicTableHeaderUI.html)    [**NO FRAMES**](http://docs.google.com/BasicTableHeaderUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [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).