| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicTabbedPaneUI.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/BasicSplitPaneUI.PropertyHandler.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.FocusHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicTabbedPaneUI.html)    [**NO FRAMES**](http://docs.google.com/BasicTabbedPaneUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#2grqrue) | [METHOD](#3fwokq0) |

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

Class BasicTabbedPaneUI

[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.TabbedPaneUI](http://docs.google.com/javax/swing/plaf/TabbedPaneUI.html)  
 **javax.swing.plaf.basic.BasicTabbedPaneUI**

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

public class **BasicTabbedPaneUI**extends [TabbedPaneUI](http://docs.google.com/javax/swing/plaf/TabbedPaneUI.html)implements [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

A Basic L&F implementation of TabbedPaneUI.

| **Nested Class Summary** | |
| --- | --- |
| class | [**BasicTabbedPaneUI.FocusHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.FocusHandler.html)            This class should be treated as a "protected" inner class. |
| class | [**BasicTabbedPaneUI.MouseHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.MouseHandler.html)            This class should be treated as a "protected" inner class. |
| class | [**BasicTabbedPaneUI.PropertyChangeHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.PropertyChangeHandler.html)            This class should be treated as a "protected" inner class. |
| class | [**BasicTabbedPaneUI.TabbedPaneLayout**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.TabbedPaneLayout.html)            This class should be treated as a "protected" inner class. |
| class | [**BasicTabbedPaneUI.TabSelectionHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.TabSelectionHandler.html)            This class should be treated as a "protected" inner class. |

| **Field Summary** | |
| --- | --- |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**calcRect**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#calcRect)            A rectangle used for general layout calculations in order to avoid constructing many new Rectangles on the fly. |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**contentBorderInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#contentBorderInsets) |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**darkShadow**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#darkShadow) |
| protected  [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**downKey**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#downKey)  **Deprecated.** *As of Java 2 platform v1.3.* |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**focus**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#focus) |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**focusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#focusListener) |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**highlight**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#highlight) |
| protected  [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**leftKey**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#leftKey)  **Deprecated.** *As of Java 2 platform v1.3.* |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**lightHighlight**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#lightHighlight) |
| protected  int | [**maxTabHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#maxTabHeight) |
| protected  int | [**maxTabWidth**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#maxTabWidth) |
| protected  [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) | [**mouseListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#mouseListener) |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**propertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#propertyChangeListener) |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html)[] | [**rects**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#rects) |
| protected  [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**rightKey**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#rightKey)  **Deprecated.** *As of Java 2 platform v1.3.* |
| protected  int | [**runCount**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#runCount) |
| protected  int | [**selectedRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#selectedRun) |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**selectedTabPadInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#selectedTabPadInsets) |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**shadow**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#shadow) |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**tabAreaInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#tabAreaInsets) |
| protected  [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) | [**tabChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#tabChangeListener) |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**tabInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#tabInsets) |
| protected  [JTabbedPane](http://docs.google.com/javax/swing/JTabbedPane.html) | [**tabPane**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#tabPane) |
| protected  int | [**tabRunOverlay**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#tabRunOverlay) |
| protected  int[] | [**tabRuns**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#tabRuns) |
| protected  int | [**textIconGap**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#textIconGap) |
| protected  [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**upKey**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#upKey)  **Deprecated.** *As of Java 2 platform v1.3.* |

| **Fields inherited from interface javax.swing.**[**SwingConstants**](http://docs.google.com/javax/swing/SwingConstants.html) |
| --- |
| [BOTTOM](http://docs.google.com/javax/swing/SwingConstants.html#BOTTOM), [CENTER](http://docs.google.com/javax/swing/SwingConstants.html#CENTER), [EAST](http://docs.google.com/javax/swing/SwingConstants.html#EAST), [HORIZONTAL](http://docs.google.com/javax/swing/SwingConstants.html#HORIZONTAL), [LEADING](http://docs.google.com/javax/swing/SwingConstants.html#LEADING), [LEFT](http://docs.google.com/javax/swing/SwingConstants.html#LEFT), [NEXT](http://docs.google.com/javax/swing/SwingConstants.html#NEXT), [NORTH](http://docs.google.com/javax/swing/SwingConstants.html#NORTH), [NORTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_EAST), [NORTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_WEST), [PREVIOUS](http://docs.google.com/javax/swing/SwingConstants.html#PREVIOUS), [RIGHT](http://docs.google.com/javax/swing/SwingConstants.html#RIGHT), [SOUTH](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH), [SOUTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_EAST), [SOUTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_WEST), [TOP](http://docs.google.com/javax/swing/SwingConstants.html#TOP), [TRAILING](http://docs.google.com/javax/swing/SwingConstants.html#TRAILING), [VERTICAL](http://docs.google.com/javax/swing/SwingConstants.html#VERTICAL), [WEST](http://docs.google.com/javax/swing/SwingConstants.html#WEST) |

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

| **Method Summary** | |
| --- | --- |
| protected  void | [**assureRectsCreated**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#assureRectsCreated(int))(int tabCount) |
| protected  int | [**calculateMaxTabHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#calculateMaxTabHeight(int))(int tabPlacement) |
| protected  int | [**calculateMaxTabWidth**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#calculateMaxTabWidth(int))(int tabPlacement) |
| protected  int | [**calculateTabAreaHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#calculateTabAreaHeight(int,%20int,%20int))(int tabPlacement, int horizRunCount, int maxTabHeight) |
| protected  int | [**calculateTabAreaWidth**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#calculateTabAreaWidth(int,%20int,%20int))(int tabPlacement, int vertRunCount, int maxTabWidth) |
| protected  int | [**calculateTabHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#calculateTabHeight(int,%20int,%20int))(int tabPlacement, int tabIndex, int fontHeight) |
| protected  int | [**calculateTabWidth**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#calculateTabWidth(int,%20int,%20java.awt.FontMetrics))(int tabPlacement, int tabIndex, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics) |
| protected  [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) | [**createChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#createChangeListener())() |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**createFocusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#createFocusListener())() |
| protected  [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) | [**createLayoutManager**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#createLayoutManager())()            Invoked by installUI to create a layout manager object to manage the JTabbedPane. |
| protected  [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) | [**createMouseListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#createMouseListener())() |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createPropertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#createPropertyChangeListener())() |
| protected  [JButton](http://docs.google.com/javax/swing/JButton.html) | [**createScrollButton**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#createScrollButton(int))(int direction)            Creates and returns a JButton that will provide the user with a way to scroll the tabs in a particular direction. |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c) |
| protected  void | [**expandTabRunsArray**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#expandTabRunsArray())() |
| protected  int | [**getBaseline**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getBaseline(int))(int tab)            Returns the baseline for the specified tab. |
| int | [**getBaseline**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getBaseline(javax.swing.JComponent,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c, int width, int height)            Returns the baseline. |
| protected  int | [**getBaselineOffset**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getBaselineOffset())()            Returns the amount the baseline is offset by. |
| [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) | [**getBaselineResizeBehavior**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getBaselineResizeBehavior(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns an enum indicating how the baseline of the component changes as the size changes. |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**getContentBorderInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getContentBorderInsets(int))(int tabPlacement) |
| protected  int | [**getFocusIndex**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getFocusIndex())()            Returns the index of the tab that has focus. |
| protected  [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) | [**getFontMetrics**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getFontMetrics())() |
| protected  [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getIconForTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getIconForTab(int))(int tabIndex) |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getMaximumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the specified component's maximum size appropriate for the look and feel. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getMinimumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the specified component's minimum size appropriate for the look and feel. |
| protected  int | [**getNextTabIndex**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getNextTabIndex(int))(int base) |
| protected  int | [**getNextTabIndexInRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getNextTabIndexInRun(int,%20int))(int tabCount, int base) |
| protected  int | [**getNextTabRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getNextTabRun(int))(int baseRun) |
| protected  int | [**getPreviousTabIndex**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getPreviousTabIndex(int))(int base) |
| protected  int | [**getPreviousTabIndexInRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getPreviousTabIndexInRun(int,%20int))(int tabCount, int base) |
| protected  int | [**getPreviousTabRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getPreviousTabRun(int))(int baseRun) |
| protected  int | [**getRolloverTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getRolloverTab())()            Returns the tab the mouse is currently over, or -1 if the mouse is no longer over any tab. |
| protected  int | [**getRunForTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getRunForTab(int,%20int))(int tabCount, int tabIndex) |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**getSelectedTabPadInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getSelectedTabPadInsets(int))(int tabPlacement) |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**getTabAreaInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabAreaInsets(int))(int tabPlacement) |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getTabBounds**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabBounds(int,%20java.awt.Rectangle))(int tabIndex, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) dest)            Returns the bounds of the specified tab in the coordinate space of the JTabbedPane component. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getTabBounds**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabBounds(javax.swing.JTabbedPane,%20int))([JTabbedPane](http://docs.google.com/javax/swing/JTabbedPane.html) pane, int i)            Returns the bounds of the specified tab index. |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**getTabInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabInsets(int,%20int))(int tabPlacement, int tabIndex) |
| protected  int | [**getTabLabelShiftX**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabLabelShiftX(int,%20int,%20boolean))(int tabPlacement, int tabIndex, boolean isSelected) |
| protected  int | [**getTabLabelShiftY**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabLabelShiftY(int,%20int,%20boolean))(int tabPlacement, int tabIndex, boolean isSelected) |
| int | [**getTabRunCount**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabRunCount(javax.swing.JTabbedPane))([JTabbedPane](http://docs.google.com/javax/swing/JTabbedPane.html) pane) |
| protected  int | [**getTabRunIndent**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabRunIndent(int,%20int))(int tabPlacement, int run) |
| protected  int | [**getTabRunOffset**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabRunOffset(int,%20int,%20int,%20boolean))(int tabPlacement, int tabCount, int tabIndex, boolean forward) |
| protected  int | [**getTabRunOverlay**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTabRunOverlay(int))(int tabPlacement) |
| protected  [View](http://docs.google.com/javax/swing/text/View.html) | [**getTextViewForTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getTextViewForTab(int))(int tabIndex)            Returns the text View object required to render stylized text (HTML) for the specified tab or null if no specialized text rendering is needed for this tab. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**getVisibleComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#getVisibleComponent())() |
| protected  void | [**installComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#installComponents())()            Creates and installs any required subcomponents for the JTabbedPane. |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#installDefaults())() |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#installKeyboardActions())() |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#installListeners())() |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.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. |
| protected  int | [**lastTabInRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#lastTabInRun(int,%20int))(int tabCount, int run) |
| protected  void | [**layoutLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#layoutLabel(int,%20java.awt.FontMetrics,%20int,%20java.lang.String,%20javax.swing.Icon,%20java.awt.Rectangle,%20java.awt.Rectangle,%20java.awt.Rectangle,%20boolean))(int tabPlacement, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics, int tabIndex, [String](http://docs.google.com/java/lang/String.html) title, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tabRect, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect, boolean isSelected) |
| protected  void | [**navigateSelectedTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#navigateSelectedTab(int))(int direction) |
| void | [**paint**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.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 | [**paintContentBorder**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintContentBorder(java.awt.Graphics,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int selectedIndex) |
| protected  void | [**paintContentBorderBottomEdge**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintContentBorderBottomEdge(java.awt.Graphics,%20int,%20int,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int selectedIndex, int x, int y, int w, int h) |
| protected  void | [**paintContentBorderLeftEdge**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintContentBorderLeftEdge(java.awt.Graphics,%20int,%20int,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int selectedIndex, int x, int y, int w, int h) |
| protected  void | [**paintContentBorderRightEdge**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintContentBorderRightEdge(java.awt.Graphics,%20int,%20int,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int selectedIndex, int x, int y, int w, int h) |
| protected  void | [**paintContentBorderTopEdge**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintContentBorderTopEdge(java.awt.Graphics,%20int,%20int,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int selectedIndex, int x, int y, int w, int h) |
| protected  void | [**paintFocusIndicator**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintFocusIndicator(java.awt.Graphics,%20int,%20java.awt.Rectangle%5B%5D,%20int,%20java.awt.Rectangle,%20java.awt.Rectangle,%20boolean))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, [Rectangle](http://docs.google.com/java/awt/Rectangle.html)[] rects, int tabIndex, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect, boolean isSelected) |
| protected  void | [**paintIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintIcon(java.awt.Graphics,%20int,%20int,%20javax.swing.Icon,%20java.awt.Rectangle,%20boolean))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int tabIndex, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect, boolean isSelected) |
| protected  void | [**paintTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintTab(java.awt.Graphics,%20int,%20java.awt.Rectangle%5B%5D,%20int,%20java.awt.Rectangle,%20java.awt.Rectangle))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, [Rectangle](http://docs.google.com/java/awt/Rectangle.html)[] rects, int tabIndex, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect) |
| protected  void | [**paintTabArea**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintTabArea(java.awt.Graphics,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int selectedIndex)            Paints the tabs in the tab area. |
| protected  void | [**paintTabBackground**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintTabBackground(java.awt.Graphics,%20int,%20int,%20int,%20int,%20int,%20int,%20boolean))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int tabIndex, int x, int y, int w, int h, boolean isSelected) |
| protected  void | [**paintTabBorder**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintTabBorder(java.awt.Graphics,%20int,%20int,%20int,%20int,%20int,%20int,%20boolean))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, int tabIndex, int x, int y, int w, int h, boolean isSelected)            this function draws the border around each tab note that this function does now draw the background of the tab. |
| protected  void | [**paintText**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#paintText(java.awt.Graphics,%20int,%20java.awt.Font,%20java.awt.FontMetrics,%20int,%20java.lang.String,%20java.awt.Rectangle,%20boolean))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int tabPlacement, [Font](http://docs.google.com/java/awt/Font.html) font, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics, int tabIndex, [String](http://docs.google.com/java/lang/String.html) title, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect, boolean isSelected) |
| protected static void | [**rotateInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#rotateInsets(java.awt.Insets,%20java.awt.Insets,%20int))([Insets](http://docs.google.com/java/awt/Insets.html) topInsets, [Insets](http://docs.google.com/java/awt/Insets.html) targetInsets, int targetPlacement) |
| protected  void | [**selectAdjacentRunTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#selectAdjacentRunTab(int,%20int,%20int))(int tabPlacement, int tabIndex, int offset) |
| protected  void | [**selectNextTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#selectNextTab(int))(int current) |
| protected  void | [**selectNextTabInRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#selectNextTabInRun(int))(int current) |
| protected  void | [**selectPreviousTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#selectPreviousTab(int))(int current) |
| protected  void | [**selectPreviousTabInRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#selectPreviousTabInRun(int))(int current) |
| protected  void | [**setRolloverTab**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#setRolloverTab(int))(int index)            Sets the tab the mouse is currently over to index. |
| protected  void | [**setVisibleComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#setVisibleComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) component) |
| protected  boolean | [**shouldPadTabRun**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#shouldPadTabRun(int,%20int))(int tabPlacement, int run) |
| protected  boolean | [**shouldRotateTabRuns**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#shouldRotateTabRuns(int))(int tabPlacement) |
| int | [**tabForCoordinate**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#tabForCoordinate(javax.swing.JTabbedPane,%20int,%20int))([JTabbedPane](http://docs.google.com/javax/swing/JTabbedPane.html) pane, int x, int y)            Returns the tab index which intersects the specified point in the JTabbedPane's coordinate space. |
| protected  void | [**uninstallComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#uninstallComponents())()            Removes any installed subcomponents from the JTabbedPane. |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#uninstallDefaults())() |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#uninstallKeyboardActions())() |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.html#uninstallListeners())() |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.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)), [getPreferredSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(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** |
| --- |

### tabPane

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

### highlight

protected [Color](http://docs.google.com/java/awt/Color.html) **highlight**

### lightHighlight

protected [Color](http://docs.google.com/java/awt/Color.html) **lightHighlight**

### shadow

protected [Color](http://docs.google.com/java/awt/Color.html) **shadow**

### darkShadow

protected [Color](http://docs.google.com/java/awt/Color.html) **darkShadow**

### focus

protected [Color](http://docs.google.com/java/awt/Color.html) **focus**

### textIconGap

protected int **textIconGap**

### tabRunOverlay

protected int **tabRunOverlay**

### tabInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **tabInsets**

### selectedTabPadInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **selectedTabPadInsets**

### tabAreaInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **tabAreaInsets**

### contentBorderInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **contentBorderInsets**

### upKey

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **upKey**

**Deprecated.** *As of Java 2 platform v1.3.*As of Java 2 platform v1.3 this previously undocumented field is no longer used. Key bindings are now defined by the LookAndFeel, please refer to the key bindings specification for further details.

### downKey

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **downKey**

**Deprecated.** *As of Java 2 platform v1.3.*As of Java 2 platform v1.3 this previously undocumented field is no longer used. Key bindings are now defined by the LookAndFeel, please refer to the key bindings specification for further details.

### leftKey

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **leftKey**

**Deprecated.** *As of Java 2 platform v1.3.*As of Java 2 platform v1.3 this previously undocumented field is no longer used. Key bindings are now defined by the LookAndFeel, please refer to the key bindings specification for further details.

### rightKey

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) **rightKey**

**Deprecated.** *As of Java 2 platform v1.3.*As of Java 2 platform v1.3 this previously undocumented field is no longer used. Key bindings are now defined by the LookAndFeel, please refer to the key bindings specification for further details.

### tabRuns

protected int[] **tabRuns**

### runCount

protected int **runCount**

### selectedRun

protected int **selectedRun**

### rects

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html)[] **rects**

### maxTabHeight

protected int **maxTabHeight**

### maxTabWidth

protected int **maxTabWidth**

### tabChangeListener

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

### propertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **propertyChangeListener**

### mouseListener

protected [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) **mouseListener**

### focusListener

protected [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) **focusListener**

### calcRect

protected transient [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **calcRect**

A rectangle used for general layout calculations in order to avoid constructing many new Rectangles on the fly.

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

### BasicTabbedPaneUI

public **BasicTabbedPaneUI**()

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

### createUI

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

### 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())

### 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())

### createLayoutManager

protected [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **createLayoutManager**()

Invoked by installUI to create a layout manager object to manage the JTabbedPane.

**Returns:**a layout manager object**See Also:**[BasicTabbedPaneUI.TabbedPaneLayout](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.TabbedPaneLayout.html), [JTabbedPane.getTabLayoutPolicy()](http://docs.google.com/javax/swing/JTabbedPane.html#getTabLayoutPolicy())

### installComponents

protected void **installComponents**()

Creates and installs any required subcomponents for the JTabbedPane. Invoked by installUI.

**Since:** 1.4

### createScrollButton

protected [JButton](http://docs.google.com/javax/swing/JButton.html) **createScrollButton**(int direction)

Creates and returns a JButton that will provide the user with a way to scroll the tabs in a particular direction. The returned JButton must be instance of UIResource.

**Parameters:**direction - One of the SwingConstants constants: SOUTH, NORTH, EAST or WEST **Returns:**Widget for user to **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if direction is not one of NORTH, SOUTH, EAST or WEST**Since:** 1.5 **See Also:**[JTabbedPane.setTabPlacement(int)](http://docs.google.com/javax/swing/JTabbedPane.html#setTabPlacement(int)), [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

### uninstallComponents

protected void **uninstallComponents**()

Removes any installed subcomponents from the JTabbedPane. Invoked by uninstallUI.

**Since:** 1.4

### installDefaults

protected void **installDefaults**()

### uninstallDefaults

protected void **uninstallDefaults**()

### installListeners

protected void **installListeners**()

### uninstallListeners

protected void **uninstallListeners**()

### createMouseListener

protected [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) **createMouseListener**()

### createFocusListener

protected [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) **createFocusListener**()

### createChangeListener

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

### createPropertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **createPropertyChangeListener**()

### installKeyboardActions

protected void **installKeyboardActions**()

### uninstallKeyboardActions

protected void **uninstallKeyboardActions**()

### setRolloverTab

protected void **setRolloverTab**(int index)

Sets the tab the mouse is currently over to index. index will be -1 if the mouse is no longer over any tab. No checking is done to ensure the passed in index identifies a valid tab.

**Parameters:**index - Index of the tab the mouse is over.**Since:** 1.5

### getRolloverTab

protected int **getRolloverTab**()

Returns the tab the mouse is currently over, or -1 if the mouse is no longer over any tab.

**Returns:**the tab the mouse is currently over, or -1 if the mouse is no longer over any tab**Since:** 1.5

### getMinimumSize

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

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMinimumSize(javax.swing.JComponent)) Returns the specified component's minimum size appropriate for the look and feel. If null is returned, the minimum size will be calculated by the component's layout manager instead (this is the preferred approach for any component with a specific layout manager installed). The default implementation of this method invokes getPreferredSize and returns that value.

**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))

### getMaximumSize

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

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)) Returns the specified component's maximum size appropriate for the look and feel. If null is returned, the maximum size will be calculated by the component's layout manager instead (this is the preferred approach for any component with a specific layout manager installed). The default implementation of this method invokes getPreferredSize and returns that value.

**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))

### 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))

### getBaselineResizeBehavior

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

Returns an enum indicating how the baseline of the component changes as the size changes.

**Overrides:**[getBaselineResizeBehavior](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaselineResizeBehavior(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - JComponent to return baseline resize behavior for **Returns:**an enum indicating how the baseline changes as the component size changes **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if c is null**Since:** 1.6 **See Also:**[JComponent.getBaseline(int, int)](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int))

### getBaseline

protected int **getBaseline**(int tab)

Returns the baseline for the specified tab.

**Parameters:**tab - index of tab to get baseline for **Returns:**baseline or a value < 0 indicating there is no reasonable baseline **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if index is out of range (index < 0 || index >= tab count)**Since:** 1.6

### getBaselineOffset

protected int **getBaselineOffset**()

Returns the amount the baseline is offset by. This is typically the same as getTabLabelShiftY.

**Returns:**amount to offset the baseline by**Since:** 1.6

### 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))

### paintTabArea

protected void **paintTabArea**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int selectedIndex)

Paints the tabs in the tab area. Invoked by paint(). The graphics parameter must be a valid Graphics object. Tab placement may be either: JTabbedPane.TOP, JTabbedPane.BOTTOM, JTabbedPane.LEFT, or JTabbedPane.RIGHT. The selected index must be a valid tabbed pane tab index (0 to tab count - 1, inclusive) or -1 if no tab is currently selected. The handling of invalid parameters is unspecified.

**Parameters:**g - the graphics object to use for renderingtabPlacement - the placement for the tabs within the JTabbedPaneselectedIndex - the tab index of the selected component**Since:** 1.4

### paintTab

protected void **paintTab**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html)[] rects,  
 int tabIndex,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect)

### layoutLabel

protected void **layoutLabel**(int tabPlacement,  
 [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics,  
 int tabIndex,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tabRect,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect,  
 boolean isSelected)

### paintIcon

protected void **paintIcon**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int tabIndex,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect,  
 boolean isSelected)

### paintText

protected void **paintText**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 [Font](http://docs.google.com/java/awt/Font.html) font,  
 [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics,  
 int tabIndex,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect,  
 boolean isSelected)

### getTabLabelShiftX

protected int **getTabLabelShiftX**(int tabPlacement,  
 int tabIndex,  
 boolean isSelected)

### getTabLabelShiftY

protected int **getTabLabelShiftY**(int tabPlacement,  
 int tabIndex,  
 boolean isSelected)

### paintFocusIndicator

protected void **paintFocusIndicator**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html)[] rects,  
 int tabIndex,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect,  
 boolean isSelected)

### paintTabBorder

protected void **paintTabBorder**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int tabIndex,  
 int x,  
 int y,  
 int w,  
 int h,  
 boolean isSelected)

this function draws the border around each tab note that this function does now draw the background of the tab. that is done elsewhere

### paintTabBackground

protected void **paintTabBackground**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int tabIndex,  
 int x,  
 int y,  
 int w,  
 int h,  
 boolean isSelected)

### paintContentBorder

protected void **paintContentBorder**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int selectedIndex)

### paintContentBorderTopEdge

protected void **paintContentBorderTopEdge**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int selectedIndex,  
 int x,  
 int y,  
 int w,  
 int h)

### paintContentBorderLeftEdge

protected void **paintContentBorderLeftEdge**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int selectedIndex,  
 int x,  
 int y,  
 int w,  
 int h)

### paintContentBorderBottomEdge

protected void **paintContentBorderBottomEdge**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int selectedIndex,  
 int x,  
 int y,  
 int w,  
 int h)

### paintContentBorderRightEdge

protected void **paintContentBorderRightEdge**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int tabPlacement,  
 int selectedIndex,  
 int x,  
 int y,  
 int w,  
 int h)

### getTabBounds

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getTabBounds**([JTabbedPane](http://docs.google.com/javax/swing/JTabbedPane.html) pane,  
 int i)

Returns the bounds of the specified tab index. The bounds are with respect to the JTabbedPane's coordinate space.

**Specified by:**[getTabBounds](http://docs.google.com/javax/swing/plaf/TabbedPaneUI.html#getTabBounds(javax.swing.JTabbedPane,%20int)) in class [TabbedPaneUI](http://docs.google.com/javax/swing/plaf/TabbedPaneUI.html)

### getTabRunCount

public int **getTabRunCount**([JTabbedPane](http://docs.google.com/javax/swing/JTabbedPane.html) pane)

**Specified by:**[getTabRunCount](http://docs.google.com/javax/swing/plaf/TabbedPaneUI.html#getTabRunCount(javax.swing.JTabbedPane)) in class [TabbedPaneUI](http://docs.google.com/javax/swing/plaf/TabbedPaneUI.html)

### tabForCoordinate

public int **tabForCoordinate**([JTabbedPane](http://docs.google.com/javax/swing/JTabbedPane.html) pane,  
 int x,  
 int y)

Returns the tab index which intersects the specified point in the JTabbedPane's coordinate space.

**Specified by:**[tabForCoordinate](http://docs.google.com/javax/swing/plaf/TabbedPaneUI.html#tabForCoordinate(javax.swing.JTabbedPane,%20int,%20int)) in class [TabbedPaneUI](http://docs.google.com/javax/swing/plaf/TabbedPaneUI.html)

### getTabBounds

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getTabBounds**(int tabIndex,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) dest)

Returns the bounds of the specified tab in the coordinate space of the JTabbedPane component. This is required because the tab rects are by default defined in the coordinate space of the component where they are rendered, which could be the JTabbedPane (for WRAP\_TAB\_LAYOUT) or a ScrollableTabPanel (SCROLL\_TAB\_LAYOUT). This method should be used whenever the tab rectangle must be relative to the JTabbedPane itself and the result should be placed in a designated Rectangle object (rather than instantiating and returning a new Rectangle each time). The tab index parameter must be a valid tabbed pane tab index (0 to tab count - 1, inclusive). The destination rectangle parameter must be a valid Rectangle instance. The handling of invalid parameters is unspecified.

**Parameters:**tabIndex - the index of the tabdest - the rectangle where the result should be placed **Returns:**the resulting rectangle**Since:** 1.4

### getVisibleComponent

protected [Component](http://docs.google.com/java/awt/Component.html) **getVisibleComponent**()

### setVisibleComponent

protected void **setVisibleComponent**([Component](http://docs.google.com/java/awt/Component.html) component)

### assureRectsCreated

protected void **assureRectsCreated**(int tabCount)

### expandTabRunsArray

protected void **expandTabRunsArray**()

### getRunForTab

protected int **getRunForTab**(int tabCount,  
 int tabIndex)

### lastTabInRun

protected int **lastTabInRun**(int tabCount,  
 int run)

### getTabRunOverlay

protected int **getTabRunOverlay**(int tabPlacement)

### getTabRunIndent

protected int **getTabRunIndent**(int tabPlacement,  
 int run)

### shouldPadTabRun

protected boolean **shouldPadTabRun**(int tabPlacement,  
 int run)

### shouldRotateTabRuns

protected boolean **shouldRotateTabRuns**(int tabPlacement)

### getIconForTab

protected [Icon](http://docs.google.com/javax/swing/Icon.html) **getIconForTab**(int tabIndex)

### getTextViewForTab

protected [View](http://docs.google.com/javax/swing/text/View.html) **getTextViewForTab**(int tabIndex)

Returns the text View object required to render stylized text (HTML) for the specified tab or null if no specialized text rendering is needed for this tab. This is provided to support html rendering inside tabs.

**Parameters:**tabIndex - the index of the tab **Returns:**the text view to render the tab's text or null if no specialized rendering is required**Since:** 1.4

### calculateTabHeight

protected int **calculateTabHeight**(int tabPlacement,  
 int tabIndex,  
 int fontHeight)

### calculateMaxTabHeight

protected int **calculateMaxTabHeight**(int tabPlacement)

### calculateTabWidth

protected int **calculateTabWidth**(int tabPlacement,  
 int tabIndex,  
 [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics)

### calculateMaxTabWidth

protected int **calculateMaxTabWidth**(int tabPlacement)

### calculateTabAreaHeight

protected int **calculateTabAreaHeight**(int tabPlacement,  
 int horizRunCount,  
 int maxTabHeight)

### calculateTabAreaWidth

protected int **calculateTabAreaWidth**(int tabPlacement,  
 int vertRunCount,  
 int maxTabWidth)

### getTabInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **getTabInsets**(int tabPlacement,  
 int tabIndex)

### getSelectedTabPadInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **getSelectedTabPadInsets**(int tabPlacement)

### getTabAreaInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **getTabAreaInsets**(int tabPlacement)

### getContentBorderInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **getContentBorderInsets**(int tabPlacement)

### getFontMetrics

protected [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) **getFontMetrics**()

### navigateSelectedTab

protected void **navigateSelectedTab**(int direction)

### selectNextTabInRun

protected void **selectNextTabInRun**(int current)

### selectPreviousTabInRun

protected void **selectPreviousTabInRun**(int current)

### selectNextTab

protected void **selectNextTab**(int current)

### selectPreviousTab

protected void **selectPreviousTab**(int current)

### selectAdjacentRunTab

protected void **selectAdjacentRunTab**(int tabPlacement,  
 int tabIndex,  
 int offset)

### getFocusIndex

protected int **getFocusIndex**()

Returns the index of the tab that has focus.

**Returns:**index of tab that has focus**Since:** 1.5

### getTabRunOffset

protected int **getTabRunOffset**(int tabPlacement,  
 int tabCount,  
 int tabIndex,  
 boolean forward)

### getPreviousTabIndex

protected int **getPreviousTabIndex**(int base)

### getNextTabIndex

protected int **getNextTabIndex**(int base)

### getNextTabIndexInRun

protected int **getNextTabIndexInRun**(int tabCount,  
 int base)

### getPreviousTabIndexInRun

protected int **getPreviousTabIndexInRun**(int tabCount,  
 int base)

### getPreviousTabRun

protected int **getPreviousTabRun**(int baseRun)

### getNextTabRun

protected int **getNextTabRun**(int baseRun)

### rotateInsets

protected static void **rotateInsets**([Insets](http://docs.google.com/java/awt/Insets.html) topInsets,  
 [Insets](http://docs.google.com/java/awt/Insets.html) targetInsets,  
 int targetPlacement)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicTabbedPaneUI.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/BasicSplitPaneUI.PropertyHandler.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.FocusHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicTabbedPaneUI.html)    [**NO FRAMES**](http://docs.google.com/BasicTabbedPaneUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#2grqrue) | [METHOD](#3fwokq0) |

[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).