| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicToolBarUI.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/BasicToolBarSeparatorUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DockingListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicToolBarUI.html)    [**NO FRAMES**](http://docs.google.com/BasicToolBarUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#3as4poj) | [METHOD](#49x2ik5) |

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

Class BasicToolBarUI

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

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

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

A Basic L&F implementation of ToolBarUI. This implementation is a "combined" view/controller.

| **Nested Class Summary** | |
| --- | --- |
| class | [**BasicToolBarUI.DockingListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DockingListener.html)            This class should be treated as a "protected" inner class. |
| protected  class | [**BasicToolBarUI.DragWindow**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DragWindow.html) |
| protected  class | [**BasicToolBarUI.FrameListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.FrameListener.html) |
| protected  class | [**BasicToolBarUI.PropertyListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.PropertyListener.html) |
| protected  class | [**BasicToolBarUI.ToolBarContListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.ToolBarContListener.html) |
| protected  class | [**BasicToolBarUI.ToolBarFocusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.ToolBarFocusListener.html) |

| **Field Summary** | |
| --- | --- |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**constraintBeforeFloating**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#constraintBeforeFloating) |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**dockingBorderColor**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dockingBorderColor) |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**dockingColor**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dockingColor) |
| protected  [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) | [**dockingListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dockingListener) |
| protected  [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**downKey**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#downKey)  **Deprecated.** *As of Java 2 platform v1.3.* |
| protected  [BasicToolBarUI.DragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DragWindow.html) | [**dragWindow**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dragWindow) |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**floatingBorderColor**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#floatingBorderColor) |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**floatingColor**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#floatingColor) |
| protected  int | [**focusedCompIndex**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#focusedCompIndex) |
| protected  [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**leftKey**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#leftKey)  **Deprecated.** *As of Java 2 platform v1.3.* |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**propertyListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#propertyListener) |
| protected  [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**rightKey**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#rightKey)  **Deprecated.** *As of Java 2 platform v1.3.* |
| protected  [JToolBar](http://docs.google.com/javax/swing/JToolBar.html) | [**toolBar**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#toolBar) |
| protected  [ContainerListener](http://docs.google.com/java/awt/event/ContainerListener.html) | [**toolBarContListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#toolBarContListener) |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**toolBarFocusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#toolBarFocusListener) |
| protected  [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**upKey**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.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** | |
| --- | --- |
| [**BasicToolBarUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#BasicToolBarUI())() |

| **Method Summary** | |
| --- | --- |
| boolean | [**canDock**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#canDock(java.awt.Component,%20java.awt.Point))([Component](http://docs.google.com/java/awt/Component.html) c, [Point](http://docs.google.com/java/awt/Point.html) p) |
| protected  [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) | [**createDockingListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createDockingListener())() |
| protected  [BasicToolBarUI.DragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DragWindow.html) | [**createDragWindow**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createDragWindow(javax.swing.JToolBar))([JToolBar](http://docs.google.com/javax/swing/JToolBar.html) toolbar) |
| protected  [JFrame](http://docs.google.com/javax/swing/JFrame.html) | [**createFloatingFrame**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createFloatingFrame(javax.swing.JToolBar))([JToolBar](http://docs.google.com/javax/swing/JToolBar.html) toolbar)            No longer used, use BasicToolBarUI.createFloatingWindow(JToolBar) |
| protected  [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) | [**createFloatingWindow**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createFloatingWindow(javax.swing.JToolBar))([JToolBar](http://docs.google.com/javax/swing/JToolBar.html) toolbar)            Creates a window which contains the toolbar after it has been dragged out from its container |
| protected  [WindowListener](http://docs.google.com/java/awt/event/WindowListener.html) | [**createFrameListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createFrameListener())() |
| protected  [Border](http://docs.google.com/javax/swing/border/Border.html) | [**createNonRolloverBorder**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createNonRolloverBorder())()            Creates the non rollover border for toolbar components. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createPropertyListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createPropertyListener())() |
| protected  [Border](http://docs.google.com/javax/swing/border/Border.html) | [**createRolloverBorder**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createRolloverBorder())()            Creates a rollover border for toolbar components. |
| protected  [ContainerListener](http://docs.google.com/java/awt/event/ContainerListener.html) | [**createToolBarContListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createToolBarContListener())() |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**createToolBarFocusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createToolBarFocusListener())() |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c) |
| protected  void | [**dragTo**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dragTo(java.awt.Point,%20java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) position, [Point](http://docs.google.com/java/awt/Point.html) origin) |
| protected  void | [**floatAt**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#floatAt(java.awt.Point,%20java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) position, [Point](http://docs.google.com/java/awt/Point.html) origin) |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getDockingColor**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#getDockingColor())()            Gets the color displayed when over a docking area |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getFloatingColor**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#getFloatingColor())()            Gets the color displayed when over a floating area |
| protected  [Border](http://docs.google.com/javax/swing/border/Border.html) | [**getNonRolloverBorder**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#getNonRolloverBorder(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b)            Returns a non-rollover border for the button. |
| protected  [Border](http://docs.google.com/javax/swing/border/Border.html) | [**getRolloverBorder**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#getRolloverBorder(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b)            Returns a rollover border for the button. |
| protected  void | [**installComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installComponents())() |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installDefaults())() |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installKeyboardActions())() |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installListeners())() |
| protected  void | [**installNonRolloverBorders**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installNonRolloverBorders(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Installs non-rollover borders on all the child components of the JComponent. |
| protected  void | [**installNormalBorders**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installNormalBorders(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Installs normal borders on all the child components of the JComponent. |
| protected  void | [**installRolloverBorders**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installRolloverBorders(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Installs rollover borders on all the child components of the JComponent. |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.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. |
| boolean | [**isFloating**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#isFloating())() |
| boolean | [**isRolloverBorders**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#isRolloverBorders())()            Returns a flag to determine whether rollover button borders are enabled. |
| protected  void | [**navigateFocusedComp**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#navigateFocusedComp(int))(int direction) |
| protected  void | [**paintDragWindow**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#paintDragWindow(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Paints the contents of the window used for dragging. |
| protected  void | [**setBorderToNonRollover**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToNonRollover(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Sets the border of the component to have a non-rollover border which was created by createNonRolloverBorder. |
| protected  void | [**setBorderToNormal**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToNormal(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Sets the border of the component to have a normal border. |
| protected  void | [**setBorderToRollover**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToRollover(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Sets the border of the component to have a rollover border which was created by createRolloverBorder. |
| void | [**setDockingColor**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setDockingColor(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) c)            Sets the color displayed when over a docking area |
| void | [**setFloating**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setFloating(boolean,%20java.awt.Point))(boolean b, [Point](http://docs.google.com/java/awt/Point.html) p) |
| void | [**setFloatingColor**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setFloatingColor(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) c)            Sets the color displayed when over a floating area |
| void | [**setFloatingLocation**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setFloatingLocation(int,%20int))(int x, int y) |
| void | [**setOrientation**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setOrientation(int))(int orientation) |
| void | [**setRolloverBorders**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setRolloverBorders(boolean))(boolean rollover)            Sets the flag for enabling rollover borders on the toolbar and it will also install the apropriate border depending on the state of the flag. |
| protected  void | [**uninstallComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#uninstallComponents())() |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#uninstallDefaults())() |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#uninstallKeyboardActions())() |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#uninstallListeners())() |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.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)), [getBaseline](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaseline(javax.swing.JComponent,%20int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaselineResizeBehavior(javax.swing.JComponent)), [getMaximumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)), [getMinimumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMinimumSize(javax.swing.JComponent)), [getPreferredSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent)), [paint](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.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** |
| --- |

### toolBar

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

### dragWindow

protected [BasicToolBarUI.DragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DragWindow.html) **dragWindow**

### focusedCompIndex

protected int **focusedCompIndex**

### dockingColor

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

### floatingColor

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

### dockingBorderColor

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

### floatingBorderColor

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

### dockingListener

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

### propertyListener

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

### toolBarContListener

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

### toolBarFocusListener

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

### constraintBeforeFloating

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

### 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.

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

### BasicToolBarUI

public **BasicToolBarUI**()

| **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())

### installDefaults

protected void **installDefaults**()

### uninstallDefaults

protected void **uninstallDefaults**()

### installComponents

protected void **installComponents**()

### uninstallComponents

protected void **uninstallComponents**()

### installListeners

protected void **installListeners**()

### uninstallListeners

protected void **uninstallListeners**()

### installKeyboardActions

protected void **installKeyboardActions**()

### uninstallKeyboardActions

protected void **uninstallKeyboardActions**()

### navigateFocusedComp

protected void **navigateFocusedComp**(int direction)

### createRolloverBorder

protected [Border](http://docs.google.com/javax/swing/border/Border.html) **createRolloverBorder**()

Creates a rollover border for toolbar components. The rollover border will be installed if rollover borders are enabled.

Override this method to provide an alternate rollover border.

**Since:** 1.4

### createNonRolloverBorder

protected [Border](http://docs.google.com/javax/swing/border/Border.html) **createNonRolloverBorder**()

Creates the non rollover border for toolbar components. This border will be installed as the border for components added to the toolbar if rollover borders are not enabled.

Override this method to provide an alternate rollover border.

**Since:** 1.4

### createFloatingFrame

protected [JFrame](http://docs.google.com/javax/swing/JFrame.html) **createFloatingFrame**([JToolBar](http://docs.google.com/javax/swing/JToolBar.html) toolbar)

No longer used, use BasicToolBarUI.createFloatingWindow(JToolBar)

**See Also:**[createFloatingWindow(javax.swing.JToolBar)](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createFloatingWindow(javax.swing.JToolBar))

### createFloatingWindow

protected [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **createFloatingWindow**([JToolBar](http://docs.google.com/javax/swing/JToolBar.html) toolbar)

Creates a window which contains the toolbar after it has been dragged out from its container

**Returns:**a RootPaneContainer object, containing the toolbar.**Since:** 1.4

### createDragWindow

protected [BasicToolBarUI.DragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DragWindow.html) **createDragWindow**([JToolBar](http://docs.google.com/javax/swing/JToolBar.html) toolbar)

### isRolloverBorders

public boolean **isRolloverBorders**()

Returns a flag to determine whether rollover button borders are enabled.

**Returns:**true if rollover borders are enabled; false otherwise**Since:** 1.4 **See Also:**[setRolloverBorders(boolean)](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setRolloverBorders(boolean))

### setRolloverBorders

public void **setRolloverBorders**(boolean rollover)

Sets the flag for enabling rollover borders on the toolbar and it will also install the apropriate border depending on the state of the flag.

**Parameters:**rollover - if true, rollover borders are installed. Otherwise non-rollover borders are installed**Since:** 1.4 **See Also:**[isRolloverBorders()](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#isRolloverBorders())

### installRolloverBorders

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

Installs rollover borders on all the child components of the JComponent.

This is a convenience method to call setBorderToRollover for each child component.

**Parameters:**c - container which holds the child components (usally a JToolBar)**Since:** 1.4 **See Also:**[setBorderToRollover(java.awt.Component)](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToRollover(java.awt.Component))

### installNonRolloverBorders

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

Installs non-rollover borders on all the child components of the JComponent. A non-rollover border is the border that is installed on the child component while it is in the toolbar.

This is a convenience method to call setBorderToNonRollover for each child component.

**Parameters:**c - container which holds the child components (usally a JToolBar)**Since:** 1.4 **See Also:**[setBorderToNonRollover(java.awt.Component)](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToNonRollover(java.awt.Component))

### installNormalBorders

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

Installs normal borders on all the child components of the JComponent. A normal border is the original border that was installed on the child component before it was added to the toolbar.

This is a convenience method to call setBorderNormal for each child component.

**Parameters:**c - container which holds the child components (usally a JToolBar)**Since:** 1.4 **See Also:**[setBorderToNonRollover(java.awt.Component)](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToNonRollover(java.awt.Component))

### setBorderToRollover

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

Sets the border of the component to have a rollover border which was created by createRolloverBorder.

**Parameters:**c - component which will have a rollover border installed**Since:** 1.4 **See Also:**[createRolloverBorder()](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createRolloverBorder())

### getRolloverBorder

protected [Border](http://docs.google.com/javax/swing/border/Border.html) **getRolloverBorder**([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b)

Returns a rollover border for the button.

**Parameters:**b - the button to calculate the rollover border for **Returns:**the rollover border**Since:** 1.6 **See Also:**[setBorderToRollover(java.awt.Component)](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToRollover(java.awt.Component))

### setBorderToNonRollover

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

Sets the border of the component to have a non-rollover border which was created by createNonRolloverBorder.

**Parameters:**c - component which will have a non-rollover border installed**Since:** 1.4 **See Also:**[createNonRolloverBorder()](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createNonRolloverBorder())

### getNonRolloverBorder

protected [Border](http://docs.google.com/javax/swing/border/Border.html) **getNonRolloverBorder**([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b)

Returns a non-rollover border for the button.

**Parameters:**b - the button to calculate the non-rollover border for **Returns:**the non-rollover border**Since:** 1.6 **See Also:**[setBorderToNonRollover(java.awt.Component)](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToNonRollover(java.awt.Component))

### setBorderToNormal

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

Sets the border of the component to have a normal border. A normal border is the original border that was installed on the child component before it was added to the toolbar.

**Parameters:**c - component which will have a normal border re-installed**Since:** 1.4 **See Also:**[createNonRolloverBorder()](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createNonRolloverBorder())

### setFloatingLocation

public void **setFloatingLocation**(int x,  
 int y)

### isFloating

public boolean **isFloating**()

### setFloating

public void **setFloating**(boolean b,  
 [Point](http://docs.google.com/java/awt/Point.html) p)

### setOrientation

public void **setOrientation**(int orientation)

### getDockingColor

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

Gets the color displayed when over a docking area

### setDockingColor

public void **setDockingColor**([Color](http://docs.google.com/java/awt/Color.html) c)

Sets the color displayed when over a docking area

### getFloatingColor

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

Gets the color displayed when over a floating area

### setFloatingColor

public void **setFloatingColor**([Color](http://docs.google.com/java/awt/Color.html) c)

Sets the color displayed when over a floating area

### canDock

public boolean **canDock**([Component](http://docs.google.com/java/awt/Component.html) c,  
 [Point](http://docs.google.com/java/awt/Point.html) p)

### dragTo

protected void **dragTo**([Point](http://docs.google.com/java/awt/Point.html) position,  
 [Point](http://docs.google.com/java/awt/Point.html) origin)

### floatAt

protected void **floatAt**([Point](http://docs.google.com/java/awt/Point.html) position,  
 [Point](http://docs.google.com/java/awt/Point.html) origin)

### createToolBarContListener

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

### createToolBarFocusListener

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

### createPropertyListener

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

### createDockingListener

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

### createFrameListener

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

### paintDragWindow

protected void **paintDragWindow**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Paints the contents of the window used for dragging.

**Parameters:**g - Graphics to paint to. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - is g is null**Since:** 1.5

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicToolBarUI.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/BasicToolBarSeparatorUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DockingListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicToolBarUI.html)    [**NO FRAMES**](http://docs.google.com/BasicToolBarUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#3as4poj) | [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).