| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MetalToolBarUI.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/metal/MetalToggleButtonUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.MetalContainerListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/metal/MetalToolBarUI.html)    [**NO FRAMES**](http://docs.google.com/MetalToolBarUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: [FIELD](#lnxbz9) | [CONSTR](#44sinio) | [METHOD](#z337ya) |

## **javax.swing.plaf.metal**

Class MetalToolBarUI

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

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

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

A Metal Look and Feel implementation of ToolBarUI. This implementation is a "combined" view/controller.

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**MetalToolBarUI.MetalContainerListener**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.MetalContainerListener.html) |
| protected  class | [**MetalToolBarUI.MetalDockingListener**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.MetalDockingListener.html) |
| protected  class | [**MetalToolBarUI.MetalRolloverListener**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.MetalRolloverListener.html) |

| **Nested classes/interfaces inherited from class javax.swing.plaf.basic.**[**BasicToolBarUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html) |
| --- |
| [BasicToolBarUI.DockingListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DockingListener.html), [BasicToolBarUI.DragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.DragWindow.html), [BasicToolBarUI.FrameListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.FrameListener.html), [BasicToolBarUI.PropertyListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.PropertyListener.html), [BasicToolBarUI.ToolBarContListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.ToolBarContListener.html), [BasicToolBarUI.ToolBarFocusListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.ToolBarFocusListener.html) |

| **Field Summary** | |
| --- | --- |
| protected  [ContainerListener](http://docs.google.com/java/awt/event/ContainerListener.html) | [**contListener**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#contListener)            This protected field is implemenation specific. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**rolloverListener**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#rolloverListener)            This protected field is implemenation specific. |

| **Fields inherited from class javax.swing.plaf.basic.**[**BasicToolBarUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html) |
| --- |
| [constraintBeforeFloating](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#constraintBeforeFloating), [dockingBorderColor](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dockingBorderColor), [dockingColor](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dockingColor), [dockingListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dockingListener), [downKey](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#downKey), [dragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dragWindow), [floatingBorderColor](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#floatingBorderColor), [floatingColor](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#floatingColor), [focusedCompIndex](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#focusedCompIndex), [leftKey](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#leftKey), [propertyListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#propertyListener), [rightKey](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#rightKey), [toolBar](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#toolBar), [toolBarContListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#toolBarContListener), [toolBarFocusListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#toolBarFocusListener), [upKey](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#upKey) |

| **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** | |
| --- | --- |
| [**MetalToolBarUI**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#MetalToolBarUI())() |

| **Method Summary** | |
| --- | --- |
| protected  [ContainerListener](http://docs.google.com/java/awt/event/ContainerListener.html) | [**createContainerListener**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#createContainerListener())()            Creates a container listener that will be added to the JToolBar. |
| protected  [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) | [**createDockingListener**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#createDockingListener())() |
| protected  [Border](http://docs.google.com/javax/swing/border/Border.html) | [**createNonRolloverBorder**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#createNonRolloverBorder())()            Creates the non rollover border for toolbar components. |
| protected  [Border](http://docs.google.com/javax/swing/border/Border.html) | [**createRolloverBorder**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#createRolloverBorder())()            Creates a rollover border for toolbar components. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createRolloverListener**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#createRolloverListener())()            Creates a property change listener that will be added to the JToolBar. |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c) |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#installListeners())() |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.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  void | [**setBorderToNonRollover**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.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 | [**setDragOffset**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#setDragOffset(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p) |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#uninstallListeners())() |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.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. |
| void | [**update**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#update(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)            If necessary paints the background of the component, then invokes paint. |

| **Methods inherited from class javax.swing.plaf.basic.**[**BasicToolBarUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html) |
| --- |
| [canDock](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#canDock(java.awt.Component,%20java.awt.Point)), [createDragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createDragWindow(javax.swing.JToolBar)), [createFloatingFrame](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createFloatingFrame(javax.swing.JToolBar)), [createFloatingWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createFloatingWindow(javax.swing.JToolBar)), [createFrameListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createFrameListener()), [createPropertyListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createPropertyListener()), [createToolBarContListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createToolBarContListener()), [createToolBarFocusListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createToolBarFocusListener()), [dragTo](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#dragTo(java.awt.Point,%20java.awt.Point)), [floatAt](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#floatAt(java.awt.Point,%20java.awt.Point)), [getDockingColor](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#getDockingColor()), [getFloatingColor](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#getFloatingColor()), [getNonRolloverBorder](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#getNonRolloverBorder(javax.swing.AbstractButton)), [getRolloverBorder](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#getRolloverBorder(javax.swing.AbstractButton)), [installComponents](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installComponents()), [installDefaults](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installDefaults()), [installKeyboardActions](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installKeyboardActions()), [installNonRolloverBorders](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installNonRolloverBorders(javax.swing.JComponent)), [installNormalBorders](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installNormalBorders(javax.swing.JComponent)), [installRolloverBorders](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installRolloverBorders(javax.swing.JComponent)), [isFloating](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#isFloating()), [isRolloverBorders](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#isRolloverBorders()), [navigateFocusedComp](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#navigateFocusedComp(int)), [paintDragWindow](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#paintDragWindow(java.awt.Graphics)), [setBorderToNormal](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToNormal(java.awt.Component)), [setBorderToRollover](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToRollover(java.awt.Component)), [setDockingColor](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setDockingColor(java.awt.Color)), [setFloating](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setFloating(boolean,%20java.awt.Point)), [setFloatingColor](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setFloatingColor(java.awt.Color)), [setFloatingLocation](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setFloatingLocation(int,%20int)), [setOrientation](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setOrientation(int)), [setRolloverBorders](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setRolloverBorders(boolean)), [uninstallComponents](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#uninstallComponents()), [uninstallDefaults](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#uninstallDefaults()), [uninstallKeyboardActions](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#uninstallKeyboardActions()) |

| **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)) |

| **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** |
| --- |

### contListener

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

This protected field is implemenation specific. Do not access directly or override. Use the create method instead.

**See Also:**[createContainerListener()](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#createContainerListener())

### rolloverListener

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

This protected field is implemenation specific. Do not access directly or override. Use the create method instead.

**See Also:**[createRolloverListener()](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.html#createRolloverListener())

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

### MetalToolBarUI

public **MetalToolBarUI**()

| **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/basic/BasicToolBarUI.html#installUI(javax.swing.JComponent)) in class [BasicToolBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.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/basic/BasicToolBarUI.html#uninstallUI(javax.swing.JComponent)) in class [BasicToolBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.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())

### installListeners

protected void **installListeners**()

**Overrides:**[installListeners](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#installListeners()) in class [BasicToolBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html)

### uninstallListeners

protected void **uninstallListeners**()

**Overrides:**[uninstallListeners](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#uninstallListeners()) in class [BasicToolBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html)

### createRolloverBorder

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

**Description copied from class:** [**BasicToolBarUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.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.

**Overrides:**[createRolloverBorder](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createRolloverBorder()) in class [BasicToolBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html)

### createNonRolloverBorder

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

**Description copied from class:** [**BasicToolBarUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.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.

**Overrides:**[createNonRolloverBorder](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createNonRolloverBorder()) in class [BasicToolBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html)

### setBorderToNonRollover

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

**Description copied from class:** [**BasicToolBarUI**](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToNonRollover(java.awt.Component)) Sets the border of the component to have a non-rollover border which was created by createNonRolloverBorder.

**Overrides:**[setBorderToNonRollover](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#setBorderToNonRollover(java.awt.Component)) in class [BasicToolBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html) **Parameters:**c - component which will have a non-rollover border installed**See Also:**[BasicToolBarUI.createNonRolloverBorder()](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createNonRolloverBorder())

### createContainerListener

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

Creates a container listener that will be added to the JToolBar. If this method returns null then it will not be added to the toolbar.

**Returns:**an instance of a ContainerListener or null

### createRolloverListener

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

Creates a property change listener that will be added to the JToolBar. If this method returns null then it will not be added to the toolbar.

**Returns:**an instance of a PropertyChangeListener or null

### createDockingListener

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

**Overrides:**[createDockingListener](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html#createDockingListener()) in class [BasicToolBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicToolBarUI.html)

### setDragOffset

protected void **setDragOffset**([Point](http://docs.google.com/java/awt/Point.html) p)

### update

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

If necessary paints the background of the component, then invokes paint.

**Overrides:**[update](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**g - Graphics to paint toc - JComponent painting on **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if g or c is null**Since:** 1.5 **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)), [ComponentUI.paint(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MetalToolBarUI.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/metal/MetalToggleButtonUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/metal/MetalToolBarUI.MetalContainerListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/metal/MetalToolBarUI.html)    [**NO FRAMES**](http://docs.google.com/MetalToolBarUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: [FIELD](#lnxbz9) | [CONSTR](#44sinio) | [METHOD](#z337ya) |

[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).