| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicRootPaneUI.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/BasicRadioButtonUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicScrollBarUI.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicRootPaneUI.html)    [**NO FRAMES**](http://docs.google.com/BasicRootPaneUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

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

Class BasicRootPaneUI

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

**All Implemented Interfaces:** [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html), [EventListener](http://docs.google.com/java/util/EventListener.html) **Direct Known Subclasses:** [MetalRootPaneUI](http://docs.google.com/javax/swing/plaf/metal/MetalRootPaneUI.html)

public class **BasicRootPaneUI**extends [RootPaneUI](http://docs.google.com/javax/swing/plaf/RootPaneUI.html)implements [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html)

Basic implementation of RootPaneUI, there is one shared between all JRootPane instances.

**Since:** 1.3

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

| **Method Summary** | |
| --- | --- |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c) |
| protected  void | [**installComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#installComponents(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root) |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#installDefaults(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) c) |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#installKeyboardActions(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root) |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#installListeners(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root) |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#installUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Configures the specified component appropriate for the look and feel. |
| void | [**propertyChange**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#propertyChange(java.beans.PropertyChangeEvent))([PropertyChangeEvent](http://docs.google.com/java/beans/PropertyChangeEvent.html) e)            Invoked when a property changes on the root pane. |
| protected  void | [**uninstallComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#uninstallComponents(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root) |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#uninstallDefaults(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root) |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#uninstallKeyboardActions(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root) |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.html#uninstallListeners(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root) |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicRootPaneUI.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)) |

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

### BasicRootPaneUI

public **BasicRootPaneUI**()

| **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**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) c)

### installComponents

protected void **installComponents**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)

### installListeners

protected void **installListeners**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)

### installKeyboardActions

protected void **installKeyboardActions**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)

### uninstallDefaults

protected void **uninstallDefaults**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)

### uninstallComponents

protected void **uninstallComponents**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)

### uninstallListeners

protected void **uninstallListeners**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)

### uninstallKeyboardActions

protected void **uninstallKeyboardActions**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)

### propertyChange

public void **propertyChange**([PropertyChangeEvent](http://docs.google.com/java/beans/PropertyChangeEvent.html) e)

Invoked when a property changes on the root pane. If the event indicates the defaultButton has changed, this will reinstall the keyboard actions.

**Specified by:**[propertyChange](http://docs.google.com/java/beans/PropertyChangeListener.html#propertyChange(java.beans.PropertyChangeEvent)) in interface [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **Parameters:**e - A PropertyChangeEvent object describing the event source and the property that has changed.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicRootPaneUI.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/BasicRadioButtonUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicScrollBarUI.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicRootPaneUI.html)    [**NO FRAMES**](http://docs.google.com/BasicRootPaneUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

[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).