| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicLabelUI.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/BasicInternalFrameUI.InternalFramePropertyChangeListener.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicLabelUI.html)    [**NO FRAMES**](http://docs.google.com/BasicLabelUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

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

Class BasicLabelUI

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

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

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

A Windows L&F implementation of LabelUI. This implementation is completely static, i.e. there's only one UIView implementation that's shared by all JLabel objects.

| **Field Summary** | |
| --- | --- |
| protected static [BasicLabelUI](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html) | [**labelUI**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#labelUI)            The default BasicLabelUI instance. |

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

| **Method Summary** | |
| --- | --- |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c) |
| int | [**getBaseline**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#getBaseline(javax.swing.JComponent,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c, int width, int height)            Returns the baseline. |
| [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) | [**getBaselineResizeBehavior**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.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. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.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/BasicLabelUI.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. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#getPreferredSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the specified component's preferred size appropriate for the look and feel. |
| protected  void | [**installComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#installComponents(javax.swing.JLabel))([JLabel](http://docs.google.com/javax/swing/JLabel.html) c) |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#installDefaults(javax.swing.JLabel))([JLabel](http://docs.google.com/javax/swing/JLabel.html) c) |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#installKeyboardActions(javax.swing.JLabel))([JLabel](http://docs.google.com/javax/swing/JLabel.html) l) |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#installListeners(javax.swing.JLabel))([JLabel](http://docs.google.com/javax/swing/JLabel.html) c) |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.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  [String](http://docs.google.com/java/lang/String.html) | [**layoutCL**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#layoutCL(javax.swing.JLabel,%20java.awt.FontMetrics,%20java.lang.String,%20javax.swing.Icon,%20java.awt.Rectangle,%20java.awt.Rectangle,%20java.awt.Rectangle))([JLabel](http://docs.google.com/javax/swing/JLabel.html) label, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fontMetrics, [String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewR, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconR, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textR)            Forwards the call to SwingUtilities.layoutCompoundLabel(). |
| void | [**paint**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.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)            Paint the label text in the foreground color, if the label is opaque then paint the entire background with the background color. |
| protected  void | [**paintDisabledText**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#paintDisabledText(javax.swing.JLabel,%20java.awt.Graphics,%20java.lang.String,%20int,%20int))([JLabel](http://docs.google.com/javax/swing/JLabel.html) l, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, [String](http://docs.google.com/java/lang/String.html) s, int textX, int textY)            Paint clippedText at textX, textY with background.lighter() and then shifted down and to the right by one pixel with background.darker(). |
| protected  void | [**paintEnabledText**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#paintEnabledText(javax.swing.JLabel,%20java.awt.Graphics,%20java.lang.String,%20int,%20int))([JLabel](http://docs.google.com/javax/swing/JLabel.html) l, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, [String](http://docs.google.com/java/lang/String.html) s, int textX, int textY)            Paint clippedText at textX, textY with the labels foreground color. |
| void | [**propertyChange**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#propertyChange(java.beans.PropertyChangeEvent))([PropertyChangeEvent](http://docs.google.com/java/beans/PropertyChangeEvent.html) e)            This method gets called when a bound property is changed. |
| protected  void | [**uninstallComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#uninstallComponents(javax.swing.JLabel))([JLabel](http://docs.google.com/javax/swing/JLabel.html) c) |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#uninstallDefaults(javax.swing.JLabel))([JLabel](http://docs.google.com/javax/swing/JLabel.html) c) |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#uninstallKeyboardActions(javax.swing.JLabel))([JLabel](http://docs.google.com/javax/swing/JLabel.html) c) |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#uninstallListeners(javax.swing.JLabel))([JLabel](http://docs.google.com/javax/swing/JLabel.html) c) |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.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)), [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** |
| --- |

### labelUI

protected static [BasicLabelUI](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html) **labelUI**

The default BasicLabelUI instance. This field might not be used. To change the default instance use a subclass which overrides the createUI method, and place that class name in defaults table under the key "LabelUI".

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

### BasicLabelUI

public **BasicLabelUI**()

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

### layoutCL

protected [String](http://docs.google.com/java/lang/String.html) **layoutCL**([JLabel](http://docs.google.com/javax/swing/JLabel.html) label,  
 [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fontMetrics,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewR,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconR,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textR)

Forwards the call to SwingUtilities.layoutCompoundLabel(). This method is here so that a subclass could do Label specific layout and to shorten the method name a little.

**See Also:**[SwingUtilities.layoutCompoundLabel(javax.swing.JComponent, java.awt.FontMetrics, java.lang.String, javax.swing.Icon, int, int, int, int, java.awt.Rectangle, java.awt.Rectangle, java.awt.Rectangle, int)](http://docs.google.com/javax/swing/SwingUtilities.html#layoutCompoundLabel(javax.swing.JComponent,%20java.awt.FontMetrics,%20java.lang.String,%20javax.swing.Icon,%20int,%20int,%20int,%20int,%20java.awt.Rectangle,%20java.awt.Rectangle,%20java.awt.Rectangle,%20int))

### paintEnabledText

protected void **paintEnabledText**([JLabel](http://docs.google.com/javax/swing/JLabel.html) l,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [String](http://docs.google.com/java/lang/String.html) s,  
 int textX,  
 int textY)

Paint clippedText at textX, textY with the labels foreground color.

**See Also:**[paint(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)), [paintDisabledText(javax.swing.JLabel, java.awt.Graphics, java.lang.String, int, int)](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#paintDisabledText(javax.swing.JLabel,%20java.awt.Graphics,%20java.lang.String,%20int,%20int))

### paintDisabledText

protected void **paintDisabledText**([JLabel](http://docs.google.com/javax/swing/JLabel.html) l,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [String](http://docs.google.com/java/lang/String.html) s,  
 int textX,  
 int textY)

Paint clippedText at textX, textY with background.lighter() and then shifted down and to the right by one pixel with background.darker().

**See Also:**[paint(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)), [paintEnabledText(javax.swing.JLabel, java.awt.Graphics, java.lang.String, int, int)](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#paintEnabledText(javax.swing.JLabel,%20java.awt.Graphics,%20java.lang.String,%20int,%20int))

### paint

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

Paint the label text in the foreground color, if the label is opaque then paint the entire background with the background color. The Label text is drawn by paintEnabledText() or paintDisabledText(). The locations of the label parts are computed by layoutCL.

**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:**[paintEnabledText(javax.swing.JLabel, java.awt.Graphics, java.lang.String, int, int)](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#paintEnabledText(javax.swing.JLabel,%20java.awt.Graphics,%20java.lang.String,%20int,%20int)), [paintDisabledText(javax.swing.JLabel, java.awt.Graphics, java.lang.String, int, int)](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#paintDisabledText(javax.swing.JLabel,%20java.awt.Graphics,%20java.lang.String,%20int,%20int)), [layoutCL(javax.swing.JLabel, java.awt.FontMetrics, java.lang.String, javax.swing.Icon, java.awt.Rectangle, java.awt.Rectangle, java.awt.Rectangle)](http://docs.google.com/javax/swing/plaf/basic/BasicLabelUI.html#layoutCL(javax.swing.JLabel,%20java.awt.FontMetrics,%20java.lang.String,%20javax.swing.Icon,%20java.awt.Rectangle,%20java.awt.Rectangle,%20java.awt.Rectangle))

### getPreferredSize

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

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent)) Returns the specified component's preferred size appropriate for the look and feel. If null is returned, the preferred 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 returns null.

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

### 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:**getPreferredSize(c)**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:**getPreferredSize(c)**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))

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

### installListeners

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

### installComponents

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

### installKeyboardActions

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

### uninstallDefaults

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

### uninstallListeners

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

### uninstallComponents

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

### uninstallKeyboardActions

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

### createUI

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

### propertyChange

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

**Description copied from interface:** [**PropertyChangeListener**](http://docs.google.com/java/beans/PropertyChangeListener.html#propertyChange(java.beans.PropertyChangeEvent)) This method gets called when a bound property is changed.

**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/BasicLabelUI.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/BasicInternalFrameUI.InternalFramePropertyChangeListener.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicLabelUI.html)    [**NO FRAMES**](http://docs.google.com/BasicLabelUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).