| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicButtonUI.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/BasicButtonListener.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicCheckBoxMenuItemUI.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicButtonUI.html)    [**NO FRAMES**](http://docs.google.com/BasicButtonUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

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

Class BasicButtonUI

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

**Direct Known Subclasses:** [BasicToggleButtonUI](http://docs.google.com/javax/swing/plaf/basic/BasicToggleButtonUI.html), [MetalButtonUI](http://docs.google.com/javax/swing/plaf/metal/MetalButtonUI.html)

public class **BasicButtonUI**extends [ButtonUI](http://docs.google.com/javax/swing/plaf/ButtonUI.html)

BasicButton implementation

| **Field Summary** | |
| --- | --- |
| protected  int | [**defaultTextIconGap**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#defaultTextIconGap) |
| protected  int | [**defaultTextShiftOffset**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#defaultTextShiftOffset) |

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

| **Method Summary** | |
| --- | --- |
| protected  void | [**clearTextShiftOffset**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#clearTextShiftOffset())() |
| protected  [BasicButtonListener](http://docs.google.com/javax/swing/plaf/basic/BasicButtonListener.html) | [**createButtonListener**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#createButtonListener(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.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/BasicButtonUI.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/BasicButtonUI.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. |
| int | [**getDefaultTextIconGap**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#getDefaultTextIconGap(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.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/BasicButtonUI.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/BasicButtonUI.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  [String](http://docs.google.com/java/lang/String.html) | [**getPropertyPrefix**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#getPropertyPrefix())() |
| protected  int | [**getTextShiftOffset**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#getTextShiftOffset())() |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#installDefaults(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#installKeyboardActions(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#installListeners(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.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 | [**paint**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Paints the specified component appropriate for the look and feel. |
| protected  void | [**paintButtonPressed**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#paintButtonPressed(java.awt.Graphics,%20javax.swing.AbstractButton))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| protected  void | [**paintFocus**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#paintFocus(java.awt.Graphics,%20javax.swing.AbstractButton,%20java.awt.Rectangle,%20java.awt.Rectangle,%20java.awt.Rectangle))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewRect, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect) |
| protected  void | [**paintIcon**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#paintIcon(java.awt.Graphics,%20javax.swing.JComponent,%20java.awt.Rectangle))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [JComponent](http://docs.google.com/javax/swing/JComponent.html) c, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect) |
| protected  void | [**paintText**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#paintText(java.awt.Graphics,%20javax.swing.AbstractButton,%20java.awt.Rectangle,%20java.lang.String))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect, [String](http://docs.google.com/java/lang/String.html) text)            Method which renders the text of the current button. |
| protected  void | [**paintText**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#paintText(java.awt.Graphics,%20javax.swing.JComponent,%20java.awt.Rectangle,%20java.lang.String))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [JComponent](http://docs.google.com/javax/swing/JComponent.html) c, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect, [String](http://docs.google.com/java/lang/String.html) text)            As of Java 2 platform v 1.4 this method should not be used or overriden. |
| protected  void | [**setTextShiftOffset**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#setTextShiftOffset())() |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#uninstallDefaults(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#uninstallKeyboardActions(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.html#uninstallListeners(javax.swing.AbstractButton))([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b) |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonUI.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** |
| --- |

### defaultTextIconGap

protected int **defaultTextIconGap**

### defaultTextShiftOffset

protected int **defaultTextShiftOffset**

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

### BasicButtonUI

public **BasicButtonUI**()

| **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)

### getPropertyPrefix

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

### 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())

### installDefaults

protected void **installDefaults**([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b)

### installListeners

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

### installKeyboardActions

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

### 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())

### uninstallKeyboardActions

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

### uninstallListeners

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

### uninstallDefaults

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

### createButtonListener

protected [BasicButtonListener](http://docs.google.com/javax/swing/plaf/basic/BasicButtonListener.html) **createButtonListener**([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b)

### getDefaultTextIconGap

public int **getDefaultTextIconGap**([AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b)

### paint

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

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) Paints the specified component appropriate for the look and feel. This method is invoked from the ComponentUI.update method when the specified component is being painted. Subclasses should override this method and use the specified Graphics object to render the content of the component.

**Overrides:**[paint](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**g - the Graphics context in which to paintc - the component being painted; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.update(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent))

### paintIcon

protected void **paintIcon**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect)

### paintText

protected void **paintText**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect,  
 [String](http://docs.google.com/java/lang/String.html) text)

As of Java 2 platform v 1.4 this method should not be used or overriden. Use the paintText method which takes the AbstractButton argument.

### paintText

protected void **paintText**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect,  
 [String](http://docs.google.com/java/lang/String.html) text)

Method which renders the text of the current button.

**Parameters:**g - Graphics contextb - Current button to rendertextRect - Bounding rectangle to render the text.text - String to render**Since:** 1.4

### paintFocus

protected void **paintFocus**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewRect,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textRect,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconRect)

### paintButtonPressed

protected void **paintButtonPressed**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) b)

### clearTextShiftOffset

protected void **clearTextShiftOffset**()

### setTextShiftOffset

protected void **setTextShiftOffset**()

### getTextShiftOffset

protected int **getTextShiftOffset**()

### getMinimumSize

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

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

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

### 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))

### getMaximumSize

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

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

**Overrides:**[getMaximumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose maximum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**a Dimension object or null**See Also:**[JComponent.getMaximumSize()](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [LayoutManager2.maximumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager2.html#maximumLayoutSize(java.awt.Container))

### getBaseline

public int **getBaseline**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 int width,  
 int height)

Returns the baseline.

**Overrides:**[getBaseline](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaseline(javax.swing.JComponent,%20int,%20int)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - JComponent baseline is being requested forwidth - the width to get the baseline forheight - the height to get the baseline for **Returns:**baseline or a value < 0 indicating there is no reasonable baseline **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if c is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if width or height is < 0**Since:** 1.6 **See Also:**[JComponent.getBaseline(int, int)](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int))

### getBaselineResizeBehavior

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

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

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicButtonUI.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/BasicButtonListener.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicCheckBoxMenuItemUI.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicButtonUI.html)    [**NO FRAMES**](http://docs.google.com/BasicButtonUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

[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).