| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicListUI.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/BasicLabelUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.FocusHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicListUI.html)    [**NO FRAMES**](http://docs.google.com/BasicListUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#1pxezwc) | [METHOD](#2p2csry) |

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

Class BasicListUI

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

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

An extensible implementation of ListUI.

BasicListUI instances cannot be shared between multiple lists.

| **Nested Class Summary** | |
| --- | --- |
| class | [**BasicListUI.FocusHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.FocusHandler.html)            This inner class is marked "public" due to a compiler bug. |
| class | [**BasicListUI.ListDataHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.ListDataHandler.html)            The ListDataListener that's added to the JLists model at installUI time, and whenever the JList.model property changes. |
| class | [**BasicListUI.ListSelectionHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.ListSelectionHandler.html)            The ListSelectionListener that's added to the JLists selection model at installUI time, and whenever the JList.selectionModel property changes. |
| class | [**BasicListUI.MouseInputHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.MouseInputHandler.html)            Mouse input, and focus handling for JList. |
| class | [**BasicListUI.PropertyChangeHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.PropertyChangeHandler.html)            The PropertyChangeListener that's added to the JList at installUI time. |

| **Field Summary** | |
| --- | --- |
| protected  int | [**cellHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#cellHeight) |
| protected  int[] | [**cellHeights**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#cellHeights) |
| protected static int | [**cellRendererChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#cellRendererChanged) |
| protected  int | [**cellWidth**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#cellWidth) |
| protected static int | [**fixedCellHeightChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#fixedCellHeightChanged) |
| protected static int | [**fixedCellWidthChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#fixedCellWidthChanged) |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**focusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#focusListener) |
| protected static int | [**fontChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#fontChanged) |
| protected  [JList](http://docs.google.com/javax/swing/JList.html) | [**list**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#list) |
| protected  [ListDataListener](http://docs.google.com/javax/swing/event/ListDataListener.html) | [**listDataListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#listDataListener) |
| protected  [ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html) | [**listSelectionListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#listSelectionListener) |
| protected static int | [**modelChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#modelChanged) |
| protected  [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) | [**mouseInputListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#mouseInputListener) |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**propertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#propertyChangeListener) |
| protected static int | [**prototypeCellValueChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#prototypeCellValueChanged) |
| protected  [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html) | [**rendererPane**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#rendererPane) |
| protected static int | [**selectionModelChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#selectionModelChanged) |
| protected  int | [**updateLayoutStateNeeded**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#updateLayoutStateNeeded) |

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

| **Method Summary** | |
| --- | --- |
| protected  int | [**convertRowToY**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#convertRowToY(int))(int row)            Return the JList relative Y coordinate of the origin of the specified row or -1 if row isn't valid. |
| protected  int | [**convertYToRow**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#convertYToRow(int))(int y0)            Convert the JList relative coordinate to the row that contains it, based on the current layout. |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**createFocusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#createFocusListener())() |
| protected  [ListDataListener](http://docs.google.com/javax/swing/event/ListDataListener.html) | [**createListDataListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#createListDataListener())()            Creates an instance of ListDataListener that's added to the JLists by model as needed. |
| protected  [ListSelectionListener](http://docs.google.com/javax/swing/event/ListSelectionListener.html) | [**createListSelectionListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#createListSelectionListener())()            Creates an instance of ListSelectionHandler that's added to the JLists by selectionModel as needed. |
| protected  [MouseInputListener](http://docs.google.com/javax/swing/event/MouseInputListener.html) | [**createMouseInputListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#createMouseInputListener())()            Creates a delegate that implements MouseInputListener. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createPropertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#createPropertyChangeListener())()            Creates an instance of PropertyChangeHandler that's added to the JList by installUI(). |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) list)            Returns a new instance of BasicListUI. |
| int | [**getBaseline**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.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/BasicListUI.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. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getCellBounds**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#getCellBounds(javax.swing.JList,%20int,%20int))([JList](http://docs.google.com/javax/swing/JList.html) list, int index1, int index2)            Returns the bounding rectangle, in the given list's coordinate system, for the range of cells specified by the two indices. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#getPreferredSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            The preferredSize of the list depends upon the layout orientation. |
| protected  int | [**getRowHeight**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#getRowHeight(int))(int row)            Returns the height of the specified row based on the current layout. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**indexToLocation**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#indexToLocation(javax.swing.JList,%20int))([JList](http://docs.google.com/javax/swing/JList.html) list, int index)            Returns the origin in the given JList, of the specified item, in the list's coordinate system. |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installDefaults())()            Initialize JList properties, e.g. |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installKeyboardActions())()            Registers the keyboard bindings on the JList that the BasicListUI is associated with. |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installListeners())()            Create and install the listeners for the JList, its model, and its selectionModel. |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Initializes this.list by calling installDefaults(), installListeners(), and installKeyboardActions() in order. |
| int | [**locationToIndex**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#locationToIndex(javax.swing.JList,%20java.awt.Point))([JList](http://docs.google.com/javax/swing/JList.html) list, [Point](http://docs.google.com/java/awt/Point.html) location)            Returns the cell index in the specified JList closest to the given location in the list's coordinate system. |
| protected  void | [**maybeUpdateLayoutState**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#maybeUpdateLayoutState())()            If updateLayoutStateNeeded is non zero, call updateLayoutState() and reset updateLayoutStateNeeded. |
| void | [**paint**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.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 rows that intersect the Graphics objects clipRect. |
| protected  void | [**paintCell**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#paintCell(java.awt.Graphics,%20int,%20java.awt.Rectangle,%20javax.swing.ListCellRenderer,%20javax.swing.ListModel,%20javax.swing.ListSelectionModel,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int row, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) rowBounds, [ListCellRenderer](http://docs.google.com/javax/swing/ListCellRenderer.html) cellRenderer, [ListModel](http://docs.google.com/javax/swing/ListModel.html) dataModel, [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) selModel, int leadIndex)            Paint one List cell: compute the relevant state, get the "rubber stamp" cell renderer component, and then use the CellRendererPane to paint it. |
| protected  void | [**selectNextIndex**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#selectNextIndex())()            Selected the previous row and force it to be visible. |
| protected  void | [**selectPreviousIndex**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#selectPreviousIndex())()            Selected the previous row and force it to be visible. |
| protected  void | [**uninstallDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallDefaults())()            Set the JList properties that haven't been explicitly overridden to null. |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallKeyboardActions())()            Unregisters keyboard actions installed from installKeyboardActions. |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallListeners())()            Remove the listeners for the JList, its model, and its selectionModel. |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Uninitializes this.list by calling uninstallListeners(), uninstallKeyboardActions(), and uninstallDefaults() in order. |
| protected  void | [**updateLayoutState**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#updateLayoutState())()            Recompute the value of cellHeight or cellHeights based and cellWidth, based on the current font and the current values of fixedCellWidth, fixedCellHeight, and prototypeCellValue. |

| **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)), [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)), [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** |
| --- |

### list

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

### rendererPane

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

### focusListener

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

### mouseInputListener

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

### listSelectionListener

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

### listDataListener

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

### propertyChangeListener

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

### cellHeights

protected int[] **cellHeights**

### cellHeight

protected int **cellHeight**

### cellWidth

protected int **cellWidth**

### updateLayoutStateNeeded

protected int **updateLayoutStateNeeded**

### modelChanged

protected static final int **modelChanged**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicListUI.modelChanged)

### selectionModelChanged

protected static final int **selectionModelChanged**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicListUI.selectionModelChanged)

### fontChanged

protected static final int **fontChanged**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicListUI.fontChanged)

### fixedCellWidthChanged

protected static final int **fixedCellWidthChanged**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicListUI.fixedCellWidthChanged)

### fixedCellHeightChanged

protected static final int **fixedCellHeightChanged**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicListUI.fixedCellHeightChanged)

### prototypeCellValueChanged

protected static final int **prototypeCellValueChanged**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicListUI.prototypeCellValueChanged)

### cellRendererChanged

protected static final int **cellRendererChanged**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicListUI.cellRendererChanged)

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

### BasicListUI

public **BasicListUI**()

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

### paintCell

protected void **paintCell**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int row,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) rowBounds,  
 [ListCellRenderer](http://docs.google.com/javax/swing/ListCellRenderer.html) cellRenderer,  
 [ListModel](http://docs.google.com/javax/swing/ListModel.html) dataModel,  
 [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html) selModel,  
 int leadIndex)

Paint one List cell: compute the relevant state, get the "rubber stamp" cell renderer component, and then use the CellRendererPane to paint it. Subclasses may want to override this method rather than paint().

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

### 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 rows that intersect the Graphics objects clipRect. This method calls paintCell as necessary. Subclasses may want to override these methods.

**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:**[paintCell(java.awt.Graphics, int, java.awt.Rectangle, javax.swing.ListCellRenderer, javax.swing.ListModel, javax.swing.ListSelectionModel, int)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#paintCell(java.awt.Graphics,%20int,%20java.awt.Rectangle,%20javax.swing.ListCellRenderer,%20javax.swing.ListModel,%20javax.swing.ListSelectionModel,%20int))

### 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))

### getPreferredSize

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

The preferredSize of the list depends upon the layout orientation.

| Layout Orientation | Preferred Size |
| --- | --- |
| JList.VERTICAL | The preferredSize of the list is total height of the rows and the maximum width of the cells. If JList.fixedCellHeight is specified then the total height of the rows is just (cellVerticalMargins + fixedCellHeight) \* model.getSize() where rowVerticalMargins is the space we allocate for drawing the yellow focus outline. Similarly if fixedCellWidth is specified then we just use that. |
| JList.VERTICAL\_WRAP | If the visible row count is greater than zero, the preferredHeight is the maximum cell height \* visibleRowCount. If the visible row count is <= 0, the preferred height is either the current height of the list, or the maximum cell height, whichever is bigger. The preferred width is than the maximum cell width \* number of columns needed. Where the number of columns needs is list.height / max cell height. Max cell height is either the fixed cell height, or is determined by iterating through all the cells to find the maximum height from the ListCellRenderer. |
| JList.HORIZONTAL\_WRAP | If the visible row count is greater than zero, the preferredHeight is the maximum cell height \* adjustedRowCount. Where visibleRowCount is used to determine the number of columns. Because this lays out horizontally the number of rows is then determined from the column count. For example, lets say you have a model with 10 items and the visible row count is 8. The number of columns needed to display this is 2, but you no longer need 8 rows to display this, you only need 5, thus the adjustedRowCount is 5.  If the visible row count is <= 0, the preferred height is dictated by the number of columns, which will be as many as can fit in the width of the JList (width / max cell width), with at least one column. The preferred height then becomes the model size / number of columns \* maximum cell height. Max cell height is either the fixed cell height, or is determined by iterating through all the cells to find the maximum height from the ListCellRenderer. |

The above specifies the raw preferred width and height. The resulting preferred width is the above width + insets.left + insets.right and the resulting preferred height is the above height + insets.top + insets.bottom. Where the Insets are determined from list.getInsets().

**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 JList component. **Returns:**The total size of the list.**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))

### selectPreviousIndex

protected void **selectPreviousIndex**()

Selected the previous row and force it to be visible.

**See Also:**[JList.ensureIndexIsVisible(int)](http://docs.google.com/javax/swing/JList.html#ensureIndexIsVisible(int))

### selectNextIndex

protected void **selectNextIndex**()

Selected the previous row and force it to be visible.

**See Also:**[JList.ensureIndexIsVisible(int)](http://docs.google.com/javax/swing/JList.html#ensureIndexIsVisible(int))

### installKeyboardActions

protected void **installKeyboardActions**()

Registers the keyboard bindings on the JList that the BasicListUI is associated with. This method is called at installUI() time.

**See Also:**[installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.swing.JComponent))

### uninstallKeyboardActions

protected void **uninstallKeyboardActions**()

Unregisters keyboard actions installed from installKeyboardActions. This method is called at uninstallUI() time - subclassess should ensure that all of the keyboard actions registered at installUI time are removed here.

**See Also:**[installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.swing.JComponent))

### installListeners

protected void **installListeners**()

Create and install the listeners for the JList, its model, and its selectionModel. This method is called at installUI() time.

**See Also:**[installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.swing.JComponent)), [uninstallListeners()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallListeners())

### uninstallListeners

protected void **uninstallListeners**()

Remove the listeners for the JList, its model, and its selectionModel. All of the listener fields, are reset to null here. This method is called at uninstallUI() time, it should be kept in sync with installListeners.

**See Also:**[uninstallUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallUI(javax.swing.JComponent)), [installListeners()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installListeners())

### installDefaults

protected void **installDefaults**()

Initialize JList properties, e.g. font, foreground, and background, and add the CellRendererPane. The font, foreground, and background properties are only set if their current value is either null or a UIResource, other properties are set if the current value is null.

**See Also:**[uninstallDefaults()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallDefaults()), [installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.swing.JComponent)), [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html)

### uninstallDefaults

protected void **uninstallDefaults**()

Set the JList properties that haven't been explicitly overridden to null. A property is considered overridden if its current value is not a UIResource.

**See Also:**[installDefaults()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installDefaults()), [uninstallUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallUI(javax.swing.JComponent)), [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html)

### installUI

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

Initializes this.list by calling installDefaults(), installListeners(), and installKeyboardActions() in order.

**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:**[installDefaults()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installDefaults()), [installListeners()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installListeners()), [installKeyboardActions()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installKeyboardActions())

### uninstallUI

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

Uninitializes this.list by calling uninstallListeners(), uninstallKeyboardActions(), and uninstallDefaults() in order. Sets this.list to null.

**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:**[uninstallListeners()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallListeners()), [uninstallKeyboardActions()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallKeyboardActions()), [uninstallDefaults()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#uninstallDefaults())

### createUI

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

Returns a new instance of BasicListUI. BasicListUI delegates are allocated one per JList.

**Returns:**A new ListUI implementation for the Windows look and feel.

### locationToIndex

public int **locationToIndex**([JList](http://docs.google.com/javax/swing/JList.html) list,  
 [Point](http://docs.google.com/java/awt/Point.html) location)

Returns the cell index in the specified JList closest to the given location in the list's coordinate system. To determine if the cell actually contains the specified location, compare the point against the cell's bounds, as provided by getCellBounds. This method returns -1 if the list's model is empty.

**Specified by:**[locationToIndex](http://docs.google.com/javax/swing/plaf/ListUI.html#locationToIndex(javax.swing.JList,%20java.awt.Point)) in class [ListUI](http://docs.google.com/javax/swing/plaf/ListUI.html) **Parameters:**list - the listlocation - the coordinates of the point **Returns:**the cell index closest to the given location, or -1 **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if location is null

### indexToLocation

public [Point](http://docs.google.com/java/awt/Point.html) **indexToLocation**([JList](http://docs.google.com/javax/swing/JList.html) list,  
 int index)

Returns the origin in the given JList, of the specified item, in the list's coordinate system. Returns null if the index isn't valid.

**Specified by:**[indexToLocation](http://docs.google.com/javax/swing/plaf/ListUI.html#indexToLocation(javax.swing.JList,%20int)) in class [ListUI](http://docs.google.com/javax/swing/plaf/ListUI.html) **Parameters:**list - the listindex - the cell index **Returns:**the origin of the cell, or null

### getCellBounds

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getCellBounds**([JList](http://docs.google.com/javax/swing/JList.html) list,  
 int index1,  
 int index2)

Returns the bounding rectangle, in the given list's coordinate system, for the range of cells specified by the two indices. The indices can be supplied in any order.

If the smaller index is outside the list's range of cells, this method returns null. If the smaller index is valid, but the larger index is outside the list's range, the bounds of just the first index is returned. Otherwise, the bounds of the valid range is returned.

**Specified by:**[getCellBounds](http://docs.google.com/javax/swing/plaf/ListUI.html#getCellBounds(javax.swing.JList,%20int,%20int)) in class [ListUI](http://docs.google.com/javax/swing/plaf/ListUI.html) **Parameters:**list - the listindex1 - the first index in the rangeindex2 - the second index in the range **Returns:**the bounding rectangle for the range of cells, or null

### getRowHeight

protected int **getRowHeight**(int row)

Returns the height of the specified row based on the current layout.

**Returns:**The specified row height or -1 if row isn't valid.**See Also:**[convertYToRow(int)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#convertYToRow(int)), [convertRowToY(int)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#convertRowToY(int)), [updateLayoutState()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#updateLayoutState())

### convertYToRow

protected int **convertYToRow**(int y0)

Convert the JList relative coordinate to the row that contains it, based on the current layout. If y0 doesn't fall within any row, return -1.

**Returns:**The row that contains y0, or -1.**See Also:**[getRowHeight(int)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#getRowHeight(int)), [updateLayoutState()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#updateLayoutState())

### convertRowToY

protected int **convertRowToY**(int row)

Return the JList relative Y coordinate of the origin of the specified row or -1 if row isn't valid.

**Returns:**The Y coordinate of the origin of row, or -1.**See Also:**[getRowHeight(int)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#getRowHeight(int)), [updateLayoutState()](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#updateLayoutState())

### maybeUpdateLayoutState

protected void **maybeUpdateLayoutState**()

If updateLayoutStateNeeded is non zero, call updateLayoutState() and reset updateLayoutStateNeeded. This method should be called by methods before doing any computation based on the geometry of the list. For example it's the first call in paint() and getPreferredSize().

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

### updateLayoutState

protected void **updateLayoutState**()

Recompute the value of cellHeight or cellHeights based and cellWidth, based on the current font and the current values of fixedCellWidth, fixedCellHeight, and prototypeCellValue.

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

### createMouseInputListener

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

Creates a delegate that implements MouseInputListener. The delegate is added to the corresponding java.awt.Component listener lists at installUI() time. Subclasses can override this method to return a custom MouseInputListener, e.g.

class MyListUI extends BasicListUI {  
 protected MouseInputListener **createMouseInputListener**() {  
 return new MyMouseInputHandler();  
 }  
 public class MyMouseInputHandler extends MouseInputHandler {  
 public void mouseMoved(MouseEvent e) {  
 // do some extra work when the mouse moves  
 super.mouseMoved(e);  
 }  
 }  
 }

**See Also:**[BasicListUI.MouseInputHandler](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.MouseInputHandler.html), [installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.swing.JComponent))

### createFocusListener

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

### createListSelectionListener

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

Creates an instance of ListSelectionHandler that's added to the JLists by selectionModel as needed. Subclasses can override this method to return a custom ListSelectionListener, e.g.

class MyListUI extends BasicListUI {  
 protected ListSelectionListener **createListSelectionListener**() {  
 return new MySelectionListener();  
 }  
 public class MySelectionListener extends ListSelectionHandler {  
 public void valueChanged(ListSelectionEvent e) {  
 // do some extra work when the selection changes  
 super.valueChange(e);  
 }  
 }  
 }

**See Also:**[BasicListUI.ListSelectionHandler](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.ListSelectionHandler.html), [installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.swing.JComponent))

### createListDataListener

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

Creates an instance of ListDataListener that's added to the JLists by model as needed. Subclasses can override this method to return a custom ListDataListener, e.g.

class MyListUI extends BasicListUI {  
 protected ListDataListener **createListDataListener**() {  
 return new MyListDataListener();  
 }  
 public class MyListDataListener extends ListDataHandler {  
 public void contentsChanged(ListDataEvent e) {  
 // do some extra work when the models contents change  
 super.contentsChange(e);  
 }  
 }  
 }

**See Also:**[ListDataListener](http://docs.google.com/javax/swing/event/ListDataListener.html), [JList.getModel()](http://docs.google.com/javax/swing/JList.html#getModel()), [installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.swing.JComponent))

### createPropertyChangeListener

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

Creates an instance of PropertyChangeHandler that's added to the JList by installUI(). Subclasses can override this method to return a custom PropertyChangeListener, e.g.

class MyListUI extends BasicListUI {  
 protected PropertyChangeListener **createPropertyChangeListener**() {  
 return new MyPropertyChangeListener();  
 }  
 public class MyPropertyChangeListener extends PropertyChangeHandler {  
 public void propertyChange(PropertyChangeEvent e) {  
 if (e.getPropertyName().equals("model")) {  
 // do some extra work when the model changes  
 }  
 super.propertyChange(e);  
 }  
 }  
 }

**See Also:**[PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html), [installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.html#installUI(javax.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/BasicListUI.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/BasicLabelUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicListUI.FocusHandler.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicListUI.html)    [**NO FRAMES**](http://docs.google.com/BasicListUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#1pxezwc) | [METHOD](#2p2csry) |

[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).