| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ListCellRenderer.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/LayoutStyle.ComponentPlacement.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/ListModel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/ListCellRenderer.html)    [**NO FRAMES**](http://docs.google.com/ListCellRenderer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **javax.swing**

Interface ListCellRenderer

**All Known Implementing Classes:** [BasicComboBoxRenderer](http://docs.google.com/javax/swing/plaf/basic/BasicComboBoxRenderer.html), [BasicComboBoxRenderer.UIResource](http://docs.google.com/javax/swing/plaf/basic/BasicComboBoxRenderer.UIResource.html), [DefaultListCellRenderer](http://docs.google.com/javax/swing/DefaultListCellRenderer.html), [DefaultListCellRenderer.UIResource](http://docs.google.com/javax/swing/DefaultListCellRenderer.UIResource.html), [MetalFileChooserUI.FileRenderer](http://docs.google.com/javax/swing/plaf/metal/MetalFileChooserUI.FileRenderer.html), [MetalFileChooserUI.FilterComboBoxRenderer](http://docs.google.com/javax/swing/plaf/metal/MetalFileChooserUI.FilterComboBoxRenderer.html)

public interface **ListCellRenderer**

Identifies components that can be used as "rubber stamps" to paint the cells in a JList. For example, to use a JLabel as a ListCellRenderer, you would write something like this:

class MyCellRenderer extends JLabel implements ListCellRenderer {  
 public MyCellRenderer() {  
 setOpaque(true);  
 }  
  
 public Component getListCellRendererComponent(JList list,  
 Object value,  
 int index,  
 boolean isSelected,  
 boolean cellHasFocus) {  
  
 setText(value.toString());  
  
 Color background;  
 Color foreground;  
  
 // check if this cell represents the current DnD drop location  
 JList.DropLocation dropLocation = list.getDropLocation();  
 if (dropLocation != null  
 && !dropLocation.isInsert()  
 && dropLocation.getIndex() == index) {  
  
 background = Color.BLUE;  
 foreground = Color.WHITE;  
  
 // check if this cell is selected  
 } else if (isSelected) {  
 background = Color.RED;  
 foreground = Color.WHITE;  
  
 // unselected, and not the DnD drop location  
 } else {  
 background = Color.WHITE;  
 foreground = Color.BLACK;  
 };  
  
 setBackground(background);  
 setForeground(foreground);  
  
 return this;  
 }  
 }

**See Also:**[JList](http://docs.google.com/javax/swing/JList.html), [DefaultListCellRenderer](http://docs.google.com/javax/swing/DefaultListCellRenderer.html)

| **Method Summary** | |
| --- | --- |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getListCellRendererComponent**](http://docs.google.com/javax/swing/ListCellRenderer.html#getListCellRendererComponent(javax.swing.JList,%20java.lang.Object,%20int,%20boolean,%20boolean))([JList](http://docs.google.com/javax/swing/JList.html) list, [Object](http://docs.google.com/java/lang/Object.html) value, int index, boolean isSelected, boolean cellHasFocus)            Return a component that has been configured to display the specified value. |

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

### getListCellRendererComponent

[Component](http://docs.google.com/java/awt/Component.html) **getListCellRendererComponent**([JList](http://docs.google.com/javax/swing/JList.html) list,  
 [Object](http://docs.google.com/java/lang/Object.html) value,  
 int index,  
 boolean isSelected,  
 boolean cellHasFocus)

Return a component that has been configured to display the specified value. That component's paint method is then called to "render" the cell. If it is necessary to compute the dimensions of a list because the list cells do not have a fixed size, this method is called to generate a component on which getPreferredSize can be invoked.

**Parameters:**list - The JList we're painting.value - The value returned by list.getModel().getElementAt(index).index - The cells index.isSelected - True if the specified cell was selected.cellHasFocus - True if the specified cell has the focus. **Returns:**A component whose paint() method will render the specified value.**See Also:**[JList](http://docs.google.com/javax/swing/JList.html), [ListSelectionModel](http://docs.google.com/javax/swing/ListSelectionModel.html), [ListModel](http://docs.google.com/javax/swing/ListModel.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ListCellRenderer.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/LayoutStyle.ComponentPlacement.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/ListModel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/ListCellRenderer.html)    [**NO FRAMES**](http://docs.google.com/ListCellRenderer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).