| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JProgressBar.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/JPopupMenu.Separator.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JProgressBar.AccessibleJProgressBar.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JProgressBar.html)    [**NO FRAMES**](http://docs.google.com/JProgressBar.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#qsh70q) | [METHOD](#3o7alnk) |

## **javax.swing**

Class JProgressBar

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.Component](http://docs.google.com/java/awt/Component.html)  
 [java.awt.Container](http://docs.google.com/java/awt/Container.html)  
 [javax.swing.JComponent](http://docs.google.com/javax/swing/JComponent.html)  
 **javax.swing.JProgressBar**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html), [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

public class **JProgressBar**extends [JComponent](http://docs.google.com/javax/swing/JComponent.html)implements [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

A component that visually displays the progress of some task. As the task progresses towards completion, the progress bar displays the task's percentage of completion. This percentage is typically represented visually by a rectangle which starts out empty and gradually becomes filled in as the task progresses. In addition, the progress bar can display a textual representation of this percentage.

JProgressBar uses a BoundedRangeModel as its data model, with the value property representing the "current" state of the task, and the minimum and maximum properties representing the beginning and end points, respectively.

To indicate that a task of unknown length is executing, you can put a progress bar into indeterminate mode. While the bar is in indeterminate mode, it animates constantly to show that work is occurring. As soon as you can determine the task's length and amount of progress, you should update the progress bar's value and switch it back to determinate mode.

Here is an example of creating a progress bar, where task is an object (representing some piece of work) which returns information about the progress of the task:

progressBar = new JProgressBar(0, task.getLengthOfTask());  
progressBar.setValue(0);  
progressBar.setStringPainted(true);

Here is an example of querying the current state of the task, and using the returned value to update the progress bar:

progressBar.setValue(task.getCurrent());

Here is an example of putting a progress bar into indeterminate mode, and then switching back to determinate mode once the length of the task is known:

progressBar = new JProgressBar();  
*...//when the task of (initially) unknown length begins:*  
progressBar.setIndeterminate(true);  
*...//do some work; get length of task...*  
progressBar.setMaximum(newLength);  
progressBar.setValue(newValue);  
progressBar.setIndeterminate(false);

For complete examples and further documentation see [How to Monitor Progress](http://java.sun.com/docs/books/tutorial/uiswing/components/progress.html), a section in *The Java Tutorial.*

**Warning:** Swing is not thread safe. For more information see [Swing's Threading Policy](http://docs.google.com/package-summary.html#threading).

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

**See Also:**[BasicProgressBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicProgressBarUI.html), [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html), [SwingWorker](http://docs.google.com/javax/swing/SwingWorker.html)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JProgressBar.AccessibleJProgressBar**](http://docs.google.com/javax/swing/JProgressBar.AccessibleJProgressBar.html)            This class implements accessibility support for the JProgressBar class. |

| **Nested classes/interfaces inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [JComponent.AccessibleJComponent](http://docs.google.com/javax/swing/JComponent.AccessibleJComponent.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [Container.AccessibleAWTContainer](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [Component.AccessibleAWTComponent](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html), [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html), [Component.BltBufferStrategy](http://docs.google.com/java/awt/Component.BltBufferStrategy.html), [Component.FlipBufferStrategy](http://docs.google.com/java/awt/Component.FlipBufferStrategy.html) |

| **Field Summary** | |
| --- | --- |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | [**changeEvent**](http://docs.google.com/javax/swing/JProgressBar.html#changeEvent)            Only one ChangeEvent is needed per instance since the event's only interesting property is the immutable source, which is the progress bar. |
| protected  [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) | [**changeListener**](http://docs.google.com/javax/swing/JProgressBar.html#changeListener)            Listens for change events sent by the progress bar's model, redispatching them to change-event listeners registered upon this progress bar. |
| protected  [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) | [**model**](http://docs.google.com/javax/swing/JProgressBar.html#model)            The object that holds the data for the progress bar. |
| protected  int | [**orientation**](http://docs.google.com/javax/swing/JProgressBar.html#orientation)            Whether the progress bar is horizontal or vertical. |
| protected  boolean | [**paintBorder**](http://docs.google.com/javax/swing/JProgressBar.html#paintBorder)            Whether to display a border around the progress bar. |
| protected  boolean | [**paintString**](http://docs.google.com/javax/swing/JProgressBar.html#paintString)            Whether to display a string of text on the progress bar. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**progressString**](http://docs.google.com/javax/swing/JProgressBar.html#progressString)            An optional string that can be displayed on the progress bar. |

| **Fields inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [accessibleContext](http://docs.google.com/javax/swing/JComponent.html#accessibleContext), [listenerList](http://docs.google.com/javax/swing/JComponent.html#listenerList), [TOOL\_TIP\_TEXT\_KEY](http://docs.google.com/javax/swing/JComponent.html#TOOL_TIP_TEXT_KEY), [ui](http://docs.google.com/javax/swing/JComponent.html#ui), [UNDEFINED\_CONDITION](http://docs.google.com/javax/swing/JComponent.html#UNDEFINED_CONDITION), [WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT](http://docs.google.com/javax/swing/JComponent.html#WHEN_ANCESTOR_OF_FOCUSED_COMPONENT), [WHEN\_FOCUSED](http://docs.google.com/javax/swing/JComponent.html#WHEN_FOCUSED), [WHEN\_IN\_FOCUSED\_WINDOW](http://docs.google.com/javax/swing/JComponent.html#WHEN_IN_FOCUSED_WINDOW) |

| **Fields inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [BOTTOM\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#BOTTOM_ALIGNMENT), [CENTER\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#CENTER_ALIGNMENT), [LEFT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#LEFT_ALIGNMENT), [RIGHT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#RIGHT_ALIGNMENT), [TOP\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#TOP_ALIGNMENT) |

| **Fields inherited from interface javax.swing.**[**SwingConstants**](http://docs.google.com/javax/swing/SwingConstants.html) |
| --- |
| [BOTTOM](http://docs.google.com/javax/swing/SwingConstants.html#BOTTOM), [CENTER](http://docs.google.com/javax/swing/SwingConstants.html#CENTER), [EAST](http://docs.google.com/javax/swing/SwingConstants.html#EAST), [HORIZONTAL](http://docs.google.com/javax/swing/SwingConstants.html#HORIZONTAL), [LEADING](http://docs.google.com/javax/swing/SwingConstants.html#LEADING), [LEFT](http://docs.google.com/javax/swing/SwingConstants.html#LEFT), [NEXT](http://docs.google.com/javax/swing/SwingConstants.html#NEXT), [NORTH](http://docs.google.com/javax/swing/SwingConstants.html#NORTH), [NORTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_EAST), [NORTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_WEST), [PREVIOUS](http://docs.google.com/javax/swing/SwingConstants.html#PREVIOUS), [RIGHT](http://docs.google.com/javax/swing/SwingConstants.html#RIGHT), [SOUTH](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH), [SOUTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_EAST), [SOUTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_WEST), [TOP](http://docs.google.com/javax/swing/SwingConstants.html#TOP), [TRAILING](http://docs.google.com/javax/swing/SwingConstants.html#TRAILING), [VERTICAL](http://docs.google.com/javax/swing/SwingConstants.html#VERTICAL), [WEST](http://docs.google.com/javax/swing/SwingConstants.html#WEST) |

| **Fields inherited from interface java.awt.image.**[**ImageObserver**](http://docs.google.com/java/awt/image/ImageObserver.html) |
| --- |
| [ABORT](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT), [ALLBITS](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS), [ERROR](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR), [FRAMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS), [HEIGHT](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT), [PROPERTIES](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES), [SOMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS), [WIDTH](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH) |

| **Constructor Summary** | |
| --- | --- |
| [**JProgressBar**](http://docs.google.com/javax/swing/JProgressBar.html#JProgressBar())()            Creates a horizontal progress bar that displays a border but no progress string. |
| [**JProgressBar**](http://docs.google.com/javax/swing/JProgressBar.html#JProgressBar(javax.swing.BoundedRangeModel))([BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) newModel)            Creates a horizontal progress bar that uses the specified model to hold the progress bar's data. |
| [**JProgressBar**](http://docs.google.com/javax/swing/JProgressBar.html#JProgressBar(int))(int orient)            Creates a progress bar with the specified orientation, which can be either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL. |
| [**JProgressBar**](http://docs.google.com/javax/swing/JProgressBar.html#JProgressBar(int,%20int))(int min, int max)            Creates a horizontal progress bar with the specified minimum and maximum. |
| [**JProgressBar**](http://docs.google.com/javax/swing/JProgressBar.html#JProgressBar(int,%20int,%20int))(int orient, int min, int max)            Creates a progress bar using the specified orientation, minimum, and maximum. |

| **Method Summary** | |
| --- | --- |
| void | [**addChangeListener**](http://docs.google.com/javax/swing/JProgressBar.html#addChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Adds the specified ChangeListener to the progress bar. |
| protected  [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) | [**createChangeListener**](http://docs.google.com/javax/swing/JProgressBar.html#createChangeListener())()            Subclasses that want to handle change events from the model differently can override this to return an instance of a custom ChangeListener implementation. |
| protected  void | [**fireStateChanged**](http://docs.google.com/javax/swing/JProgressBar.html#fireStateChanged())()            Send a ChangeEvent, whose source is this JProgressBar, to all ChangeListeners that have registered interest in ChangeEvents. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JProgressBar.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JProgressBar. |
| [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html)[] | [**getChangeListeners**](http://docs.google.com/javax/swing/JProgressBar.html#getChangeListeners())()            Returns an array of all the ChangeListeners added to this progress bar with addChangeListener. |
| int | [**getMaximum**](http://docs.google.com/javax/swing/JProgressBar.html#getMaximum())()            Returns the progress bar's maximum value from the BoundedRangeModel. |
| int | [**getMinimum**](http://docs.google.com/javax/swing/JProgressBar.html#getMinimum())()            Returns the progress bar's minimum value from the BoundedRangeModel. |
| [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) | [**getModel**](http://docs.google.com/javax/swing/JProgressBar.html#getModel())()            Returns the data model used by this progress bar. |
| int | [**getOrientation**](http://docs.google.com/javax/swing/JProgressBar.html#getOrientation())()            Returns SwingConstants.VERTICAL or SwingConstants.HORIZONTAL, depending on the orientation of the progress bar. |
| double | [**getPercentComplete**](http://docs.google.com/javax/swing/JProgressBar.html#getPercentComplete())()            Returns the percent complete for the progress bar. |
| [String](http://docs.google.com/java/lang/String.html) | [**getString**](http://docs.google.com/javax/swing/JProgressBar.html#getString())()            Returns a String representation of the current progress. |
| [ProgressBarUI](http://docs.google.com/javax/swing/plaf/ProgressBarUI.html) | [**getUI**](http://docs.google.com/javax/swing/JProgressBar.html#getUI())()            Returns the look-and-feel object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JProgressBar.html#getUIClassID())()            Returns the name of the look-and-feel class that renders this component. |
| int | [**getValue**](http://docs.google.com/javax/swing/JProgressBar.html#getValue())()            Returns the progress bar's current value from the BoundedRangeModel. |
| boolean | [**isBorderPainted**](http://docs.google.com/javax/swing/JProgressBar.html#isBorderPainted())()            Returns the borderPainted property. |
| boolean | [**isIndeterminate**](http://docs.google.com/javax/swing/JProgressBar.html#isIndeterminate())()            Returns the value of the indeterminate property. |
| boolean | [**isStringPainted**](http://docs.google.com/javax/swing/JProgressBar.html#isStringPainted())()            Returns the value of the stringPainted property. |
| protected  void | [**paintBorder**](http://docs.google.com/javax/swing/JProgressBar.html#paintBorder(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Paints the progress bar's border if the borderPainted property is true. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JProgressBar.html#paramString())()            Returns a string representation of this JProgressBar. |
| void | [**removeChangeListener**](http://docs.google.com/javax/swing/JProgressBar.html#removeChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Removes a ChangeListener from the progress bar. |
| void | [**setBorderPainted**](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean))(boolean b)            Sets the borderPainted property, which is true if the progress bar should paint its border. |
| void | [**setIndeterminate**](http://docs.google.com/javax/swing/JProgressBar.html#setIndeterminate(boolean))(boolean newValue)            Sets the indeterminate property of the progress bar, which determines whether the progress bar is in determinate or indeterminate mode. |
| void | [**setMaximum**](http://docs.google.com/javax/swing/JProgressBar.html#setMaximum(int))(int n)            Sets the progress bar's maximum value (stored in the progress bar's data model) to n. |
| void | [**setMinimum**](http://docs.google.com/javax/swing/JProgressBar.html#setMinimum(int))(int n)            Sets the progress bar's minimum value (stored in the progress bar's data model) to n. |
| void | [**setModel**](http://docs.google.com/javax/swing/JProgressBar.html#setModel(javax.swing.BoundedRangeModel))([BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) newModel)            Sets the data model used by the JProgressBar. |
| void | [**setOrientation**](http://docs.google.com/javax/swing/JProgressBar.html#setOrientation(int))(int newOrientation)            Sets the progress bar's orientation to newOrientation, which must be SwingConstants.VERTICAL or SwingConstants.HORIZONTAL. |
| void | [**setString**](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Sets the value of the progress string. |
| void | [**setStringPainted**](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean))(boolean b)            Sets the value of the stringPainted property, which determines whether the progress bar should render a progress string. |
| void | [**setUI**](http://docs.google.com/javax/swing/JProgressBar.html#setUI(javax.swing.plaf.ProgressBarUI))([ProgressBarUI](http://docs.google.com/javax/swing/plaf/ProgressBarUI.html) ui)            Sets the look-and-feel object that renders this component. |
| void | [**setValue**](http://docs.google.com/javax/swing/JProgressBar.html#setValue(int))(int n)            Sets the progress bar's current value to n. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JProgressBar.html#updateUI())()            Resets the UI property to a value from the current look and feel. |

| **Methods inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [addAncestorListener](http://docs.google.com/javax/swing/JComponent.html#addAncestorListener(javax.swing.event.AncestorListener)), [addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()), [addVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#addVetoableChangeListener(java.beans.VetoableChangeListener)), [computeVisibleRect](http://docs.google.com/javax/swing/JComponent.html#computeVisibleRect(java.awt.Rectangle)), [contains](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [createToolTip](http://docs.google.com/javax/swing/JComponent.html#createToolTip()), [disable](http://docs.google.com/javax/swing/JComponent.html#disable()), [enable](http://docs.google.com/javax/swing/JComponent.html#enable()), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20int,%20int)), [fireVetoableChange](http://docs.google.com/javax/swing/JComponent.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [getActionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getActionForKeyStroke(javax.swing.KeyStroke)), [getActionMap](http://docs.google.com/javax/swing/JComponent.html#getActionMap()), [getAlignmentX](http://docs.google.com/javax/swing/JComponent.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/javax/swing/JComponent.html#getAlignmentY()), [getAncestorListeners](http://docs.google.com/javax/swing/JComponent.html#getAncestorListeners()), [getAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#getAutoscrolls()), [getBaseline](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/JComponent.html#getBaselineResizeBehavior()), [getBorder](http://docs.google.com/javax/swing/JComponent.html#getBorder()), [getBounds](http://docs.google.com/javax/swing/JComponent.html#getBounds(java.awt.Rectangle)), [getClientProperty](http://docs.google.com/javax/swing/JComponent.html#getClientProperty(java.lang.Object)), [getComponentGraphics](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [getComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getComponentPopupMenu()), [getConditionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getConditionForKeyStroke(javax.swing.KeyStroke)), [getDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#getDebugGraphicsOptions()), [getDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale()), [getFontMetrics](http://docs.google.com/javax/swing/JComponent.html#getFontMetrics(java.awt.Font)), [getGraphics](http://docs.google.com/javax/swing/JComponent.html#getGraphics()), [getHeight](http://docs.google.com/javax/swing/JComponent.html#getHeight()), [getInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getInheritsPopupMenu()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [getInputVerifier](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)), [getListeners](http://docs.google.com/javax/swing/JComponent.html#getListeners(java.lang.Class)), [getLocation](http://docs.google.com/javax/swing/JComponent.html#getLocation(java.awt.Point)), [getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [getNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent()), [getPopupLocation](http://docs.google.com/javax/swing/JComponent.html#getPopupLocation(java.awt.event.MouseEvent)), [getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [getRegisteredKeyStrokes](http://docs.google.com/javax/swing/JComponent.html#getRegisteredKeyStrokes()), [getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()), [getSize](http://docs.google.com/javax/swing/JComponent.html#getSize(java.awt.Dimension)), [getToolTipLocation](http://docs.google.com/javax/swing/JComponent.html#getToolTipLocation(java.awt.event.MouseEvent)), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText()), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText(java.awt.event.MouseEvent)), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()), [paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [paintChildren](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(int,%20int,%20int,%20int)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(java.awt.Rectangle)), [print](http://docs.google.com/javax/swing/JComponent.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/javax/swing/JComponent.html#printAll(java.awt.Graphics)), [printBorder](http://docs.google.com/javax/swing/JComponent.html#printBorder(java.awt.Graphics)), [printChildren](http://docs.google.com/javax/swing/JComponent.html#printChildren(java.awt.Graphics)), [printComponent](http://docs.google.com/javax/swing/JComponent.html#printComponent(java.awt.Graphics)), [processComponentKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processComponentKeyEvent(java.awt.event.KeyEvent)), [processKeyBinding](http://docs.google.com/javax/swing/JComponent.html#processKeyBinding(javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20int,%20boolean)), [processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [putClientProperty](http://docs.google.com/javax/swing/JComponent.html#putClientProperty(java.lang.Object,%20java.lang.Object)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20javax.swing.KeyStroke,%20int)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int)), [removeAncestorListener](http://docs.google.com/javax/swing/JComponent.html#removeAncestorListener(javax.swing.event.AncestorListener)), [removeNotify](http://docs.google.com/javax/swing/JComponent.html#removeNotify()), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

| **Methods inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [addContainerListener](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int)), [countComponents](http://docs.google.com/java/awt/Container.html#countComponents()), [deliverEvent](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event)), [doLayout](http://docs.google.com/java/awt/Container.html#doLayout()), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(int,%20int)), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(java.awt.Point)), [getComponent](http://docs.google.com/java/awt/Container.html#getComponent(int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(java.awt.Point)), [getComponentCount](http://docs.google.com/java/awt/Container.html#getComponentCount()), [getComponents](http://docs.google.com/java/awt/Container.html#getComponents()), [getComponentZOrder](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component)), [getContainerListeners](http://docs.google.com/java/awt/Container.html#getContainerListeners()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int)), [getFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [insets](http://docs.google.com/java/awt/Container.html#insets()), [invalidate](http://docs.google.com/java/awt/Container.html#invalidate()), [isAncestorOf](http://docs.google.com/java/awt/Container.html#isAncestorOf(java.awt.Component)), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [isFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider()), [isFocusTraversalPolicySet](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicySet()), [layout](http://docs.google.com/java/awt/Container.html#layout()), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Container.html#locate(int,%20int)), [minimumSize](http://docs.google.com/java/awt/Container.html#minimumSize()), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [printComponents](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics)), [processContainerEvent](http://docs.google.com/java/awt/Container.html#processContainerEvent(java.awt.event.ContainerEvent)), [processEvent](http://docs.google.com/java/awt/Container.html#processEvent(java.awt.AWTEvent)), [remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [remove](http://docs.google.com/java/awt/Container.html#remove(int)), [removeAll](http://docs.google.com/java/awt/Container.html#removeAll()), [removeContainerListener](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [setComponentZOrder](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int)), [setFocusCycleRoot](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [validate](http://docs.google.com/java/awt/Container.html#validate()), [validateTree](http://docs.google.com/java/awt/Container.html#validateTree()) |

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processInputMethodEvent](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)), [remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)), [removeComponentListener](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [removeFocusListener](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [removeHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyListener](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [removeInputMethodListener](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [removeKeyListener](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [removeMouseListener](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [removeMouseMotionListener](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseWheelListener](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [repaint](http://docs.google.com/java/awt/Component.html#repaint()), [repaint](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long)), [resize](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension)), [resize](http://docs.google.com/java/awt/Component.html#resize(int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setIgnoreRepaint](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean)), [setLocale](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [show](http://docs.google.com/java/awt/Component.html#show()), [show](http://docs.google.com/java/awt/Component.html#show(boolean)), [size](http://docs.google.com/java/awt/Component.html#size()), [toString](http://docs.google.com/java/awt/Component.html#toString()), [transferFocus](http://docs.google.com/java/awt/Component.html#transferFocus()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()) |

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

### orientation

protected int **orientation**

Whether the progress bar is horizontal or vertical. The default is HORIZONTAL.

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

### paintBorder

protected boolean **paintBorder**

Whether to display a border around the progress bar. The default is true.

**See Also:**[setBorderPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean))

### model

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

The object that holds the data for the progress bar.

**See Also:**[setModel(javax.swing.BoundedRangeModel)](http://docs.google.com/javax/swing/JProgressBar.html#setModel(javax.swing.BoundedRangeModel))

### progressString

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

An optional string that can be displayed on the progress bar. The default is null. Setting this to a non-null value does not imply that the string will be displayed. To display the string, paintString must be true.

**See Also:**[setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String)), [setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean))

### paintString

protected boolean **paintString**

Whether to display a string of text on the progress bar. The default is false. Setting this to true causes a textual display of the progress to be rendered on the progress bar. If the progressString is null, the percentage of completion is displayed on the progress bar. Otherwise, the progressString is rendered on the progress bar.

**See Also:**[setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean)), [setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String))

### changeEvent

protected transient [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) **changeEvent**

Only one ChangeEvent is needed per instance since the event's only interesting property is the immutable source, which is the progress bar. The event is lazily created the first time that an event notification is fired.

**See Also:**[fireStateChanged()](http://docs.google.com/javax/swing/JProgressBar.html#fireStateChanged())

### changeListener

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

Listens for change events sent by the progress bar's model, redispatching them to change-event listeners registered upon this progress bar.

**See Also:**[createChangeListener()](http://docs.google.com/javax/swing/JProgressBar.html#createChangeListener())

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

### JProgressBar

public **JProgressBar**()

Creates a horizontal progress bar that displays a border but no progress string. The initial and minimum values are 0, and the maximum is 100.

**See Also:**[setOrientation(int)](http://docs.google.com/javax/swing/JProgressBar.html#setOrientation(int)), [setBorderPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean)), [setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean)), [setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String)), [setIndeterminate(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setIndeterminate(boolean))

### JProgressBar

public **JProgressBar**(int orient)

Creates a progress bar with the specified orientation, which can be either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL. By default, a border is painted but a progress string is not. The initial and minimum values are 0, and the maximum is 100.

**Parameters:**orient - the desired orientation of the progress bar **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if orient is an illegal value**See Also:**[setOrientation(int)](http://docs.google.com/javax/swing/JProgressBar.html#setOrientation(int)), [setBorderPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean)), [setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean)), [setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String)), [setIndeterminate(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setIndeterminate(boolean))

### JProgressBar

public **JProgressBar**(int min,  
 int max)

Creates a horizontal progress bar with the specified minimum and maximum. Sets the initial value of the progress bar to the specified minimum. By default, a border is painted but a progress string is not.

The BoundedRangeModel that holds the progress bar's data handles any issues that may arise from improperly setting the minimum, initial, and maximum values on the progress bar. See the BoundedRangeModel documentation for details.

**Parameters:**min - the minimum value of the progress barmax - the maximum value of the progress bar**See Also:**[BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html), [setOrientation(int)](http://docs.google.com/javax/swing/JProgressBar.html#setOrientation(int)), [setBorderPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean)), [setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean)), [setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String)), [setIndeterminate(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setIndeterminate(boolean))

### JProgressBar

public **JProgressBar**(int orient,  
 int min,  
 int max)

Creates a progress bar using the specified orientation, minimum, and maximum. By default, a border is painted but a progress string is not. Sets the initial value of the progress bar to the specified minimum.

The BoundedRangeModel that holds the progress bar's data handles any issues that may arise from improperly setting the minimum, initial, and maximum values on the progress bar. See the BoundedRangeModel documentation for details.

**Parameters:**orient - the desired orientation of the progress barmin - the minimum value of the progress barmax - the maximum value of the progress bar **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if orient is an illegal value**See Also:**[BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html), [setOrientation(int)](http://docs.google.com/javax/swing/JProgressBar.html#setOrientation(int)), [setBorderPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean)), [setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean)), [setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String)), [setIndeterminate(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setIndeterminate(boolean))

### JProgressBar

public **JProgressBar**([BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) newModel)

Creates a horizontal progress bar that uses the specified model to hold the progress bar's data. By default, a border is painted but a progress string is not.

**Parameters:**newModel - the data model for the progress bar**See Also:**[setOrientation(int)](http://docs.google.com/javax/swing/JProgressBar.html#setOrientation(int)), [setBorderPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean)), [setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean)), [setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String)), [setIndeterminate(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setIndeterminate(boolean))

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

### getOrientation

public int **getOrientation**()

Returns SwingConstants.VERTICAL or SwingConstants.HORIZONTAL, depending on the orientation of the progress bar. The default orientation is SwingConstants.HORIZONTAL.

**Returns:**HORIZONTAL or VERTICAL**See Also:**[setOrientation(int)](http://docs.google.com/javax/swing/JProgressBar.html#setOrientation(int))

### setOrientation

public void **setOrientation**(int newOrientation)

Sets the progress bar's orientation to newOrientation, which must be SwingConstants.VERTICAL or SwingConstants.HORIZONTAL. The default orientation is SwingConstants.HORIZONTAL.

**Parameters:**newOrientation - HORIZONTAL or VERTICAL **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if newOrientation is an illegal value**See Also:**[getOrientation()](http://docs.google.com/javax/swing/JProgressBar.html#getOrientation())

### isStringPainted

public boolean **isStringPainted**()

Returns the value of the stringPainted property.

**Returns:**the value of the stringPainted property**See Also:**[setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean)), [setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String))

### setStringPainted

public void **setStringPainted**(boolean b)

Sets the value of the stringPainted property, which determines whether the progress bar should render a progress string. The default is false, meaning no string is painted. Some look and feels might not support progress strings or might support them only when the progress bar is in determinate mode.

**Parameters:**b - true if the progress bar should render a string**See Also:**[isStringPainted()](http://docs.google.com/javax/swing/JProgressBar.html#isStringPainted()), [setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String))

### getString

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

Returns a String representation of the current progress. By default, this returns a simple percentage String based on the value returned from getPercentComplete. An example would be the "42%". You can change this by calling setString.

**Returns:**the value of the progress string, or a simple percentage string if the progress string is null**See Also:**[setString(java.lang.String)](http://docs.google.com/javax/swing/JProgressBar.html#setString(java.lang.String))

### setString

public void **setString**([String](http://docs.google.com/java/lang/String.html) s)

Sets the value of the progress string. By default, this string is null, implying the built-in behavior of using a simple percent string. If you have provided a custom progress string and want to revert to the built-in behavior, set the string back to null.

The progress string is painted only if the isStringPainted method returns true.

**Parameters:**s - the value of the progress string**See Also:**[getString()](http://docs.google.com/javax/swing/JProgressBar.html#getString()), [setStringPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setStringPainted(boolean)), [isStringPainted()](http://docs.google.com/javax/swing/JProgressBar.html#isStringPainted())

### getPercentComplete

public double **getPercentComplete**()

Returns the percent complete for the progress bar. Note that this number is between 0.0 and 1.0.

**Returns:**the percent complete for this progress bar

### isBorderPainted

public boolean **isBorderPainted**()

Returns the borderPainted property.

**Returns:**the value of the borderPainted property**See Also:**[setBorderPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean))

### setBorderPainted

public void **setBorderPainted**(boolean b)

Sets the borderPainted property, which is true if the progress bar should paint its border. The default value for this property is true. Some look and feels might not implement painted borders; they will ignore this property.

**Parameters:**b - true if the progress bar should paint its border; otherwise, false**See Also:**[isBorderPainted()](http://docs.google.com/javax/swing/JProgressBar.html#isBorderPainted())

### paintBorder

protected void **paintBorder**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Paints the progress bar's border if the borderPainted property is true.

**Overrides:**[paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**g - the Graphics context within which to paint the border**See Also:**[JComponent.paint(java.awt.Graphics)](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [JComponent.setBorder(javax.swing.border.Border)](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [isBorderPainted()](http://docs.google.com/javax/swing/JProgressBar.html#isBorderPainted()), [setBorderPainted(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setBorderPainted(boolean))

### getUI

public [ProgressBarUI](http://docs.google.com/javax/swing/plaf/ProgressBarUI.html) **getUI**()

Returns the look-and-feel object that renders this component.

**Returns:**the ProgressBarUI object that renders this component

### setUI

public void **setUI**([ProgressBarUI](http://docs.google.com/javax/swing/plaf/ProgressBarUI.html) ui)

Sets the look-and-feel object that renders this component.

**Parameters:**ui - a ProgressBarUI object**See Also:**[UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### updateUI

public void **updateUI**()

Resets the UI property to a value from the current look and feel.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/JComponent.html#updateUI()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **See Also:**[JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### getUIClassID

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

Returns the name of the look-and-feel class that renders this component.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the string "ProgressBarUI"**See Also:**[JComponent.getUIClassID()](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()), [UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### createChangeListener

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

Subclasses that want to handle change events from the model differently can override this to return an instance of a custom ChangeListener implementation. The default ChangeListener simply calls the fireStateChanged method to forward ChangeEvents to the ChangeListeners that have been added directly to the progress bar.

**See Also:**[changeListener](http://docs.google.com/javax/swing/JProgressBar.html#changeListener), [fireStateChanged()](http://docs.google.com/javax/swing/JProgressBar.html#fireStateChanged()), [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html), [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html)

### addChangeListener

public void **addChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Adds the specified ChangeListener to the progress bar.

**Parameters:**l - the ChangeListener to add

### removeChangeListener

public void **removeChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Removes a ChangeListener from the progress bar.

**Parameters:**l - the ChangeListener to remove

### getChangeListeners

public [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html)[] **getChangeListeners**()

Returns an array of all the ChangeListeners added to this progress bar with addChangeListener.

**Returns:**all of the ChangeListeners added or an empty array if no listeners have been added**Since:** 1.4

### fireStateChanged

protected void **fireStateChanged**()

Send a ChangeEvent, whose source is this JProgressBar, to all ChangeListeners that have registered interest in ChangeEvents. This method is called each time a ChangeEvent is received from the model.

The event instance is created if necessary, and stored in changeEvent.

**See Also:**[addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JProgressBar.html#addChangeListener(javax.swing.event.ChangeListener)), [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### getModel

public [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) **getModel**()

Returns the data model used by this progress bar.

**Returns:**the BoundedRangeModel currently in use**See Also:**[setModel(javax.swing.BoundedRangeModel)](http://docs.google.com/javax/swing/JProgressBar.html#setModel(javax.swing.BoundedRangeModel)), [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html)

### setModel

public void **setModel**([BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) newModel)

Sets the data model used by the JProgressBar. Note that the BoundedRangeModel's extent is not used, and is set to 0.

**Parameters:**newModel - the BoundedRangeModel to use

### getValue

public int **getValue**()

Returns the progress bar's current value from the BoundedRangeModel. The value is always between the minimum and maximum values, inclusive.

**Returns:**the current value of the progress bar**See Also:**[setValue(int)](http://docs.google.com/javax/swing/JProgressBar.html#setValue(int)), [BoundedRangeModel.getValue()](http://docs.google.com/javax/swing/BoundedRangeModel.html#getValue())

### getMinimum

public int **getMinimum**()

Returns the progress bar's minimum value from the BoundedRangeModel.

**Returns:**the progress bar's minimum value**See Also:**[setMinimum(int)](http://docs.google.com/javax/swing/JProgressBar.html#setMinimum(int)), [BoundedRangeModel.getMinimum()](http://docs.google.com/javax/swing/BoundedRangeModel.html#getMinimum())

### getMaximum

public int **getMaximum**()

Returns the progress bar's maximum value from the BoundedRangeModel.

**Returns:**the progress bar's maximum value**See Also:**[setMaximum(int)](http://docs.google.com/javax/swing/JProgressBar.html#setMaximum(int)), [BoundedRangeModel.getMaximum()](http://docs.google.com/javax/swing/BoundedRangeModel.html#getMaximum())

### setValue

public void **setValue**(int n)

Sets the progress bar's current value to n. This method forwards the new value to the model.

The data model (an instance of BoundedRangeModel) handles any mathematical issues arising from assigning faulty values. See the BoundedRangeModel documentation for details.

If the new value is different from the previous value, all change listeners are notified.

**Parameters:**n - the new value**See Also:**[getValue()](http://docs.google.com/javax/swing/JProgressBar.html#getValue()), [addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JProgressBar.html#addChangeListener(javax.swing.event.ChangeListener)), [BoundedRangeModel.setValue(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setValue(int))

### setMinimum

public void **setMinimum**(int n)

Sets the progress bar's minimum value (stored in the progress bar's data model) to n.

The data model (a BoundedRangeModel instance) handles any mathematical issues arising from assigning faulty values. See the BoundedRangeModel documentation for details.

If the minimum value is different from the previous minimum, all change listeners are notified.

**Parameters:**n - the new minimum**See Also:**[getMinimum()](http://docs.google.com/javax/swing/JProgressBar.html#getMinimum()), [addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JProgressBar.html#addChangeListener(javax.swing.event.ChangeListener)), [BoundedRangeModel.setMinimum(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setMinimum(int))

### setMaximum

public void **setMaximum**(int n)

Sets the progress bar's maximum value (stored in the progress bar's data model) to n.

The underlying BoundedRangeModel handles any mathematical issues arising from assigning faulty values. See the BoundedRangeModel documentation for details.

If the maximum value is different from the previous maximum, all change listeners are notified.

**Parameters:**n - the new maximum**See Also:**[getMaximum()](http://docs.google.com/javax/swing/JProgressBar.html#getMaximum()), [addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JProgressBar.html#addChangeListener(javax.swing.event.ChangeListener)), [BoundedRangeModel.setMaximum(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setMaximum(int))

### setIndeterminate

public void **setIndeterminate**(boolean newValue)

Sets the indeterminate property of the progress bar, which determines whether the progress bar is in determinate or indeterminate mode. An indeterminate progress bar continuously displays animation indicating that an operation of unknown length is occurring. By default, this property is false. Some look and feels might not support indeterminate progress bars; they will ignore this property.

See [How to Monitor Progress](http://java.sun.com/docs/books/tutorial/uiswing/components/progress.html) for examples of using indeterminate progress bars.

**Parameters:**newValue - true if the progress bar should change to indeterminate mode; false if it should revert to normal.**Since:** 1.4 **See Also:**[isIndeterminate()](http://docs.google.com/javax/swing/JProgressBar.html#isIndeterminate()), [BasicProgressBarUI](http://docs.google.com/javax/swing/plaf/basic/BasicProgressBarUI.html)

### isIndeterminate

public boolean **isIndeterminate**()

Returns the value of the indeterminate property.

**Returns:**the value of the indeterminate property**Since:** 1.4 **See Also:**[setIndeterminate(boolean)](http://docs.google.com/javax/swing/JProgressBar.html#setIndeterminate(boolean))

### paramString

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

Returns a string representation of this JProgressBar. This method is intended to be used only for debugging purposes. The content and format of the returned string may vary between implementations. The returned string may be empty but may not be null.

**Overrides:**[paramString](http://docs.google.com/javax/swing/JComponent.html#paramString()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**a string representation of this JProgressBar

### getAccessibleContext

public [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) **getAccessibleContext**()

Gets the AccessibleContext associated with this JProgressBar. For progress bars, the AccessibleContext takes the form of an AccessibleJProgressBar. A new AccessibleJProgressBar instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/javax/swing/JComponent.html#getAccessibleContext()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**an AccessibleJProgressBar that serves as the AccessibleContext of this JProgressBar

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JProgressBar.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/JPopupMenu.Separator.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JProgressBar.AccessibleJProgressBar.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JProgressBar.html)    [**NO FRAMES**](http://docs.google.com/JProgressBar.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#qsh70q) | [METHOD](#3o7alnk) |

[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).