| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JViewport.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/JTree.TreeSelectionRedirector.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JViewport.AccessibleJViewport.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JViewport.html)    [**NO FRAMES**](http://docs.google.com/JViewport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#qsh70q) | [METHOD](#1pxezwc) |

## **javax.swing**

Class JViewport

[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.JViewport**

**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)

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

The "viewport" or "porthole" through which you see the underlying information. When you scroll, what moves is the viewport. It is like peering through a camera's viewfinder. Moving the viewfinder upwards brings new things into view at the top of the picture and loses things that were at the bottom.

By default, JViewport is opaque. To change this, use the setOpaque method.

**NOTE:**We have implemented a faster scrolling algorithm that does not require a buffer to draw in. The algorithm works as follows:

1. The view and parent view and checked to see if they are JComponents, if they aren't, stop and repaint the whole viewport.
2. If the viewport is obscured by an ancestor, stop and repaint the whole viewport.
3. Compute the region that will become visible, if it is as big as the viewport, stop and repaint the whole view region.
4. Obtain the ancestor Window's graphics and do a copyArea on the scrolled region.
5. Message the view to repaint the newly visible region.
6. The next time paint is invoked on the viewport, if the clip region is smaller than the viewport size a timer is kicked off to repaint the whole region.

In general this approach is much faster. Compared to the backing store approach this avoids the overhead of maintaining an offscreen buffer and having to do two copyAreas. Compared to the non backing store case this approach will greatly reduce the painted region.

This approach can cause slower times than the backing store approach when the viewport is obscured by another window, or partially offscreen. When another window obscures the viewport the copyArea will copy garbage and a paint event will be generated by the system to inform us we need to paint the newly exposed region. The only way to handle this is to repaint the whole viewport, which can cause slower performance than the backing store case. In most applications very rarely will the user be scrolling while the viewport is obscured by another window or offscreen, so this optimization is usually worth the performance hit when obscured.

**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:**[JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JViewport.AccessibleJViewport**](http://docs.google.com/javax/swing/JViewport.AccessibleJViewport.html)            This class implements accessibility support for the JViewport class. |
| protected  class | [**JViewport.ViewListener**](http://docs.google.com/javax/swing/JViewport.ViewListener.html)            A listener for the view. |

| **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  boolean | [**backingStore**](http://docs.google.com/javax/swing/JViewport.html#backingStore)  **Deprecated.** *As of Java 2 platform v1.3* |
| static int | [**BACKINGSTORE\_SCROLL\_MODE**](http://docs.google.com/javax/swing/JViewport.html#BACKINGSTORE_SCROLL_MODE)            Draws viewport contents into an offscreen image. |
| protected  [Image](http://docs.google.com/java/awt/Image.html) | [**backingStoreImage**](http://docs.google.com/javax/swing/JViewport.html#backingStoreImage)            The view image used for a backing store. |
| static int | [**BLIT\_SCROLL\_MODE**](http://docs.google.com/javax/swing/JViewport.html#BLIT_SCROLL_MODE)            Use graphics.copyArea to implement scrolling. |
| protected  boolean | [**isViewSizeSet**](http://docs.google.com/javax/swing/JViewport.html#isViewSizeSet)            True when the viewport dimensions have been determined. |
| protected  [Point](http://docs.google.com/java/awt/Point.html) | [**lastPaintPosition**](http://docs.google.com/javax/swing/JViewport.html#lastPaintPosition)            The last viewPosition that we've painted, so we know how much of the backing store image is valid. |
| protected  boolean | [**scrollUnderway**](http://docs.google.com/javax/swing/JViewport.html#scrollUnderway)            The scrollUnderway flag is used for components like JList. |
| static int | [**SIMPLE\_SCROLL\_MODE**](http://docs.google.com/javax/swing/JViewport.html#SIMPLE_SCROLL_MODE)            This mode uses the very simple method of redrawing the entire contents of the scrollpane each time it is scrolled. |

| **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 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** | |
| --- | --- |
| [**JViewport**](http://docs.google.com/javax/swing/JViewport.html#JViewport())()            Creates a JViewport. |

| **Method Summary** | |
| --- | --- |
| void | [**addChangeListener**](http://docs.google.com/javax/swing/JViewport.html#addChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Adds a ChangeListener to the list that is notified each time the view's size, position, or the viewport's extent size has changed. |
| protected  void | [**addImpl**](http://docs.google.com/javax/swing/JViewport.html#addImpl(java.awt.Component,%20java.lang.Object,%20int))([Component](http://docs.google.com/java/awt/Component.html) child, [Object](http://docs.google.com/java/lang/Object.html) constraints, int index)            Sets the JViewport's one lightweight child, which can be null. |
| protected  boolean | [**computeBlit**](http://docs.google.com/javax/swing/JViewport.html#computeBlit(int,%20int,%20java.awt.Point,%20java.awt.Point,%20java.awt.Dimension,%20java.awt.Rectangle))(int dx, int dy, [Point](http://docs.google.com/java/awt/Point.html) blitFrom, [Point](http://docs.google.com/java/awt/Point.html) blitTo, [Dimension](http://docs.google.com/java/awt/Dimension.html) blitSize, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) blitPaint)            Computes the parameters for a blit where the backing store image currently contains oldLoc in the upper left hand corner and we're scrolling to newLoc. |
| protected  [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) | [**createLayoutManager**](http://docs.google.com/javax/swing/JViewport.html#createLayoutManager())()            Subclassers can override this to install a different layout manager (or null) in the constructor. |
| protected  [JViewport.ViewListener](http://docs.google.com/javax/swing/JViewport.ViewListener.html) | [**createViewListener**](http://docs.google.com/javax/swing/JViewport.html#createViewListener())()            Creates a listener for the view. |
| protected  void | [**firePropertyChange**](http://docs.google.com/javax/swing/JViewport.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) propertyName, [Object](http://docs.google.com/java/lang/Object.html) oldValue, [Object](http://docs.google.com/java/lang/Object.html) newValue)            Notifies listeners of a property change. |
| protected  void | [**fireStateChanged**](http://docs.google.com/javax/swing/JViewport.html#fireStateChanged())()            Notifies all ChangeListeners when the views size, position, or the viewports extent size has changed. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JViewport.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JViewport. |
| [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html)[] | [**getChangeListeners**](http://docs.google.com/javax/swing/JViewport.html#getChangeListeners())()            Returns an array of all the ChangeListeners added to this JViewport with addChangeListener(). |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getExtentSize**](http://docs.google.com/javax/swing/JViewport.html#getExtentSize())()            Returns the size of the visible part of the view in view coordinates. |
| [Insets](http://docs.google.com/java/awt/Insets.html) | [**getInsets**](http://docs.google.com/javax/swing/JViewport.html#getInsets())()            Returns the insets (border) dimensions as (0,0,0,0), since borders are not supported on a JViewport. |
| [Insets](http://docs.google.com/java/awt/Insets.html) | [**getInsets**](http://docs.google.com/javax/swing/JViewport.html#getInsets(java.awt.Insets))([Insets](http://docs.google.com/java/awt/Insets.html) insets)            Returns an Insets object containing this JViewports inset values. |
| int | [**getScrollMode**](http://docs.google.com/javax/swing/JViewport.html#getScrollMode())()            Returns the current scrolling mode. |
| [ViewportUI](http://docs.google.com/javax/swing/plaf/ViewportUI.html) | [**getUI**](http://docs.google.com/javax/swing/JViewport.html#getUI())()            Returns the L&F object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JViewport.html#getUIClassID())()            Returns a string that specifies the name of the L&F class that renders this component. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getView**](http://docs.google.com/javax/swing/JViewport.html#getView())()            Returns the JViewport's one child or null. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getViewPosition**](http://docs.google.com/javax/swing/JViewport.html#getViewPosition())()            Returns the view coordinates that appear in the upper left hand corner of the viewport, or 0,0 if there's no view. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getViewRect**](http://docs.google.com/javax/swing/JViewport.html#getViewRect())()            Returns a rectangle whose origin is getViewPosition and size is getExtentSize. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getViewSize**](http://docs.google.com/javax/swing/JViewport.html#getViewSize())()            If the view's size hasn't been explicitly set, return the preferred size, otherwise return the view's current size. |
| boolean | [**isBackingStoreEnabled**](http://docs.google.com/javax/swing/JViewport.html#isBackingStoreEnabled())()  **Deprecated.** *As of Java 2 platform v1.3, replaced by getScrollMode().* |
| boolean | [**isOptimizedDrawingEnabled**](http://docs.google.com/javax/swing/JViewport.html#isOptimizedDrawingEnabled())()            The JViewport overrides the default implementation of this method (in JComponent) to return false. |
| void | [**paint**](http://docs.google.com/javax/swing/JViewport.html#paint(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Depending on whether the backingStore is enabled, either paint the image through the backing store or paint just the recently exposed part, using the backing store to "blit" the remainder. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JViewport.html#paramString())()            Returns a string representation of this JViewport. |
| void | [**remove**](http://docs.google.com/javax/swing/JViewport.html#remove(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) child)            Removes the Viewports one lightweight child. |
| void | [**removeChangeListener**](http://docs.google.com/javax/swing/JViewport.html#removeChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Removes a ChangeListener from the list that's notified each time the views size, position, or the viewports extent size has changed. |
| void | [**repaint**](http://docs.google.com/javax/swing/JViewport.html#repaint(long,%20int,%20int,%20int,%20int))(long tm, int x, int y, int w, int h)            Always repaint in the parents coordinate system to make sure only one paint is performed by the RepaintManager. |
| void | [**reshape**](http://docs.google.com/javax/swing/JViewport.html#reshape(int,%20int,%20int,%20int))(int x, int y, int w, int h)            Sets the bounds of this viewport. |
| void | [**scrollRectToVisible**](http://docs.google.com/javax/swing/JViewport.html#scrollRectToVisible(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) contentRect)            Scrolls the view so that Rectangle within the view becomes visible. |
| void | [**setBackingStoreEnabled**](http://docs.google.com/javax/swing/JViewport.html#setBackingStoreEnabled(boolean))(boolean enabled)  **Deprecated.** *As of Java 2 platform v1.3, replaced by setScrollMode().* |
| void | [**setBorder**](http://docs.google.com/javax/swing/JViewport.html#setBorder(javax.swing.border.Border))([Border](http://docs.google.com/javax/swing/border/Border.html) border)            The viewport "scrolls" its child (called the "view") by the normal parent/child clipping (typically the view is moved in the opposite direction of the scroll). |
| void | [**setExtentSize**](http://docs.google.com/javax/swing/JViewport.html#setExtentSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) newExtent)            Sets the size of the visible part of the view using view coordinates. |
| void | [**setScrollMode**](http://docs.google.com/javax/swing/JViewport.html#setScrollMode(int))(int mode)            Used to control the method of scrolling the viewport contents. |
| void | [**setUI**](http://docs.google.com/javax/swing/JViewport.html#setUI(javax.swing.plaf.ViewportUI))([ViewportUI](http://docs.google.com/javax/swing/plaf/ViewportUI.html) ui)            Sets the L&F object that renders this component. |
| void | [**setView**](http://docs.google.com/javax/swing/JViewport.html#setView(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) view)            Sets the JViewport's one lightweight child (view), which can be null. |
| void | [**setViewPosition**](http://docs.google.com/javax/swing/JViewport.html#setViewPosition(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Sets the view coordinates that appear in the upper left hand corner of the viewport, does nothing if there's no view. |
| void | [**setViewSize**](http://docs.google.com/javax/swing/JViewport.html#setViewSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) newSize)            Sets the size of the view. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**toViewCoordinates**](http://docs.google.com/javax/swing/JViewport.html#toViewCoordinates(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) size)            Converts a size in pixel coordinates to view coordinates. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**toViewCoordinates**](http://docs.google.com/javax/swing/JViewport.html#toViewCoordinates(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Converts a point in pixel coordinates to view coordinates. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JViewport.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()), [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()), [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()), [paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(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(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()), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [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)), [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)), [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(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,%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** |
| --- |

### isViewSizeSet

protected boolean **isViewSizeSet**

True when the viewport dimensions have been determined. The default is false.

### lastPaintPosition

protected [Point](http://docs.google.com/java/awt/Point.html) **lastPaintPosition**

The last viewPosition that we've painted, so we know how much of the backing store image is valid.

### backingStore

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected boolean **backingStore**

**Deprecated.** *As of Java 2 platform v1.3*True when this viewport is maintaining an offscreen image of its contents, so that some scrolling can take place using fast "bit-blit" operations instead of by accessing the view object to construct the display. The default is false.

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

### backingStoreImage

protected transient [Image](http://docs.google.com/java/awt/Image.html) **backingStoreImage**

The view image used for a backing store.

### scrollUnderway

protected boolean **scrollUnderway**

The scrollUnderway flag is used for components like JList. When the downarrow key is pressed on a JList and the selected cell is the last in the list, the scrollpane autoscrolls. Here, the old selected cell needs repainting and so we need a flag to make the viewport do the optimized painting only when there is an explicit call to setViewPosition(Point). When setBounds is called through other routes, the flag is off and the view repaints normally. Another approach would be to remove this from the JViewport class and have the JList manage this case by using setBackingStoreEnabled. The default is false.

### BLIT\_SCROLL\_MODE

public static final int **BLIT\_SCROLL\_MODE**

Use graphics.copyArea to implement scrolling. This is the fastest for most applications.

**Since:** 1.3 **See Also:**[setScrollMode(int)](http://docs.google.com/javax/swing/JViewport.html#setScrollMode(int)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JViewport.BLIT_SCROLL_MODE)

### BACKINGSTORE\_SCROLL\_MODE

public static final int **BACKINGSTORE\_SCROLL\_MODE**

Draws viewport contents into an offscreen image. This was previously the default mode for JTable. This mode may offer advantages over "blit mode" in some cases, but it requires a large chunk of extra RAM.

**Since:** 1.3 **See Also:**[setScrollMode(int)](http://docs.google.com/javax/swing/JViewport.html#setScrollMode(int)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JViewport.BACKINGSTORE_SCROLL_MODE)

### SIMPLE\_SCROLL\_MODE

public static final int **SIMPLE\_SCROLL\_MODE**

This mode uses the very simple method of redrawing the entire contents of the scrollpane each time it is scrolled. This was the default behavior in Swing 1.0 and Swing 1.1. Either of the other two options will provide better performance in most cases.

**Since:** 1.3 **See Also:**[setScrollMode(int)](http://docs.google.com/javax/swing/JViewport.html#setScrollMode(int)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JViewport.SIMPLE_SCROLL_MODE)

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

### JViewport

public **JViewport**()

Creates a JViewport.

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

### getUI

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

Returns the L&F object that renders this component.

**Returns:**a ViewportUI object**Since:** 1.3

### setUI

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

Sets the L&F object that renders this component.

**Parameters:**ui - the ViewportUI L&F object**Since:** 1.3 **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 a string that specifies the name of the L&F 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 "ViewportUI"**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))

### addImpl

protected void **addImpl**([Component](http://docs.google.com/java/awt/Component.html) child,  
 [Object](http://docs.google.com/java/lang/Object.html) constraints,  
 int index)

Sets the JViewport's one lightweight child, which can be null. (Since there is only one child which occupies the entire viewport, the constraints and index arguments are ignored.)

**Overrides:**[addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**child - the lightweight child of the viewportconstraints - the constraints to be respectedindex - the index**See Also:**[setView(java.awt.Component)](http://docs.google.com/javax/swing/JViewport.html#setView(java.awt.Component))

### remove

public void **remove**([Component](http://docs.google.com/java/awt/Component.html) child)

Removes the Viewports one lightweight child.

**Overrides:**[remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**child - the component to be removed**See Also:**[setView(java.awt.Component)](http://docs.google.com/javax/swing/JViewport.html#setView(java.awt.Component))

### scrollRectToVisible

public void **scrollRectToVisible**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) contentRect)

Scrolls the view so that Rectangle within the view becomes visible.

This attempts to validate the view before scrolling if the view is currently not valid - isValid returns false. To avoid excessive validation when the containment hierarchy is being created this will not validate if one of the ancestors does not have a peer, or there is no validate root ancestor, or one of the ancestors is not a Window or Applet.

Note that this method will not scroll outside of the valid viewport; for example, if contentRect is larger than the viewport, scrolling will be confined to the viewport's bounds.

**Overrides:**[scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**contentRect - the Rectangle to display**See Also:**[JComponent.isValidateRoot()](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()), [Component.isValid()](http://docs.google.com/java/awt/Component.html#isValid()), [Component.getPeer()](http://docs.google.com/java/awt/Component.html#getPeer())

### setBorder

public final void **setBorder**([Border](http://docs.google.com/javax/swing/border/Border.html) border)

The viewport "scrolls" its child (called the "view") by the normal parent/child clipping (typically the view is moved in the opposite direction of the scroll). A non-null border, or non-zero insets, isn't supported, to prevent the geometry of this component from becoming complex enough to inhibit subclassing. To create a JViewport with a border, add it to a JPanel that has a border.

Note: If border is non-null, this method will throw an exception as borders are not supported on a JViewPort.

**Overrides:**[setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**border - the Border to set **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - this method is not implemented**See Also:**[Border](http://docs.google.com/javax/swing/border/Border.html), [CompoundBorder](http://docs.google.com/javax/swing/border/CompoundBorder.html)

### getInsets

public final [Insets](http://docs.google.com/java/awt/Insets.html) **getInsets**()

Returns the insets (border) dimensions as (0,0,0,0), since borders are not supported on a JViewport.

**Overrides:**[getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**a Rectange of zero dimension and zero origin**See Also:**[setBorder(javax.swing.border.Border)](http://docs.google.com/javax/swing/JViewport.html#setBorder(javax.swing.border.Border))

### getInsets

public final [Insets](http://docs.google.com/java/awt/Insets.html) **getInsets**([Insets](http://docs.google.com/java/awt/Insets.html) insets)

Returns an Insets object containing this JViewports inset values. The passed-in Insets object will be reinitialized, and all existing values within this object are overwritten.

**Overrides:**[getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**insets - the Insets object which can be reused **Returns:**this viewports inset values**See Also:**[getInsets()](http://docs.google.com/javax/swing/JViewport.html#getInsets())

### isOptimizedDrawingEnabled

public boolean **isOptimizedDrawingEnabled**()

The JViewport overrides the default implementation of this method (in JComponent) to return false. This ensures that the drawing machinery will call the Viewport's paint implementation rather than messaging the JViewport's children directly.

**Overrides:**[isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**false

### paint

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

Depending on whether the backingStore is enabled, either paint the image through the backing store or paint just the recently exposed part, using the backing store to "blit" the remainder.The term "blit" is the pronounced version of the PDP-10 BLT (BLock Transfer) instruction, which copied a block of bits. (In case you were curious.)

**Overrides:**[paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**g - the Graphics context within which to paint**See Also:**[JComponent.paintComponent(java.awt.Graphics)](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)), [JComponent.paintBorder(java.awt.Graphics)](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)), [JComponent.paintChildren(java.awt.Graphics)](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [JComponent.getComponentGraphics(java.awt.Graphics)](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [JComponent.repaint(long, int, int, int, int)](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int))

### reshape

public void **reshape**(int x,  
 int y,  
 int w,  
 int h)

Sets the bounds of this viewport. If the viewport's width or height has changed, fire a StateChanged event.

**Overrides:**[reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**x - left edge of the originy - top edge of the originw - width in pixelsh - height in pixels**See Also:**[JComponent.reshape(int, int, int, int)](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int))

### setScrollMode

public void **setScrollMode**(int mode)

Used to control the method of scrolling the viewport contents. You may want to change this mode to get maximum performance for your use case.

**Parameters:**mode - one of the following values:

* JViewport.BLIT\_SCROLL\_MODE
* JViewport.BACKINGSTORE\_SCROLL\_MODE
* JViewport.SIMPLE\_SCROLL\_MODE

**Since:** 1.3 **See Also:**[BLIT\_SCROLL\_MODE](http://docs.google.com/javax/swing/JViewport.html#BLIT_SCROLL_MODE), [BACKINGSTORE\_SCROLL\_MODE](http://docs.google.com/javax/swing/JViewport.html#BACKINGSTORE_SCROLL_MODE), [SIMPLE\_SCROLL\_MODE](http://docs.google.com/javax/swing/JViewport.html#SIMPLE_SCROLL_MODE)

### getScrollMode

public int **getScrollMode**()

Returns the current scrolling mode.

**Returns:**the scrollMode property**Since:** 1.3 **See Also:**[setScrollMode(int)](http://docs.google.com/javax/swing/JViewport.html#setScrollMode(int))

### isBackingStoreEnabled

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **isBackingStoreEnabled**()

**Deprecated.** *As of Java 2 platform v1.3, replaced by getScrollMode().*

Returns true if this viewport is maintaining an offscreen image of its contents.

**Returns:**true if scrollMode is BACKINGSTORE\_SCROLL\_MODE

### setBackingStoreEnabled

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **setBackingStoreEnabled**(boolean enabled)

**Deprecated.** *As of Java 2 platform v1.3, replaced by setScrollMode().*

If true if this viewport will maintain an offscreen image of its contents. The image is used to reduce the cost of small one dimensional changes to the viewPosition. Rather than repainting the entire viewport we use Graphics.copyArea to effect some of the scroll.

**Parameters:**enabled - if true, maintain an offscreen backing store

### getView

public [Component](http://docs.google.com/java/awt/Component.html) **getView**()

Returns the JViewport's one child or null.

**Returns:**the viewports child, or null if none exists**See Also:**[setView(java.awt.Component)](http://docs.google.com/javax/swing/JViewport.html#setView(java.awt.Component))

### setView

public void **setView**([Component](http://docs.google.com/java/awt/Component.html) view)

Sets the JViewport's one lightweight child (view), which can be null.

**Parameters:**view - the viewport's new lightweight child**See Also:**[getView()](http://docs.google.com/javax/swing/JViewport.html#getView())

### getViewSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getViewSize**()

If the view's size hasn't been explicitly set, return the preferred size, otherwise return the view's current size. If there is no view, return 0,0.

**Returns:**a Dimension object specifying the size of the view

### setViewSize

public void **setViewSize**([Dimension](http://docs.google.com/java/awt/Dimension.html) newSize)

Sets the size of the view. A state changed event will be fired.

**Parameters:**newSize - a Dimension object specifying the new size of the view

### getViewPosition

public [Point](http://docs.google.com/java/awt/Point.html) **getViewPosition**()

Returns the view coordinates that appear in the upper left hand corner of the viewport, or 0,0 if there's no view.

**Returns:**a Point object giving the upper left coordinates

### setViewPosition

public void **setViewPosition**([Point](http://docs.google.com/java/awt/Point.html) p)

Sets the view coordinates that appear in the upper left hand corner of the viewport, does nothing if there's no view.

**Parameters:**p - a Point object giving the upper left coordinates

### getViewRect

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getViewRect**()

Returns a rectangle whose origin is getViewPosition and size is getExtentSize. This is the visible part of the view, in view coordinates.

**Returns:**a Rectangle giving the visible part of the view using view coordinates.

### computeBlit

protected boolean **computeBlit**(int dx,  
 int dy,  
 [Point](http://docs.google.com/java/awt/Point.html) blitFrom,  
 [Point](http://docs.google.com/java/awt/Point.html) blitTo,  
 [Dimension](http://docs.google.com/java/awt/Dimension.html) blitSize,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) blitPaint)

Computes the parameters for a blit where the backing store image currently contains oldLoc in the upper left hand corner and we're scrolling to newLoc. The parameters are modified to return the values required for the blit.

**Parameters:**dx - the horizontal deltady - the vertical deltablitFrom - the Point we're blitting fromblitTo - the Point we're blitting toblitSize - the Dimension of the area to blitblitPaint - the area to blit **Returns:**true if the parameters are modified and we're ready to blit; false otherwise

### getExtentSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getExtentSize**()

Returns the size of the visible part of the view in view coordinates.

**Returns:**a Dimension object giving the size of the view

### toViewCoordinates

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **toViewCoordinates**([Dimension](http://docs.google.com/java/awt/Dimension.html) size)

Converts a size in pixel coordinates to view coordinates. Subclasses of viewport that support "logical coordinates" will override this method.

**Parameters:**size - a Dimension object using pixel coordinates **Returns:**a Dimension object converted to view coordinates

### toViewCoordinates

public [Point](http://docs.google.com/java/awt/Point.html) **toViewCoordinates**([Point](http://docs.google.com/java/awt/Point.html) p)

Converts a point in pixel coordinates to view coordinates. Subclasses of viewport that support "logical coordinates" will override this method.

**Parameters:**p - a Point object using pixel coordinates **Returns:**a Point object converted to view coordinates

### setExtentSize

public void **setExtentSize**([Dimension](http://docs.google.com/java/awt/Dimension.html) newExtent)

Sets the size of the visible part of the view using view coordinates.

**Parameters:**newExtent - a Dimension object specifying the size of the view

### createViewListener

protected [JViewport.ViewListener](http://docs.google.com/javax/swing/JViewport.ViewListener.html) **createViewListener**()

Creates a listener for the view.

**Returns:**a ViewListener

### createLayoutManager

protected [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **createLayoutManager**()

Subclassers can override this to install a different layout manager (or null) in the constructor. Returns the LayoutManager to install on the JViewport.

**Returns:**a LayoutManager

### addChangeListener

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

Adds a ChangeListener to the list that is notified each time the view's size, position, or the viewport's extent size has changed.

**Parameters:**l - the ChangeListener to add**See Also:**[removeChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JViewport.html#removeChangeListener(javax.swing.event.ChangeListener)), [setViewPosition(java.awt.Point)](http://docs.google.com/javax/swing/JViewport.html#setViewPosition(java.awt.Point)), [setViewSize(java.awt.Dimension)](http://docs.google.com/javax/swing/JViewport.html#setViewSize(java.awt.Dimension)), [setExtentSize(java.awt.Dimension)](http://docs.google.com/javax/swing/JViewport.html#setExtentSize(java.awt.Dimension))

### removeChangeListener

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

Removes a ChangeListener from the list that's notified each time the views size, position, or the viewports extent size has changed.

**Parameters:**l - the ChangeListener to remove**See Also:**[addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JViewport.html#addChangeListener(javax.swing.event.ChangeListener))

### getChangeListeners

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

Returns an array of all the ChangeListeners added to this JViewport 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**()

Notifies all ChangeListeners when the views size, position, or the viewports extent size has changed.

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

### repaint

public void **repaint**(long tm,  
 int x,  
 int y,  
 int w,  
 int h)

Always repaint in the parents coordinate system to make sure only one paint is performed by the RepaintManager.

**Overrides:**[repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**tm - maximum time in milliseconds before updatex - the x coordinate (pixels over from left)y - the y coordinate (pixels down from top)w - the widthh - the height**See Also:**[Component.update(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### paramString

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

Returns a string representation of this JViewport. This method is intended to be used only for debugging purposes, and 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 JViewport

### firePropertyChange

protected void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 [Object](http://docs.google.com/java/lang/Object.html) oldValue,  
 [Object](http://docs.google.com/java/lang/Object.html) newValue)

Notifies listeners of a property change. This is subclassed to update the windowBlit property. (The putClientProperty property is final).

**Overrides:**[firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**propertyName - a string containing the property nameoldValue - the old value of the propertynewValue - the new value of the property

### getAccessibleContext

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

Gets the AccessibleContext associated with this JViewport. For viewports, the AccessibleContext takes the form of an AccessibleJViewport. A new AccessibleJViewport 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 AccessibleJViewport that serves as the AccessibleContext of this JViewport

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JViewport.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/JTree.TreeSelectionRedirector.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JViewport.AccessibleJViewport.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JViewport.html)    [**NO FRAMES**](http://docs.google.com/JViewport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#qsh70q) | [METHOD](#1pxezwc) |

[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).