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

## **javax.swing**

Interface Scrollable

**All Known Implementing Classes:** [DefaultTreeCellEditor.DefaultTextField](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.DefaultTextField.html), [JEditorPane](http://docs.google.com/javax/swing/JEditorPane.html), [JFormattedTextField](http://docs.google.com/javax/swing/JFormattedTextField.html), [JList](http://docs.google.com/javax/swing/JList.html), [JPasswordField](http://docs.google.com/javax/swing/JPasswordField.html), [JTable](http://docs.google.com/javax/swing/JTable.html), [JTextArea](http://docs.google.com/javax/swing/JTextArea.html), [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html), [JTextField](http://docs.google.com/javax/swing/JTextField.html), [JTextPane](http://docs.google.com/javax/swing/JTextPane.html), [JTree](http://docs.google.com/javax/swing/JTree.html)

public interface **Scrollable**

An interface that provides information to a scrolling container like JScrollPane. A complex component that's likely to be used as a viewing a JScrollPane viewport (or other scrolling container) should implement this interface.

**See Also:**[JViewport](http://docs.google.com/javax/swing/JViewport.html), [JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html), [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html)

| **Method Summary** | |
| --- | --- |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredScrollableViewportSize**](http://docs.google.com/javax/swing/Scrollable.html#getPreferredScrollableViewportSize())()            Returns the preferred size of the viewport for a view component. |
| int | [**getScrollableBlockIncrement**](http://docs.google.com/javax/swing/Scrollable.html#getScrollableBlockIncrement(java.awt.Rectangle,%20int,%20int))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) visibleRect, int orientation, int direction)            Components that display logical rows or columns should compute the scroll increment that will completely expose one block of rows or columns, depending on the value of orientation. |
| boolean | [**getScrollableTracksViewportHeight**](http://docs.google.com/javax/swing/Scrollable.html#getScrollableTracksViewportHeight())()            Return true if a viewport should always force the height of this Scrollable to match the height of the viewport. |
| boolean | [**getScrollableTracksViewportWidth**](http://docs.google.com/javax/swing/Scrollable.html#getScrollableTracksViewportWidth())()            Return true if a viewport should always force the width of this Scrollable to match the width of the viewport. |
| int | [**getScrollableUnitIncrement**](http://docs.google.com/javax/swing/Scrollable.html#getScrollableUnitIncrement(java.awt.Rectangle,%20int,%20int))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) visibleRect, int orientation, int direction)            Components that display logical rows or columns should compute the scroll increment that will completely expose one new row or column, depending on the value of orientation. |

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

### getPreferredScrollableViewportSize

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

Returns the preferred size of the viewport for a view component. For example, the preferred size of a JList component is the size required to accommodate all of the cells in its list. However, the value of preferredScrollableViewportSize is the size required for JList.getVisibleRowCount rows. A component without any properties that would affect the viewport size should just return getPreferredSize here.

**Returns:**the preferredSize of a JViewport whose view is this Scrollable**See Also:**[JComponent.getPreferredSize()](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize())

### getScrollableUnitIncrement

int **getScrollableUnitIncrement**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) visibleRect,  
 int orientation,  
 int direction)

Components that display logical rows or columns should compute the scroll increment that will completely expose one new row or column, depending on the value of orientation. Ideally, components should handle a partially exposed row or column by returning the distance required to completely expose the item.

Scrolling containers, like JScrollPane, will use this method each time the user requests a unit scroll.

**Parameters:**visibleRect - The view area visible within the viewportorientation - Either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL.direction - Less than zero to scroll up/left, greater than zero for down/right. **Returns:**The "unit" increment for scrolling in the specified direction. This value should always be positive.**See Also:**[JScrollBar.setUnitIncrement(int)](http://docs.google.com/javax/swing/JScrollBar.html#setUnitIncrement(int))

### getScrollableBlockIncrement

int **getScrollableBlockIncrement**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) visibleRect,  
 int orientation,  
 int direction)

Components that display logical rows or columns should compute the scroll increment that will completely expose one block of rows or columns, depending on the value of orientation.

Scrolling containers, like JScrollPane, will use this method each time the user requests a block scroll.

**Parameters:**visibleRect - The view area visible within the viewportorientation - Either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL.direction - Less than zero to scroll up/left, greater than zero for down/right. **Returns:**The "block" increment for scrolling in the specified direction. This value should always be positive.**See Also:**[JScrollBar.setBlockIncrement(int)](http://docs.google.com/javax/swing/JScrollBar.html#setBlockIncrement(int))

### getScrollableTracksViewportWidth

boolean **getScrollableTracksViewportWidth**()

Return true if a viewport should always force the width of this Scrollable to match the width of the viewport. For example a normal text view that supported line wrapping would return true here, since it would be undesirable for wrapped lines to disappear beyond the right edge of the viewport. Note that returning true for a Scrollable whose ancestor is a JScrollPane effectively disables horizontal scrolling.

Scrolling containers, like JViewport, will use this method each time they are validated.

**Returns:**True if a viewport should force the Scrollables width to match its own.

### getScrollableTracksViewportHeight

boolean **getScrollableTracksViewportHeight**()

Return true if a viewport should always force the height of this Scrollable to match the height of the viewport. For example a columnar text view that flowed text in left to right columns could effectively disable vertical scrolling by returning true here.

Scrolling containers, like JViewport, will use this method each time they are validated.

**Returns:**True if a viewport should force the Scrollables height to match its own.

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

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).