| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicSplitPaneDivider.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSpinnerUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicSplitPaneDivider.html)    [**NO FRAMES**](http://docs.google.com/BasicSplitPaneDivider.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#35nkun2) | [CONSTR](#qsh70q) | [METHOD](#1pxezwc) |

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

Class BasicSplitPaneDivider

[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.plaf.basic.BasicSplitPaneDivider**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [EventListener](http://docs.google.com/java/util/EventListener.html)

public class **BasicSplitPaneDivider**extends [Container](http://docs.google.com/java/awt/Container.html)implements [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html)

Divider used by BasicSplitPaneUI. Subclassers may wish to override paint to do something more interesting. The border effect is drawn in BasicSplitPaneUI, so if you don't like that border, reset it there. To conditionally drag from certain areas subclass mousePressed and call super when you wish the dragging to begin.

**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).

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**BasicSplitPaneDivider.DividerLayout**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html)            Used to layout a BasicSplitPaneDivider. |
| protected  class | [**BasicSplitPaneDivider.DragController**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DragController.html)            Handles the events during a dragging session for a HORIZONTAL\_SPLIT oriented split pane. |
| protected  class | [**BasicSplitPaneDivider.MouseHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.MouseHandler.html)            MouseHandler is responsible for converting mouse events (released, dragged...) into the appropriate DragController methods. |
| protected  class | [**BasicSplitPaneDivider.VerticalDragController**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.VerticalDragController.html)            Handles the events during a dragging session for a VERTICAL\_SPLIT oriented split pane. |

| **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  int | [**dividerSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#dividerSize)            Size of the divider. |
| protected  [BasicSplitPaneDivider.DragController](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DragController.html) | [**dragger**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#dragger)            Handles mouse dragging message to do the actual dragging. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**hiddenDivider**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#hiddenDivider)            Divider that is used for noncontinuous layout mode. |
| protected  [JButton](http://docs.google.com/javax/swing/JButton.html) | [**leftButton**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#leftButton)            Button for quickly toggling the left component. |
| protected  [BasicSplitPaneDivider.MouseHandler](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.MouseHandler.html) | [**mouseHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#mouseHandler)            Handles mouse events from both this class, and the split pane. |
| protected static int | [**ONE\_TOUCH\_OFFSET**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#ONE_TOUCH_OFFSET) |
| protected static int | [**ONE\_TOUCH\_SIZE**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#ONE_TOUCH_SIZE)            Width or height of the divider based on orientation BasicSplitPaneUI adds two to this. |
| protected  int | [**orientation**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#orientation)            Orientation of the JSplitPane. |
| protected  [JButton](http://docs.google.com/javax/swing/JButton.html) | [**rightButton**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#rightButton)            Button for quickly toggling the right component. |
| protected  [JSplitPane](http://docs.google.com/javax/swing/JSplitPane.html) | [**splitPane**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#splitPane)            JSplitPane the receiver is contained in. |
| protected  [BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html) | [**splitPaneUI**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#splitPaneUI)            UI this instance was created from. |

| **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** | |
| --- | --- |
| [**BasicSplitPaneDivider**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#BasicSplitPaneDivider(javax.swing.plaf.basic.BasicSplitPaneUI))([BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html) ui)            Creates an instance of BasicSplitPaneDivider. |

| **Method Summary** | |
| --- | --- |
| protected  [JButton](http://docs.google.com/javax/swing/JButton.html) | [**createLeftOneTouchButton**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#createLeftOneTouchButton())()            Creates and return an instance of JButton that can be used to collapse the left component in the split pane. |
| protected  [JButton](http://docs.google.com/javax/swing/JButton.html) | [**createRightOneTouchButton**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#createRightOneTouchButton())()            Creates and return an instance of JButton that can be used to collapse the right component in the split pane. |
| protected  void | [**dragDividerTo**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#dragDividerTo(int))(int location)            Messages the BasicSplitPaneUI with dragDividerTo that this instance is contained in. |
| protected  void | [**finishDraggingTo**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#finishDraggingTo(int))(int location)            Messages the BasicSplitPaneUI with finishDraggingTo that this instance is contained in. |
| [BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html) | [**getBasicSplitPaneUI**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#getBasicSplitPaneUI())()            Returns the SplitPaneUI the receiver is currently in. |
| [Border](http://docs.google.com/javax/swing/border/Border.html) | [**getBorder**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#getBorder())()            Returns the border of this component or null if no border is currently set. |
| int | [**getDividerSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#getDividerSize())()            Returns the size of the divider, that is the width if the splitpane is HORIZONTAL\_SPLIT, or the height of VERTICAL\_SPLIT. |
| [Insets](http://docs.google.com/java/awt/Insets.html) | [**getInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#getInsets())()            If a border has been set on this component, returns the border's insets, else calls super.getInsets. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#getMinimumSize())()            Returns dividerSize x dividerSize |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#getPreferredSize())()            Returns dividerSize x dividerSize |
| boolean | [**isMouseOver**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#isMouseOver())()            Returns whether or not the mouse is currently over the divider |
| protected  void | [**oneTouchExpandableChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#oneTouchExpandableChanged())()            Messaged when the oneTouchExpandable value of the JSplitPane the receiver is contained in changes. |
| void | [**paint**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#paint(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Paints the divider. |
| protected  void | [**prepareForDragging**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#prepareForDragging())()            Message to prepare for dragging. |
| void | [**propertyChange**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#propertyChange(java.beans.PropertyChangeEvent))([PropertyChangeEvent](http://docs.google.com/java/beans/PropertyChangeEvent.html) e)            Property change event, presumably from the JSplitPane, will message updateOrientation if necessary. |
| void | [**setBasicSplitPaneUI**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#setBasicSplitPaneUI(javax.swing.plaf.basic.BasicSplitPaneUI))([BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html) newUI)            Sets the SplitPaneUI that is using the receiver. |
| void | [**setBorder**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#setBorder(javax.swing.border.Border))([Border](http://docs.google.com/javax/swing/border/Border.html) border)            Sets the border of this component. |
| void | [**setDividerSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#setDividerSize(int))(int newSize)            Sets the size of the divider to newSize. |
| protected  void | [**setMouseOver**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#setMouseOver(boolean))(boolean mouseOver)            Sets whether or not the mouse is currently over the divider. |

| **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)), [addNotify](http://docs.google.com/java/awt/Container.html#addNotify()), [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)), [getAlignmentX](http://docs.google.com/java/awt/Container.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/java/awt/Container.html#getAlignmentY()), [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()), [getListeners](http://docs.google.com/java/awt/Container.html#getListeners(java.lang.Class)), [getMaximumSize](http://docs.google.com/java/awt/Container.html#getMaximumSize()), [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)), [paramString](http://docs.google.com/java/awt/Container.html#paramString()), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [print](http://docs.google.com/java/awt/Container.html#print(java.awt.Graphics)), [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)), [removeNotify](http://docs.google.com/java/awt/Container.html#removeNotify()), [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)), [setFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setFont](http://docs.google.com/java/awt/Container.html#setFont(java.awt.Font)), [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()), [update](http://docs.google.com/java/awt/Container.html#update(java.awt.Graphics)), [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(int,%20int)), [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)), [disable](http://docs.google.com/java/awt/Component.html#disable()), [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()), [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,%20boolean,%20boolean)), [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,%20char,%20char)), [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,%20int,%20int)), [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)), [getAccessibleContext](http://docs.google.com/java/awt/Component.html#getAccessibleContext()), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBaseline](http://docs.google.com/java/awt/Component.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/java/awt/Component.html#getBaselineResizeBehavior()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds(java.awt.Rectangle)), [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()), [getFontMetrics](http://docs.google.com/java/awt/Component.html#getFontMetrics(java.awt.Font)), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphics](http://docs.google.com/java/awt/Component.html#getGraphics()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHeight](http://docs.google.com/java/awt/Component.html#getHeight()), [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()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation(java.awt.Point)), [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()), [getSize](http://docs.google.com/java/awt/Component.html#getSize(java.awt.Dimension)), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [getWidth](http://docs.google.com/java/awt/Component.html#getWidth()), [getX](http://docs.google.com/java/awt/Component.html#getX()), [getY](http://docs.google.com/java/awt/Component.html#getY()), [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()), [isDoubleBuffered](http://docs.google.com/java/awt/Component.html#isDoubleBuffered()), [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()), [isOpaque](http://docs.google.com/java/awt/Component.html#isOpaque()), [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)), [printAll](http://docs.google.com/java/awt/Component.html#printAll(java.awt.Graphics)), [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)), [processKeyEvent](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [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)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long,%20int,%20int,%20int,%20int)), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus()), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow(boolean)), [reshape](http://docs.google.com/java/awt/Component.html#reshape(int,%20int,%20int,%20int)), [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)), [setBackground](http://docs.google.com/java/awt/Component.html#setBackground(java.awt.Color)), [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)), [setEnabled](http://docs.google.com/java/awt/Component.html#setEnabled(boolean)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setForeground](http://docs.google.com/java/awt/Component.html#setForeground(java.awt.Color)), [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)), [setMaximumSize](http://docs.google.com/java/awt/Component.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/java/awt/Component.html#setMinimumSize(java.awt.Dimension)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setPreferredSize](http://docs.google.com/java/awt/Component.html#setPreferredSize(java.awt.Dimension)), [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)), [setVisible](http://docs.google.com/java/awt/Component.html#setVisible(boolean)), [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** |
| --- |

### ONE\_TOUCH\_SIZE

protected static final int **ONE\_TOUCH\_SIZE**

Width or height of the divider based on orientation BasicSplitPaneUI adds two to this.

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

### ONE\_TOUCH\_OFFSET

protected static final int **ONE\_TOUCH\_OFFSET**

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

### dragger

protected [BasicSplitPaneDivider.DragController](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DragController.html) **dragger**

Handles mouse dragging message to do the actual dragging.

### splitPaneUI

protected [BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html) **splitPaneUI**

UI this instance was created from.

### dividerSize

protected int **dividerSize**

Size of the divider.

### hiddenDivider

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

Divider that is used for noncontinuous layout mode.

### splitPane

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

JSplitPane the receiver is contained in.

### mouseHandler

protected [BasicSplitPaneDivider.MouseHandler](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.MouseHandler.html) **mouseHandler**

Handles mouse events from both this class, and the split pane. Mouse events are handled for the splitpane since you want to be able to drag when clicking on the border of the divider, which is not drawn by the divider.

### orientation

protected int **orientation**

Orientation of the JSplitPane.

### leftButton

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

Button for quickly toggling the left component.

### rightButton

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

Button for quickly toggling the right component.

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

### BasicSplitPaneDivider

public **BasicSplitPaneDivider**([BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html) ui)

Creates an instance of BasicSplitPaneDivider. Registers this instance for mouse events and mouse dragged events.

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

### setBasicSplitPaneUI

public void **setBasicSplitPaneUI**([BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html) newUI)

Sets the SplitPaneUI that is using the receiver.

### getBasicSplitPaneUI

public [BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html) **getBasicSplitPaneUI**()

Returns the SplitPaneUI the receiver is currently in.

### setDividerSize

public void **setDividerSize**(int newSize)

Sets the size of the divider to newSize. That is the width if the splitpane is HORIZONTAL\_SPLIT, or the height of VERTICAL\_SPLIT.

### getDividerSize

public int **getDividerSize**()

Returns the size of the divider, that is the width if the splitpane is HORIZONTAL\_SPLIT, or the height of VERTICAL\_SPLIT.

### setBorder

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

Sets the border of this component.

**Since:** 1.3

### getBorder

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

Returns the border of this component or null if no border is currently set.

**Returns:**the border object for this component**Since:** 1.3 **See Also:**[setBorder(javax.swing.border.Border)](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#setBorder(javax.swing.border.Border))

### getInsets

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

If a border has been set on this component, returns the border's insets, else calls super.getInsets.

**Overrides:**[getInsets](http://docs.google.com/java/awt/Container.html#getInsets()) in class [Container](http://docs.google.com/java/awt/Container.html) **Returns:**the value of the insets property.**See Also:**[setBorder(javax.swing.border.Border)](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html#setBorder(javax.swing.border.Border))

### setMouseOver

protected void **setMouseOver**(boolean mouseOver)

Sets whether or not the mouse is currently over the divider.

**Parameters:**mouseOver - whether or not the mouse is currently over the divider**Since:** 1.5

### isMouseOver

public boolean **isMouseOver**()

Returns whether or not the mouse is currently over the divider

**Returns:**whether or not the mouse is currently over the divider**Since:** 1.5

### getPreferredSize

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

Returns dividerSize x dividerSize

**Overrides:**[getPreferredSize](http://docs.google.com/java/awt/Container.html#getPreferredSize()) in class [Container](http://docs.google.com/java/awt/Container.html) **Returns:**an instance of Dimension that represents the preferred size of this container.**See Also:**[Container.getMinimumSize()](http://docs.google.com/java/awt/Container.html#getMinimumSize()), [Container.getMaximumSize()](http://docs.google.com/java/awt/Container.html#getMaximumSize()), [Container.getLayout()](http://docs.google.com/java/awt/Container.html#getLayout()), [LayoutManager.preferredLayoutSize(Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container)), [Component.getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize())

### getMinimumSize

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

Returns dividerSize x dividerSize

**Overrides:**[getMinimumSize](http://docs.google.com/java/awt/Container.html#getMinimumSize()) in class [Container](http://docs.google.com/java/awt/Container.html) **Returns:**an instance of Dimension that represents the minimum size of this container.**See Also:**[Container.getPreferredSize()](http://docs.google.com/java/awt/Container.html#getPreferredSize()), [Container.getMaximumSize()](http://docs.google.com/java/awt/Container.html#getMaximumSize()), [Container.getLayout()](http://docs.google.com/java/awt/Container.html#getLayout()), [LayoutManager.minimumLayoutSize(Container)](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)), [Component.getMinimumSize()](http://docs.google.com/java/awt/Component.html#getMinimumSize())

### propertyChange

public void **propertyChange**([PropertyChangeEvent](http://docs.google.com/java/beans/PropertyChangeEvent.html) e)

Property change event, presumably from the JSplitPane, will message updateOrientation if necessary.

**Specified by:**[propertyChange](http://docs.google.com/java/beans/PropertyChangeListener.html#propertyChange(java.beans.PropertyChangeEvent)) in interface [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **Parameters:**e - A PropertyChangeEvent object describing the event source and the property that has changed.

### paint

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

Paints the divider.

**Overrides:**[paint](http://docs.google.com/java/awt/Container.html#paint(java.awt.Graphics)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**g - the specified Graphics window**See Also:**[Component.update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### oneTouchExpandableChanged

protected void **oneTouchExpandableChanged**()

Messaged when the oneTouchExpandable value of the JSplitPane the receiver is contained in changes. Will create the leftButton and rightButton if they are null. invalidates the receiver as well.

### createLeftOneTouchButton

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

Creates and return an instance of JButton that can be used to collapse the left component in the split pane.

### createRightOneTouchButton

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

Creates and return an instance of JButton that can be used to collapse the right component in the split pane.

### prepareForDragging

protected void **prepareForDragging**()

Message to prepare for dragging. This messages the BasicSplitPaneUI with startDragging.

### dragDividerTo

protected void **dragDividerTo**(int location)

Messages the BasicSplitPaneUI with dragDividerTo that this instance is contained in.

### finishDraggingTo

protected void **finishDraggingTo**(int location)

Messages the BasicSplitPaneUI with finishDraggingTo that this instance is contained in.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicSplitPaneDivider.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSpinnerUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicSplitPaneDivider.html)    [**NO FRAMES**](http://docs.google.com/BasicSplitPaneDivider.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#35nkun2) | [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).