| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicSplitPaneUI.BasicHorizontalLayoutManager.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/BasicSplitPaneUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicVerticalLayoutManager.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html)    [**NO FRAMES**](http://docs.google.com/BasicSplitPaneUI.BasicHorizontalLayoutManager.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#2s8eyo1) |

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

Class BasicSplitPaneUI.BasicHorizontalLayoutManager

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.plaf.basic.BasicSplitPaneUI.BasicHorizontalLayoutManager**

**All Implemented Interfaces:** [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html) **Direct Known Subclasses:** [BasicSplitPaneUI.BasicVerticalLayoutManager](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicVerticalLayoutManager.html) **Enclosing class:**[BasicSplitPaneUI](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.html)

public class **BasicSplitPaneUI.BasicHorizontalLayoutManager**extends [Object](http://docs.google.com/java/lang/Object.html)implements [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html)

LayoutManager for JSplitPanes that have an orientation of HORIZONTAL\_SPLIT.

| **Field Summary** | |
| --- | --- |
| protected  [Component](http://docs.google.com/java/awt/Component.html)[] | [**components**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#components) |
| protected  int[] | [**sizes**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#sizes) |

| **Method Summary** | |
| --- | --- |
| void | [**addLayoutComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#addLayoutComponent(java.awt.Component,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) comp, [Object](http://docs.google.com/java/lang/Object.html) constraints)            Adds the specified component to the layout, using the specified constraint object. |
| void | [**addLayoutComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#addLayoutComponent(java.lang.String,%20java.awt.Component))([String](http://docs.google.com/java/lang/String.html) place, [Component](http://docs.google.com/java/awt/Component.html) component)            Adds the component at place. |
| protected  int | [**getAvailableSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#getAvailableSize(java.awt.Dimension,%20java.awt.Insets))([Dimension](http://docs.google.com/java/awt/Dimension.html) containerSize, [Insets](http://docs.google.com/java/awt/Insets.html) insets)            Returns the available width based on the container size and Insets. |
| protected  int | [**getInitialLocation**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#getInitialLocation(java.awt.Insets))([Insets](http://docs.google.com/java/awt/Insets.html) insets)            Returns the left inset, unless the Insets are null in which case 0 is returned. |
| float | [**getLayoutAlignmentX**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#getLayoutAlignmentX(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the alignment along the x axis. |
| float | [**getLayoutAlignmentY**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#getLayoutAlignmentY(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the alignment along the y axis. |
| protected  int | [**getPreferredSizeOfComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#getPreferredSizeOfComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Returns the width of the passed in Components preferred size. |
| protected  int | [**getSizeOfComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#getSizeOfComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Returns the width of the passed in component. |
| protected  int[] | [**getSizes**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#getSizes())()            Returns the sizes of the components. |
| void | [**invalidateLayout**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#invalidateLayout(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) c)            Does nothing. |
| void | [**layoutContainer**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#layoutContainer(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) container)            Does the actual layout. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**maximumLayoutSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#maximumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the maximum layout size, which is Integer.MAX\_VALUE in both directions. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumLayoutSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#minimumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) container)            Returns the minimum size needed to contain the children. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredLayoutSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#preferredLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) container)            Returns the preferred size needed to contain the children. |
| void | [**removeLayoutComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#removeLayoutComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) component)            Removes the specified component from our knowledge. |
| protected  void | [**resetSizeAt**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#resetSizeAt(int))(int index)            Resets the size of the Component at the passed in location. |
| void | [**resetToPreferredSizes**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#resetToPreferredSizes())()            Marks the receiver so that the next time this instance is laid out it'll ask for the preferred sizes. |
| protected  void | [**setComponentToSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#setComponentToSize(java.awt.Component,%20int,%20int,%20java.awt.Insets,%20java.awt.Dimension))([Component](http://docs.google.com/java/awt/Component.html) c, int size, int location, [Insets](http://docs.google.com/java/awt/Insets.html) insets, [Dimension](http://docs.google.com/java/awt/Dimension.html) containerSize)            Sets the width of the component c to be size, placing its x location at location, y to the insets.top and height to the containersize.height less the top and bottom insets. |
| protected  void | [**setSizes**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#setSizes(int%5B%5D))(int[] newSizes)            Sets the sizes to newSizes. |
| protected  void | [**updateComponents**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html#updateComponents())()            Determines the components. |

| **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()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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** |
| --- |

### sizes

protected int[] **sizes**

### components

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

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

### layoutContainer

public void **layoutContainer**([Container](http://docs.google.com/java/awt/Container.html) container)

Does the actual layout.

**Specified by:**[layoutContainer](http://docs.google.com/java/awt/LayoutManager.html#layoutContainer(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**container - the container to be laid out

### addLayoutComponent

public void **addLayoutComponent**([String](http://docs.google.com/java/lang/String.html) place,  
 [Component](http://docs.google.com/java/awt/Component.html) component)

Adds the component at place. Place must be one of JSplitPane.LEFT, RIGHT, TOP, BOTTOM, or null (for the divider).

**Specified by:**[addLayoutComponent](http://docs.google.com/java/awt/LayoutManager.html#addLayoutComponent(java.lang.String,%20java.awt.Component)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**place - the string to be associated with the componentcomponent - the component to be added

### minimumLayoutSize

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

Returns the minimum size needed to contain the children. The width is the sum of all the childrens min widths and the height is the largest of the childrens minimum heights.

**Specified by:**[minimumLayoutSize](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**container - the component to be laid out**See Also:**[LayoutManager.preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container))

### preferredLayoutSize

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

Returns the preferred size needed to contain the children. The width is the sum of all the childrens preferred widths and the height is the largest of the childrens preferred heights.

**Specified by:**[preferredLayoutSize](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**container - the container to be laid out**See Also:**[LayoutManager.minimumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container))

### removeLayoutComponent

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

Removes the specified component from our knowledge.

**Specified by:**[removeLayoutComponent](http://docs.google.com/java/awt/LayoutManager.html#removeLayoutComponent(java.awt.Component)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**component - the component to be removed

### addLayoutComponent

public void **addLayoutComponent**([Component](http://docs.google.com/java/awt/Component.html) comp,  
 [Object](http://docs.google.com/java/lang/Object.html) constraints)

Adds the specified component to the layout, using the specified constraint object.

**Specified by:**[addLayoutComponent](http://docs.google.com/java/awt/LayoutManager2.html#addLayoutComponent(java.awt.Component,%20java.lang.Object)) in interface [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html) **Parameters:**comp - the component to be addedconstraints - where/how the component is added to the layout.

### getLayoutAlignmentX

public float **getLayoutAlignmentX**([Container](http://docs.google.com/java/awt/Container.html) target)

Returns the alignment along the x axis. This specifies how the component would like to be aligned relative to other components. The value should be a number between 0 and 1 where 0 represents alignment along the origin, 1 is aligned the furthest away from the origin, 0.5 is centered, etc.

**Specified by:**[getLayoutAlignmentX](http://docs.google.com/java/awt/LayoutManager2.html#getLayoutAlignmentX(java.awt.Container)) in interface [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html)

### getLayoutAlignmentY

public float **getLayoutAlignmentY**([Container](http://docs.google.com/java/awt/Container.html) target)

Returns the alignment along the y axis. This specifies how the component would like to be aligned relative to other components. The value should be a number between 0 and 1 where 0 represents alignment along the origin, 1 is aligned the furthest away from the origin, 0.5 is centered, etc.

**Specified by:**[getLayoutAlignmentY](http://docs.google.com/java/awt/LayoutManager2.html#getLayoutAlignmentY(java.awt.Container)) in interface [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html)

### invalidateLayout

public void **invalidateLayout**([Container](http://docs.google.com/java/awt/Container.html) c)

Does nothing. If the developer really wants to change the size of one of the views JSplitPane.resetToPreferredSizes should be messaged.

**Specified by:**[invalidateLayout](http://docs.google.com/java/awt/LayoutManager2.html#invalidateLayout(java.awt.Container)) in interface [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html)

### maximumLayoutSize

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

Returns the maximum layout size, which is Integer.MAX\_VALUE in both directions.

**Specified by:**[maximumLayoutSize](http://docs.google.com/java/awt/LayoutManager2.html#maximumLayoutSize(java.awt.Container)) in interface [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html) **See Also:**[Component.getMaximumSize()](http://docs.google.com/java/awt/Component.html#getMaximumSize()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### resetToPreferredSizes

public void **resetToPreferredSizes**()

Marks the receiver so that the next time this instance is laid out it'll ask for the preferred sizes.

### resetSizeAt

protected void **resetSizeAt**(int index)

Resets the size of the Component at the passed in location.

### setSizes

protected void **setSizes**(int[] newSizes)

Sets the sizes to newSizes.

### getSizes

protected int[] **getSizes**()

Returns the sizes of the components.

### getPreferredSizeOfComponent

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

Returns the width of the passed in Components preferred size.

### getSizeOfComponent

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

Returns the width of the passed in component.

### getAvailableSize

protected int **getAvailableSize**([Dimension](http://docs.google.com/java/awt/Dimension.html) containerSize,  
 [Insets](http://docs.google.com/java/awt/Insets.html) insets)

Returns the available width based on the container size and Insets.

### getInitialLocation

protected int **getInitialLocation**([Insets](http://docs.google.com/java/awt/Insets.html) insets)

Returns the left inset, unless the Insets are null in which case 0 is returned.

### setComponentToSize

protected void **setComponentToSize**([Component](http://docs.google.com/java/awt/Component.html) c,  
 int size,  
 int location,  
 [Insets](http://docs.google.com/java/awt/Insets.html) insets,  
 [Dimension](http://docs.google.com/java/awt/Dimension.html) containerSize)

Sets the width of the component c to be size, placing its x location at location, y to the insets.top and height to the containersize.height less the top and bottom insets.

### updateComponents

protected void **updateComponents**()

Determines the components. This should be called whenever a new instance of this is installed into an existing SplitPane.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicSplitPaneUI.BasicHorizontalLayoutManager.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/BasicSplitPaneUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneUI.BasicVerticalLayoutManager.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicSplitPaneUI.BasicHorizontalLayoutManager.html)    [**NO FRAMES**](http://docs.google.com/BasicSplitPaneUI.BasicHorizontalLayoutManager.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#2s8eyo1) |

[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).