| | [**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.DividerLayout.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/BasicSplitPaneDivider.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DragController.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html)    [**NO FRAMES**](http://docs.google.com/BasicSplitPaneDivider.DividerLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

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

Class BasicSplitPaneDivider.DividerLayout

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

**All Implemented Interfaces:** [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Enclosing class:**[BasicSplitPaneDivider](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.html)

protected class **BasicSplitPaneDivider.DividerLayout**extends [Object](http://docs.google.com/java/lang/Object.html)implements [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

Used to layout a BasicSplitPaneDivider. Layout for the divider involves appropriately moving the left/right buttons around.

| **Constructor Summary** | |
| --- | --- |
| protected | [**BasicSplitPaneDivider.DividerLayout**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html#BasicSplitPaneDivider.DividerLayout())() |

| **Method Summary** | |
| --- | --- |
| void | [**addLayoutComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html#addLayoutComponent(java.lang.String,%20java.awt.Component))([String](http://docs.google.com/java/lang/String.html) string, [Component](http://docs.google.com/java/awt/Component.html) c)            If the layout manager uses a per-component string, adds the component comp to the layout, associating it with the string specified by name. |
| void | [**layoutContainer**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html#layoutContainer(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) c)            Lays out the specified container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumLayoutSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html#minimumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) c)            Calculates the minimum size dimensions for the specified container, given the components it contains. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredLayoutSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html#preferredLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) c)            Calculates the preferred size dimensions for the specified container, given the components it contains. |
| void | [**removeLayoutComponent**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html#removeLayoutComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Removes the specified component from the layout. |

| **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)) |

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

### BasicSplitPaneDivider.DividerLayout

protected **BasicSplitPaneDivider.DividerLayout**()

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

### layoutContainer

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

**Description copied from interface:** [**LayoutManager**](http://docs.google.com/java/awt/LayoutManager.html#layoutContainer(java.awt.Container)) Lays out the specified container.

**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:**c - the container to be laid out

### minimumLayoutSize

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

**Description copied from interface:** [**LayoutManager**](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)) Calculates the minimum size dimensions for the specified container, given the components it contains.

**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:**c - 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) c)

**Description copied from interface:** [**LayoutManager**](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container)) Calculates the preferred size dimensions for the specified container, given the components it contains.

**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:**c - 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) c)

**Description copied from interface:** [**LayoutManager**](http://docs.google.com/java/awt/LayoutManager.html#removeLayoutComponent(java.awt.Component)) Removes the specified component from the layout.

**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:**c - the component to be removed

### addLayoutComponent

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

**Description copied from interface:** [**LayoutManager**](http://docs.google.com/java/awt/LayoutManager.html#addLayoutComponent(java.lang.String,%20java.awt.Component)) If the layout manager uses a per-component string, adds the component comp to the layout, associating it with the string specified by name.

**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:**string - the string to be associated with the componentc - the component to be added

| | [**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.DividerLayout.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/BasicSplitPaneDivider.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSplitPaneDivider.DragController.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicSplitPaneDivider.DividerLayout.html)    [**NO FRAMES**](http://docs.google.com/BasicSplitPaneDivider.DividerLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).