| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BorderLayout.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/java/awt/BasicStroke.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/BufferCapabilities.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/BorderLayout.html)    [**NO FRAMES**](http://docs.google.com/BorderLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#4i7ojhp) | [METHOD](#3whwml4) |

## **java.awt**

Class BorderLayout

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.awt.BorderLayout**

**All Implemented Interfaces:** [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

public class **BorderLayout**extends [Object](http://docs.google.com/java/lang/Object.html)implements [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

A border layout lays out a container, arranging and resizing its components to fit in five regions: north, south, east, west, and center. Each region may contain no more than one component, and is identified by a corresponding constant: NORTH, SOUTH, EAST, WEST, and CENTER. When adding a component to a container with a border layout, use one of these five constants, for example:

Panel p = new Panel();  
 p.setLayout(new BorderLayout());  
 p.add(new Button("Okay"), BorderLayout.SOUTH);

As a convenience, BorderLayout interprets the absence of a string specification the same as the constant CENTER:

Panel p2 = new Panel();  
 p2.setLayout(new BorderLayout());  
 p2.add(new TextArea()); // Same as p.add(new TextArea(), BorderLayout.CENTER);

In addition, BorderLayout supports the relative positioning constants, PAGE\_START, PAGE\_END, LINE\_START, and LINE\_END. In a container whose ComponentOrientation is set to ComponentOrientation.LEFT\_TO\_RIGHT, these constants map to NORTH, SOUTH, WEST, and EAST, respectively.

For compatibility with previous releases, BorderLayout also includes the relative positioning constants BEFORE\_FIRST\_LINE, AFTER\_LAST\_LINE, BEFORE\_LINE\_BEGINS and AFTER\_LINE\_ENDS. These are equivalent to PAGE\_START, PAGE\_END, LINE\_START and LINE\_END respectively. For consistency with the relative positioning constants used by other components, the latter constants are preferred.

Mixing both absolute and relative positioning constants can lead to unpredicable results. If you use both types, the relative constants will take precedence. For example, if you add components using both the NORTH and PAGE\_START constants in a container whose orientation is LEFT\_TO\_RIGHT, only the PAGE\_START will be layed out.

NOTE: Currently (in the Java 2 platform v1.2), BorderLayout does not support vertical orientations. The isVertical setting on the container's ComponentOrientation is not respected.

The components are laid out according to their preferred sizes and the constraints of the container's size. The NORTH and SOUTH components may be stretched horizontally; the EAST and WEST components may be stretched vertically; the CENTER component may stretch both horizontally and vertically to fill any space left over.

Here is an example of five buttons in an applet laid out using the BorderLayout layout manager:



The code for this applet is as follows:

import java.awt.\*;  
 import java.applet.Applet;  
  
 public class buttonDir extends Applet {  
 public void init() {  
 setLayout(new BorderLayout());  
 add(new Button("North"), BorderLayout.NORTH);  
 add(new Button("South"), BorderLayout.SOUTH);  
 add(new Button("East"), BorderLayout.EAST);  
 add(new Button("West"), BorderLayout.WEST);  
 add(new Button("Center"), BorderLayout.CENTER);  
 }  
 }

**Since:** JDK1.0 **See Also:**[Container.add(String, Component)](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.BorderLayout)

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**AFTER\_LAST\_LINE**](http://docs.google.com/java/awt/BorderLayout.html#AFTER_LAST_LINE)            Synonym for PAGE\_END. |
| static [String](http://docs.google.com/java/lang/String.html) | [**AFTER\_LINE\_ENDS**](http://docs.google.com/java/awt/BorderLayout.html#AFTER_LINE_ENDS)            Synonym for LINE\_END. |
| static [String](http://docs.google.com/java/lang/String.html) | [**BEFORE\_FIRST\_LINE**](http://docs.google.com/java/awt/BorderLayout.html#BEFORE_FIRST_LINE)            Synonym for PAGE\_START. |
| static [String](http://docs.google.com/java/lang/String.html) | [**BEFORE\_LINE\_BEGINS**](http://docs.google.com/java/awt/BorderLayout.html#BEFORE_LINE_BEGINS)            Synonym for LINE\_START. |
| static [String](http://docs.google.com/java/lang/String.html) | [**CENTER**](http://docs.google.com/java/awt/BorderLayout.html#CENTER)            The center layout constraint (middle of container). |
| static [String](http://docs.google.com/java/lang/String.html) | [**EAST**](http://docs.google.com/java/awt/BorderLayout.html#EAST)            The east layout constraint (right side of container). |
| static [String](http://docs.google.com/java/lang/String.html) | [**LINE\_END**](http://docs.google.com/java/awt/BorderLayout.html#LINE_END)            The component goes at the end of the line direction for the layout. |
| static [String](http://docs.google.com/java/lang/String.html) | [**LINE\_START**](http://docs.google.com/java/awt/BorderLayout.html#LINE_START)            The component goes at the beginning of the line direction for the layout. |
| static [String](http://docs.google.com/java/lang/String.html) | [**NORTH**](http://docs.google.com/java/awt/BorderLayout.html#NORTH)            The north layout constraint (top of container). |
| static [String](http://docs.google.com/java/lang/String.html) | [**PAGE\_END**](http://docs.google.com/java/awt/BorderLayout.html#PAGE_END)            The component comes after the last line of the layout's content. |
| static [String](http://docs.google.com/java/lang/String.html) | [**PAGE\_START**](http://docs.google.com/java/awt/BorderLayout.html#PAGE_START)            The component comes before the first line of the layout's content. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SOUTH**](http://docs.google.com/java/awt/BorderLayout.html#SOUTH)            The south layout constraint (bottom of container). |
| static [String](http://docs.google.com/java/lang/String.html) | [**WEST**](http://docs.google.com/java/awt/BorderLayout.html#WEST)            The west layout constraint (left side of container). |

| **Constructor Summary** | |
| --- | --- |
| [**BorderLayout**](http://docs.google.com/java/awt/BorderLayout.html#BorderLayout())()            Constructs a new border layout with no gaps between components. |
| [**BorderLayout**](http://docs.google.com/java/awt/BorderLayout.html#BorderLayout(int,%20int))(int hgap, int vgap)            Constructs a border layout with the specified gaps between components. |

| **Method Summary** | |
| --- | --- |
| void | [**addLayoutComponent**](http://docs.google.com/java/awt/BorderLayout.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/java/awt/BorderLayout.html#addLayoutComponent(java.lang.String,%20java.awt.Component))([String](http://docs.google.com/java/lang/String.html) name, [Component](http://docs.google.com/java/awt/Component.html) comp)  **Deprecated.** *replaced by addLayoutComponent(Component, Object).* |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getConstraints**](http://docs.google.com/java/awt/BorderLayout.html#getConstraints(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Gets the constraints for the specified component |
| int | [**getHgap**](http://docs.google.com/java/awt/BorderLayout.html#getHgap())()            Returns the horizontal gap between components. |
| float | [**getLayoutAlignmentX**](http://docs.google.com/java/awt/BorderLayout.html#getLayoutAlignmentX(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Returns the alignment along the x axis. |
| float | [**getLayoutAlignmentY**](http://docs.google.com/java/awt/BorderLayout.html#getLayoutAlignmentY(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Returns the alignment along the y axis. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getLayoutComponent**](http://docs.google.com/java/awt/BorderLayout.html#getLayoutComponent(java.awt.Container,%20java.lang.Object))([Container](http://docs.google.com/java/awt/Container.html) target, [Object](http://docs.google.com/java/lang/Object.html) constraints)            Returns the component that corresponds to the given constraint location based on the target Container's component orientation. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getLayoutComponent**](http://docs.google.com/java/awt/BorderLayout.html#getLayoutComponent(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) constraints)            Gets the component that was added using the given constraint |
| int | [**getVgap**](http://docs.google.com/java/awt/BorderLayout.html#getVgap())()            Returns the vertical gap between components. |
| void | [**invalidateLayout**](http://docs.google.com/java/awt/BorderLayout.html#invalidateLayout(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Invalidates the layout, indicating that if the layout manager has cached information it should be discarded. |
| void | [**layoutContainer**](http://docs.google.com/java/awt/BorderLayout.html#layoutContainer(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Lays out the container argument using this border layout. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**maximumLayoutSize**](http://docs.google.com/java/awt/BorderLayout.html#maximumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the maximum dimensions for this layout given the components in the specified target container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumLayoutSize**](http://docs.google.com/java/awt/BorderLayout.html#minimumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Determines the minimum size of the target container using this layout manager. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredLayoutSize**](http://docs.google.com/java/awt/BorderLayout.html#preferredLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Determines the preferred size of the target container using this layout manager, based on the components in the container. |
| void | [**removeLayoutComponent**](http://docs.google.com/java/awt/BorderLayout.html#removeLayoutComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Removes the specified component from this border layout. |
| void | [**setHgap**](http://docs.google.com/java/awt/BorderLayout.html#setHgap(int))(int hgap)            Sets the horizontal gap between components. |
| void | [**setVgap**](http://docs.google.com/java/awt/BorderLayout.html#setVgap(int))(int vgap)            Sets the vertical gap between components. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/awt/BorderLayout.html#toString())()            Returns a string representation of the state of this border 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()), [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** |
| --- |

### NORTH

public static final [String](http://docs.google.com/java/lang/String.html) **NORTH**

The north layout constraint (top of container).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.NORTH)

### SOUTH

public static final [String](http://docs.google.com/java/lang/String.html) **SOUTH**

The south layout constraint (bottom of container).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.SOUTH)

### EAST

public static final [String](http://docs.google.com/java/lang/String.html) **EAST**

The east layout constraint (right side of container).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.EAST)

### WEST

public static final [String](http://docs.google.com/java/lang/String.html) **WEST**

The west layout constraint (left side of container).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.WEST)

### CENTER

public static final [String](http://docs.google.com/java/lang/String.html) **CENTER**

The center layout constraint (middle of container).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.CENTER)

### BEFORE\_FIRST\_LINE

public static final [String](http://docs.google.com/java/lang/String.html) **BEFORE\_FIRST\_LINE**

Synonym for PAGE\_START. Exists for compatibility with previous versions. PAGE\_START is preferred.

**Since:** 1.2 **See Also:**[PAGE\_START](http://docs.google.com/java/awt/BorderLayout.html#PAGE_START), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.BEFORE_FIRST_LINE)

### AFTER\_LAST\_LINE

public static final [String](http://docs.google.com/java/lang/String.html) **AFTER\_LAST\_LINE**

Synonym for PAGE\_END. Exists for compatibility with previous versions. PAGE\_END is preferred.

**Since:** 1.2 **See Also:**[PAGE\_END](http://docs.google.com/java/awt/BorderLayout.html#PAGE_END), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.AFTER_LAST_LINE)

### BEFORE\_LINE\_BEGINS

public static final [String](http://docs.google.com/java/lang/String.html) **BEFORE\_LINE\_BEGINS**

Synonym for LINE\_START. Exists for compatibility with previous versions. LINE\_START is preferred.

**Since:** 1.2 **See Also:**[LINE\_START](http://docs.google.com/java/awt/BorderLayout.html#LINE_START), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.BEFORE_LINE_BEGINS)

### AFTER\_LINE\_ENDS

public static final [String](http://docs.google.com/java/lang/String.html) **AFTER\_LINE\_ENDS**

Synonym for LINE\_END. Exists for compatibility with previous versions. LINE\_END is preferred.

**Since:** 1.2 **See Also:**[LINE\_END](http://docs.google.com/java/awt/BorderLayout.html#LINE_END), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.AFTER_LINE_ENDS)

### PAGE\_START

public static final [String](http://docs.google.com/java/lang/String.html) **PAGE\_START**

The component comes before the first line of the layout's content. For Western, left-to-right and top-to-bottom orientations, this is equivalent to NORTH.

**Since:** 1.4 **See Also:**[Component.getComponentOrientation()](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.PAGE_START)

### PAGE\_END

public static final [String](http://docs.google.com/java/lang/String.html) **PAGE\_END**

The component comes after the last line of the layout's content. For Western, left-to-right and top-to-bottom orientations, this is equivalent to SOUTH.

**Since:** 1.4 **See Also:**[Component.getComponentOrientation()](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.PAGE_END)

### LINE\_START

public static final [String](http://docs.google.com/java/lang/String.html) **LINE\_START**

The component goes at the beginning of the line direction for the layout. For Western, left-to-right and top-to-bottom orientations, this is equivalent to WEST.

**Since:** 1.4 **See Also:**[Component.getComponentOrientation()](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.LINE_START)

### LINE\_END

public static final [String](http://docs.google.com/java/lang/String.html) **LINE\_END**

The component goes at the end of the line direction for the layout. For Western, left-to-right and top-to-bottom orientations, this is equivalent to EAST.

**Since:** 1.4 **See Also:**[Component.getComponentOrientation()](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.BorderLayout.LINE_END)

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

### BorderLayout

public **BorderLayout**()

Constructs a new border layout with no gaps between components.

### BorderLayout

public **BorderLayout**(int hgap,  
 int vgap)

Constructs a border layout with the specified gaps between components. The horizontal gap is specified by hgap and the vertical gap is specified by vgap.

**Parameters:**hgap - the horizontal gap.vgap - the vertical gap.

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

### getHgap

public int **getHgap**()

Returns the horizontal gap between components.

**Since:** JDK1.1

### setHgap

public void **setHgap**(int hgap)

Sets the horizontal gap between components.

**Parameters:**hgap - the horizontal gap between components**Since:** JDK1.1

### getVgap

public int **getVgap**()

Returns the vertical gap between components.

**Since:** JDK1.1

### setVgap

public void **setVgap**(int vgap)

Sets the vertical gap between components.

**Parameters:**vgap - the vertical gap between components**Since:** JDK1.1

### 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. For border layouts, the constraint must be one of the following constants: NORTH, SOUTH, EAST, WEST, or CENTER.

Most applications do not call this method directly. This method is called when a component is added to a container using the Container.add method with the same argument types.

**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 added.constraints - an object that specifies how and where the component is added to the layout. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the constraint object is not a string, or if it not one of the five specified constants.**Since:** JDK1.1 **See Also:**[Container.add(java.awt.Component, java.lang.Object)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object))

### addLayoutComponent

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **addLayoutComponent**([String](http://docs.google.com/java/lang/String.html) name,  
 [Component](http://docs.google.com/java/awt/Component.html) comp)

**Deprecated.** *replaced by addLayoutComponent(Component, Object).*

**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:**name - the string to be associated with the componentcomp - the component to be added

### removeLayoutComponent

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

Removes the specified component from this border layout. This method is called when a container calls its remove or removeAll methods. Most applications do not call this method directly.

**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:**comp - the component to be removed.**See Also:**[Container.remove(java.awt.Component)](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [Container.removeAll()](http://docs.google.com/java/awt/Container.html#removeAll())

### getLayoutComponent

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

Gets the component that was added using the given constraint

**Parameters:**constraints - the desired constraint, one of CENTER, NORTH, SOUTH, WEST, EAST, PAGE\_START, PAGE\_END, LINE\_START, LINE\_END **Returns:**the component at the given location, or null if the location is empty **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the constraint object is not one of the nine specified constants**Since:** 1.5 **See Also:**[addLayoutComponent(java.awt.Component, java.lang.Object)](http://docs.google.com/java/awt/BorderLayout.html#addLayoutComponent(java.awt.Component,%20java.lang.Object))

### getLayoutComponent

public [Component](http://docs.google.com/java/awt/Component.html) **getLayoutComponent**([Container](http://docs.google.com/java/awt/Container.html) target,  
 [Object](http://docs.google.com/java/lang/Object.html) constraints)

Returns the component that corresponds to the given constraint location based on the target Container's component orientation. Components added with the relative constraints PAGE\_START, PAGE\_END, LINE\_START, and LINE\_END take precedence over components added with the explicit constraints NORTH, SOUTH, WEST, and EAST. The Container's component orientation is used to determine the location of components added with LINE\_START and LINE\_END.

**Parameters:**constraints - the desired absolute position, one of CENTER, NORTH, SOUTH, EAST, WESTtarget - the Container used to obtain the constraint location based on the target Container's component orientation. **Returns:**the component at the given location, or null if the location is empty **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the constraint object is not one of the five specified constants [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the target parameter is null**Since:** 1.5 **See Also:**[addLayoutComponent(java.awt.Component, java.lang.Object)](http://docs.google.com/java/awt/BorderLayout.html#addLayoutComponent(java.awt.Component,%20java.lang.Object))

### getConstraints

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

Gets the constraints for the specified component

**Parameters:**comp - the component to be queried **Returns:**the constraint for the specified component, or null if component is null or is not present in this layout**Since:** 1.5 **See Also:**[addLayoutComponent(java.awt.Component, java.lang.Object)](http://docs.google.com/java/awt/BorderLayout.html#addLayoutComponent(java.awt.Component,%20java.lang.Object))

### minimumLayoutSize

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

Determines the minimum size of the target container using this layout manager.

This method is called when a container calls its getMinimumSize method. Most applications do not call this method directly.

**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:**target - the container in which to do the layout. **Returns:**the minimum dimensions needed to lay out the subcomponents of the specified container.**See Also:**[Container](http://docs.google.com/java/awt/Container.html), [preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/BorderLayout.html#preferredLayoutSize(java.awt.Container)), [Container.getMinimumSize()](http://docs.google.com/java/awt/Container.html#getMinimumSize())

### preferredLayoutSize

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

Determines the preferred size of the target container using this layout manager, based on the components in the container.

Most applications do not call this method directly. This method is called when a container calls its getPreferredSize method.

**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:**target - the container in which to do the layout. **Returns:**the preferred dimensions to lay out the subcomponents of the specified container.**See Also:**[Container](http://docs.google.com/java/awt/Container.html), [minimumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/BorderLayout.html#minimumLayoutSize(java.awt.Container)), [Container.getPreferredSize()](http://docs.google.com/java/awt/Container.html#getPreferredSize())

### 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 dimensions for this layout given the components in the specified target container.

**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) **Parameters:**target - the component which needs to be laid out**See Also:**[Container](http://docs.google.com/java/awt/Container.html), [minimumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/BorderLayout.html#minimumLayoutSize(java.awt.Container)), [preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/BorderLayout.html#preferredLayoutSize(java.awt.Container))

### getLayoutAlignmentX

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

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) parent)

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) target)

Invalidates the layout, indicating that if the layout manager has cached information it should be discarded.

**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)

### layoutContainer

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

Lays out the container argument using this border layout.

This method actually reshapes the components in the specified container in order to satisfy the constraints of this BorderLayout object. The NORTH and SOUTH components, if any, are placed at the top and bottom of the container, respectively. The WEST and EAST components are then placed on the left and right, respectively. Finally, the CENTER object is placed in any remaining space in the middle.

Most applications do not call this method directly. This method is called when a container calls its doLayout method.

**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:**target - the container in which to do the layout.**See Also:**[Container](http://docs.google.com/java/awt/Container.html), [Container.doLayout()](http://docs.google.com/java/awt/Container.html#doLayout())

### toString

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

Returns a string representation of the state of this border layout.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of this border layout.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BorderLayout.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/java/awt/BasicStroke.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/BufferCapabilities.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/BorderLayout.html)    [**NO FRAMES**](http://docs.google.com/BorderLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#4i7ojhp) | [METHOD](#3whwml4) |

[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).