| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CardLayout.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/Canvas.AccessibleAWTCanvas.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Checkbox.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/CardLayout.html)    [**NO FRAMES**](http://docs.google.com/CardLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **java.awt**

Class CardLayout

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

**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 **CardLayout**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 CardLayout object is a layout manager for a container. It treats each component in the container as a card. Only one card is visible at a time, and the container acts as a stack of cards. The first component added to a CardLayout object is the visible component when the container is first displayed.

The ordering of cards is determined by the container's own internal ordering of its component objects. CardLayout defines a set of methods that allow an application to flip through these cards sequentially, or to show a specified card. The [addLayoutComponent(java.awt.Component, java.lang.Object)](http://docs.google.com/java/awt/CardLayout.html#addLayoutComponent(java.awt.Component,%20java.lang.Object)) method can be used to associate a string identifier with a given card for fast random access.

**Since:** JDK1.0 **See Also:**[Container](http://docs.google.com/java/awt/Container.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.CardLayout)

| **Constructor Summary** | |
| --- | --- |
| [**CardLayout**](http://docs.google.com/java/awt/CardLayout.html#CardLayout())()            Creates a new card layout with gaps of size zero. |
| [**CardLayout**](http://docs.google.com/java/awt/CardLayout.html#CardLayout(int,%20int))(int hgap, int vgap)            Creates a new card layout with the specified horizontal and vertical gaps. |

| **Method Summary** | |
| --- | --- |
| void | [**addLayoutComponent**](http://docs.google.com/java/awt/CardLayout.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 this card layout's internal table of names. |
| void | [**addLayoutComponent**](http://docs.google.com/java/awt/CardLayout.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).* |
| void | [**first**](http://docs.google.com/java/awt/CardLayout.html#first(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Flips to the first card of the container. |
| int | [**getHgap**](http://docs.google.com/java/awt/CardLayout.html#getHgap())()            Gets the horizontal gap between components. |
| float | [**getLayoutAlignmentX**](http://docs.google.com/java/awt/CardLayout.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/CardLayout.html#getLayoutAlignmentY(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Returns the alignment along the y axis. |
| int | [**getVgap**](http://docs.google.com/java/awt/CardLayout.html#getVgap())()            Gets the vertical gap between components. |
| void | [**invalidateLayout**](http://docs.google.com/java/awt/CardLayout.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 | [**last**](http://docs.google.com/java/awt/CardLayout.html#last(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Flips to the last card of the container. |
| void | [**layoutContainer**](http://docs.google.com/java/awt/CardLayout.html#layoutContainer(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Lays out the specified container using this card layout. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**maximumLayoutSize**](http://docs.google.com/java/awt/CardLayout.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/CardLayout.html#minimumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Calculates the minimum size for the specified panel. |
| void | [**next**](http://docs.google.com/java/awt/CardLayout.html#next(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Flips to the next card of the specified container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredLayoutSize**](http://docs.google.com/java/awt/CardLayout.html#preferredLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Determines the preferred size of the container argument using this card layout. |
| void | [**previous**](http://docs.google.com/java/awt/CardLayout.html#previous(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Flips to the previous card of the specified container. |
| void | [**removeLayoutComponent**](http://docs.google.com/java/awt/CardLayout.html#removeLayoutComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Removes the specified component from the layout. |
| void | [**setHgap**](http://docs.google.com/java/awt/CardLayout.html#setHgap(int))(int hgap)            Sets the horizontal gap between components. |
| void | [**setVgap**](http://docs.google.com/java/awt/CardLayout.html#setVgap(int))(int vgap)            Sets the vertical gap between components. |
| void | [**show**](http://docs.google.com/java/awt/CardLayout.html#show(java.awt.Container,%20java.lang.String))([Container](http://docs.google.com/java/awt/Container.html) parent, [String](http://docs.google.com/java/lang/String.html) name)            Flips to the component that was added to this layout with the specified name, using addLayoutComponent. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/awt/CardLayout.html#toString())()            Returns a string representation of the state of this card 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)) |

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

### CardLayout

public **CardLayout**()

Creates a new card layout with gaps of size zero.

### CardLayout

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

Creates a new card layout with the specified horizontal and vertical gaps. The horizontal gaps are placed at the left and right edges. The vertical gaps are placed at the top and bottom edges.

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

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

### getHgap

public int **getHgap**()

Gets the horizontal gap between components.

**Returns:**the horizontal gap between components.**Since:** JDK1.1 **See Also:**[setHgap(int)](http://docs.google.com/java/awt/CardLayout.html#setHgap(int)), [getVgap()](http://docs.google.com/java/awt/CardLayout.html#getVgap())

### setHgap

public void **setHgap**(int hgap)

Sets the horizontal gap between components.

**Parameters:**hgap - the horizontal gap between components.**Since:** JDK1.1 **See Also:**[getHgap()](http://docs.google.com/java/awt/CardLayout.html#getHgap()), [setVgap(int)](http://docs.google.com/java/awt/CardLayout.html#setVgap(int))

### getVgap

public int **getVgap**()

Gets the vertical gap between components.

**Returns:**the vertical gap between components.**See Also:**[setVgap(int)](http://docs.google.com/java/awt/CardLayout.html#setVgap(int)), [getHgap()](http://docs.google.com/java/awt/CardLayout.html#getHgap())

### setVgap

public void **setVgap**(int vgap)

Sets the vertical gap between components.

**Parameters:**vgap - the vertical gap between components.**Since:** JDK1.1 **See Also:**[getVgap()](http://docs.google.com/java/awt/CardLayout.html#getVgap()), [setHgap(int)](http://docs.google.com/java/awt/CardLayout.html#setHgap(int))

### 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 this card layout's internal table of names. The object specified by constraints must be a string. The card layout stores this string as a key-value pair that can be used for random access to a particular card. By calling the show method, an application can display the component with the specified name.

**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 - a tag that identifies a particular card in the layout. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the constraint is not a string.**See Also:**[show(java.awt.Container, java.lang.String)](http://docs.google.com/java/awt/CardLayout.html#show(java.awt.Container,%20java.lang.String))

### 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 the layout. If the card was visible on top, the next card underneath it is shown.

**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())

### preferredLayoutSize

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

Determines the preferred size of the container argument using this card layout.

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

### minimumLayoutSize

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

Calculates the minimum size for the specified panel.

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

### 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/CardLayout.html#minimumLayoutSize(java.awt.Container)), [preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/CardLayout.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) parent)

Lays out the specified container using this card layout.

Each component in the parent container is reshaped to be the size of the container, minus space for surrounding insets, horizontal gaps, and vertical gaps.

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

### first

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

Flips to the first card of the container.

**Parameters:**parent - the parent container in which to do the layout**See Also:**[last(java.awt.Container)](http://docs.google.com/java/awt/CardLayout.html#last(java.awt.Container))

### next

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

Flips to the next card of the specified container. If the currently visible card is the last one, this method flips to the first card in the layout.

**Parameters:**parent - the parent container in which to do the layout**See Also:**[previous(java.awt.Container)](http://docs.google.com/java/awt/CardLayout.html#previous(java.awt.Container))

### previous

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

Flips to the previous card of the specified container. If the currently visible card is the first one, this method flips to the last card in the layout.

**Parameters:**parent - the parent container in which to do the layout**See Also:**[next(java.awt.Container)](http://docs.google.com/java/awt/CardLayout.html#next(java.awt.Container))

### last

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

Flips to the last card of the container.

**Parameters:**parent - the parent container in which to do the layout**See Also:**[first(java.awt.Container)](http://docs.google.com/java/awt/CardLayout.html#first(java.awt.Container))

### show

public void **show**([Container](http://docs.google.com/java/awt/Container.html) parent,  
 [String](http://docs.google.com/java/lang/String.html) name)

Flips to the component that was added to this layout with the specified name, using addLayoutComponent. If no such component exists, then nothing happens.

**Parameters:**parent - the parent container in which to do the layoutname - the component name**See Also:**[addLayoutComponent(java.awt.Component, java.lang.Object)](http://docs.google.com/java/awt/CardLayout.html#addLayoutComponent(java.awt.Component,%20java.lang.Object))

### toString

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

Returns a string representation of the state of this card 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 card 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/CardLayout.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/Canvas.AccessibleAWTCanvas.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Checkbox.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/CardLayout.html)    [**NO FRAMES**](http://docs.google.com/CardLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [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).