| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/OverlayLayout.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/MutableComboBoxModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/Popup.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/OverlayLayout.html)    [**NO FRAMES**](http://docs.google.com/OverlayLayout.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**

Class OverlayLayout

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.OverlayLayout**

**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 **OverlayLayout**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 layout manager to arrange components over the top of each other. The requested size of the container will be the largest requested size of the children, taking alignment needs into consideration. The alignment is based upon what is needed to properly fit the children in the allocation area. The children will be placed such that their alignment points are all on top of each other.

**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).

| **Constructor Summary** | |
| --- | --- |
| [**OverlayLayout**](http://docs.google.com/javax/swing/OverlayLayout.html#OverlayLayout(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Constructs a layout manager that performs overlay arrangement of the children. |

| **Method Summary** | |
| --- | --- |
| void | [**addLayoutComponent**](http://docs.google.com/javax/swing/OverlayLayout.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/OverlayLayout.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)            Adds the specified component to the layout. |
| float | [**getLayoutAlignmentX**](http://docs.google.com/javax/swing/OverlayLayout.html#getLayoutAlignmentX(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the alignment along the x axis for the container. |
| float | [**getLayoutAlignmentY**](http://docs.google.com/javax/swing/OverlayLayout.html#getLayoutAlignmentY(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the alignment along the y axis for the container. |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getTarget**](http://docs.google.com/javax/swing/OverlayLayout.html#getTarget())()            Returns the container that uses this layout manager. |
| void | [**invalidateLayout**](http://docs.google.com/javax/swing/OverlayLayout.html#invalidateLayout(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Indicates a child has changed its layout related information, which causes any cached calculations to be flushed. |
| void | [**layoutContainer**](http://docs.google.com/javax/swing/OverlayLayout.html#layoutContainer(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Called by the AWT when the specified container needs to be laid out. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**maximumLayoutSize**](http://docs.google.com/javax/swing/OverlayLayout.html#maximumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the maximum dimensions needed to lay out the components contained in the specified target container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumLayoutSize**](http://docs.google.com/javax/swing/OverlayLayout.html#minimumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the minimum dimensions needed to lay out the components contained in the specified target container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredLayoutSize**](http://docs.google.com/javax/swing/OverlayLayout.html#preferredLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the preferred dimensions for this layout given the components in the specified target container. |
| void | [**removeLayoutComponent**](http://docs.google.com/javax/swing/OverlayLayout.html#removeLayoutComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            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** |
| --- |

### OverlayLayout

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

Constructs a layout manager that performs overlay arrangement of the children. The layout manager created is dedicated to the given container.

**Parameters:**target - the container to do layout against

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

### getTarget

public final [Container](http://docs.google.com/java/awt/Container.html) **getTarget**()

Returns the container that uses this layout manager.

**Returns:**the container that uses this layout manager**Since:** 1.6

### invalidateLayout

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

Indicates a child has changed its layout related information, which causes any cached calculations to be flushed.

**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) **Parameters:**target - the container

### addLayoutComponent

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

Adds the specified component to the layout. Used by this class to know when to invalidate layout.

**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 name of the componentcomp - the 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. Used by this class to know when to invalidate 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:**comp - the component to remove

### 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. Used by this class to know when to invalidate layout.

**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.

### preferredLayoutSize

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

Returns the preferred dimensions for this layout given the components in the specified target container. Recomputes the layout if it has been invalidated. Factors in the current inset setting returned by getInsets().

**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 component which needs to be laid out **Returns:**a Dimension object containing the preferred dimensions**See Also:**[minimumLayoutSize(java.awt.Container)](http://docs.google.com/javax/swing/OverlayLayout.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) target)

Returns the minimum dimensions needed to lay out the components contained in the specified target container. Recomputes the layout if it has been invalidated, and factors in the current inset setting.

**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 component which needs to be laid out **Returns:**a Dimension object containing the minimum dimensions**See Also:**[preferredLayoutSize(java.awt.Container)](http://docs.google.com/javax/swing/OverlayLayout.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 needed to lay out the components contained in the specified target container. Recomputes the layout if it has been invalidated, and factors in the inset setting returned by getInset.

**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 that needs to be laid out **Returns:**a Dimension object containing the maximum dimensions**See Also:**[preferredLayoutSize(java.awt.Container)](http://docs.google.com/javax/swing/OverlayLayout.html#preferredLayoutSize(java.awt.Container))

### getLayoutAlignmentX

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

Returns the alignment along the x axis for the container.

**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) **Parameters:**target - the container **Returns:**the alignment >= 0.0f && <= 1.0f

### getLayoutAlignmentY

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

Returns the alignment along the y axis for the container.

**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) **Parameters:**target - the container **Returns:**the alignment >= 0.0f && <= 1.0f

### layoutContainer

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

Called by the AWT when the specified container needs to be laid out.

**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 to lay out **Throws:** [AWTError](http://docs.google.com/java/awt/AWTError.html) - if the target isn't the container specified to the constructor

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/OverlayLayout.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/MutableComboBoxModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/Popup.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/OverlayLayout.html)    [**NO FRAMES**](http://docs.google.com/OverlayLayout.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).