| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LayoutStyle.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/LayoutFocusTraversalPolicy.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/LayoutStyle.ComponentPlacement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/LayoutStyle.html)    [**NO FRAMES**](http://docs.google.com/LayoutStyle.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **javax.swing**

Class LayoutStyle

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

public abstract class **LayoutStyle**extends [Object](http://docs.google.com/java/lang/Object.html)

LayoutStyle provides information about how to position components. This class is primarily useful for visual tools and layout managers. Most developers will not need to use this class.

You typically don't set or create a LayoutStyle. Instead use the static method getInstance to obtain the current instance.

**Since:** 1.6

| **Nested Class Summary** | |
| --- | --- |
| static class | [**LayoutStyle.ComponentPlacement**](http://docs.google.com/javax/swing/LayoutStyle.ComponentPlacement.html)            ComponentPlacement is an enumeration of the possible ways two components can be placed relative to each other. |

| **Constructor Summary** | |
| --- | --- |
| [**LayoutStyle**](http://docs.google.com/javax/swing/LayoutStyle.html#LayoutStyle())()            Creates a new LayoutStyle. |

| **Method Summary** | |
| --- | --- |
| abstract  int | [**getContainerGap**](http://docs.google.com/javax/swing/LayoutStyle.html#getContainerGap(javax.swing.JComponent,%20int,%20java.awt.Container))([JComponent](http://docs.google.com/javax/swing/JComponent.html) component, int position, [Container](http://docs.google.com/java/awt/Container.html) parent)            Returns the amount of space to place between the component and specified edge of its parent. |
| static [LayoutStyle](http://docs.google.com/javax/swing/LayoutStyle.html) | [**getInstance**](http://docs.google.com/javax/swing/LayoutStyle.html#getInstance())()            Returns the shared instance of LayoutStyle. |
| abstract  int | [**getPreferredGap**](http://docs.google.com/javax/swing/LayoutStyle.html#getPreferredGap(javax.swing.JComponent,%20javax.swing.JComponent,%20javax.swing.LayoutStyle.ComponentPlacement,%20int,%20java.awt.Container))([JComponent](http://docs.google.com/javax/swing/JComponent.html) component1, [JComponent](http://docs.google.com/javax/swing/JComponent.html) component2, [LayoutStyle.ComponentPlacement](http://docs.google.com/javax/swing/LayoutStyle.ComponentPlacement.html) type, int position, [Container](http://docs.google.com/java/awt/Container.html) parent)            Returns the amount of space to use between two components. |
| static void | [**setInstance**](http://docs.google.com/javax/swing/LayoutStyle.html#setInstance(javax.swing.LayoutStyle))([LayoutStyle](http://docs.google.com/javax/swing/LayoutStyle.html) style)            Sets the shared instance of LayoutStyle. |

| **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** |
| --- |

### LayoutStyle

public **LayoutStyle**()

Creates a new LayoutStyle. You generally don't create a LayoutStyle. Instead use the method getInstance to obtain the current LayoutStyle.

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

### setInstance

public static void **setInstance**([LayoutStyle](http://docs.google.com/javax/swing/LayoutStyle.html) style)

Sets the shared instance of LayoutStyle. Specifying null results in using the LayoutStyle from the current LookAndFeel.

**Parameters:**style - the LayoutStyle, or null**See Also:**[getInstance()](http://docs.google.com/javax/swing/LayoutStyle.html#getInstance())

### getInstance

public static [LayoutStyle](http://docs.google.com/javax/swing/LayoutStyle.html) **getInstance**()

Returns the shared instance of LayoutStyle. If an instance has not been specified in setInstance, this will return the LayoutStyle from the current LookAndFeel.

**Returns:**the shared instance of LayoutStyle**See Also:**[LookAndFeel.getLayoutStyle()](http://docs.google.com/javax/swing/LookAndFeel.html#getLayoutStyle())

### getPreferredGap

public abstract int **getPreferredGap**([JComponent](http://docs.google.com/javax/swing/JComponent.html) component1,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) component2,  
 [LayoutStyle.ComponentPlacement](http://docs.google.com/javax/swing/LayoutStyle.ComponentPlacement.html) type,  
 int position,  
 [Container](http://docs.google.com/java/awt/Container.html) parent)

Returns the amount of space to use between two components. The return value indicates the distance to place component2 relative to component1. For example, the following returns the amount of space to place between component2 and component1 when component2 is placed vertically above component1:

int gap = getPreferredGap(component1, component2,  
 ComponentPlacement.RELATED,  
 SwingConstants.NORTH, parent);

The type parameter indicates the relation between the two components. If the two components will be contained in the same parent and are showing similar logically related items, use RELATED. If the two components will be contained in the same parent but show logically unrelated items use UNRELATED. Some look and feels may not distinguish between the RELATED and UNRELATED types.

The return value is not intended to take into account the current size and position of component2 or component1. The return value may take into consideration various properties of the components. For example, the space may vary based on font size, or the preferred size of the component.

**Parameters:**component1 - the JComponent component2 is being placed relative tocomponent2 - the JComponent being placedposition - the position component2 is being placed relative to component1; one of SwingConstants.NORTH, SwingConstants.SOUTH, SwingConstants.EAST or SwingConstants.WESTtype - how the two components are being placedparent - the parent of component2; this may differ from the actual parent and it may be null **Returns:**the amount of space to place between the two components **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if component1, component2 or type is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if position is not one of SwingConstants.NORTH, SwingConstants.SOUTH, SwingConstants.EAST or SwingConstants.WEST**Since:** 1.6 **See Also:**[LookAndFeel.getLayoutStyle()](http://docs.google.com/javax/swing/LookAndFeel.html#getLayoutStyle())

### getContainerGap

public abstract int **getContainerGap**([JComponent](http://docs.google.com/javax/swing/JComponent.html) component,  
 int position,  
 [Container](http://docs.google.com/java/awt/Container.html) parent)

Returns the amount of space to place between the component and specified edge of its parent.

**Parameters:**component - the JComponent being positionedposition - the position component is being placed relative to its parent; one of SwingConstants.NORTH, SwingConstants.SOUTH, SwingConstants.EAST or SwingConstants.WESTparent - the parent of component; this may differ from the actual parent and may be null **Returns:**the amount of space to place between the component and specified edge **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if position is not one of SwingConstants.NORTH, SwingConstants.SOUTH, SwingConstants.EAST or SwingConstants.WEST

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LayoutStyle.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/LayoutFocusTraversalPolicy.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/LayoutStyle.ComponentPlacement.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/LayoutStyle.html)    [**NO FRAMES**](http://docs.google.com/LayoutStyle.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [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).