| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SpringLayout.Constraints.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/SpringLayout.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/SwingConstants.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/SpringLayout.Constraints.html)    [**NO FRAMES**](http://docs.google.com/SpringLayout.Constraints.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

## **javax.swing**

Class SpringLayout.Constraints

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

**Enclosing class:**[SpringLayout](http://docs.google.com/javax/swing/SpringLayout.html)

public static class **SpringLayout.Constraints**extends [Object](http://docs.google.com/java/lang/Object.html)

A Constraints object holds the constraints that govern the way a component's size and position change in a container controlled by a SpringLayout. A Constraints object is like a Rectangle, in that it has x, y, width, and height properties. In the Constraints object, however, these properties have Spring values instead of integers. In addition, a Constraints object can be manipulated as four edges -- north, south, east, and west -- using the constraint property.

The following formulas are always true for a Constraints object (here WEST and x are synonyms, as are and NORTH and y):

EAST = WEST + WIDTH  
 SOUTH = NORTH + HEIGHT  
 HORIZONTAL\_CENTER = WEST + WIDTH/2  
 VERTICAL\_CENTER = NORTH + HEIGHT/2  
 ABSOLUTE\_BASELINE = NORTH + RELATIVE\_BASELINE\*

For example, if you have specified the WIDTH and WEST (X) location the EAST is calculated as WEST + WIDTH. If you instead specified the WIDTH and EAST locations the WEST (X) location is then calculated as EAST - WIDTH.

[RELATIVE\_BASELINE is a private constraint that is set automatically when the SpringLayout.Constraints(Component) constuctor is called or when a constraints object is registered with a SpringLayout object.]

**Note**: In this document, operators represent methods in the Spring class. For example, "a + b" is equal to Spring.sum(a, b), and "a - b" is equal to Spring.sum(a, Spring.minus(b)). See the [Spring API documentation](http://docs.google.com/javax/swing/Spring.html) for further details of spring arithmetic.

Because a Constraints object's properties -- representing its edges, size, and location -- can all be set independently and yet are interrelated, a Constraints object can become *over-constrained*. For example, if the WEST, WIDTH and EAST edges are all set, steps must be taken to ensure that the first of the formulas above holds. To do this, the Constraints object throws away the *least recently set* constraint so as to make the formulas hold.

**Since:** 1.4

| **Constructor Summary** | |
| --- | --- |
| [**SpringLayout.Constraints**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#SpringLayout.Constraints())()            Creates an empty Constraints object. |
| [**SpringLayout.Constraints**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#SpringLayout.Constraints(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Creates a Constraints object with suitable x, y, width and height springs for component, c. |
| [**SpringLayout.Constraints**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#SpringLayout.Constraints(javax.swing.Spring,%20javax.swing.Spring))([Spring](http://docs.google.com/javax/swing/Spring.html) x, [Spring](http://docs.google.com/javax/swing/Spring.html) y)            Creates a Constraints object with the specified values for its x and y properties. |
| [**SpringLayout.Constraints**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#SpringLayout.Constraints(javax.swing.Spring,%20javax.swing.Spring,%20javax.swing.Spring,%20javax.swing.Spring))([Spring](http://docs.google.com/javax/swing/Spring.html) x, [Spring](http://docs.google.com/javax/swing/Spring.html) y, [Spring](http://docs.google.com/javax/swing/Spring.html) width, [Spring](http://docs.google.com/javax/swing/Spring.html) height)            Creates a Constraints object with the specified values for its x, y, width, and height properties. |

| **Method Summary** | |
| --- | --- |
| [Spring](http://docs.google.com/javax/swing/Spring.html) | [**getConstraint**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getConstraint(java.lang.String))([String](http://docs.google.com/java/lang/String.html) edgeName)            Returns the value of the specified edge, which may be a derived value, or even null. |
| [Spring](http://docs.google.com/javax/swing/Spring.html) | [**getHeight**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getHeight())()            Returns the value of the height property. |
| [Spring](http://docs.google.com/javax/swing/Spring.html) | [**getWidth**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getWidth())()            Returns the value of the width property. |
| [Spring](http://docs.google.com/javax/swing/Spring.html) | [**getX**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getX())()            Returns the value of the x property. |
| [Spring](http://docs.google.com/javax/swing/Spring.html) | [**getY**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getY())()            Returns the value of the y property. |
| void | [**setConstraint**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setConstraint(java.lang.String,%20javax.swing.Spring))([String](http://docs.google.com/java/lang/String.html) edgeName, [Spring](http://docs.google.com/javax/swing/Spring.html) s)            Sets the spring controlling the specified edge. |
| void | [**setHeight**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setHeight(javax.swing.Spring))([Spring](http://docs.google.com/javax/swing/Spring.html) height)            Sets the height property, which controls the height of a component. |
| void | [**setWidth**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setWidth(javax.swing.Spring))([Spring](http://docs.google.com/javax/swing/Spring.html) width)            Sets the width property, which controls the width of a component. |
| void | [**setX**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setX(javax.swing.Spring))([Spring](http://docs.google.com/javax/swing/Spring.html) x)            Sets the x property, which controls the x value of a component's location. |
| void | [**setY**](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setY(javax.swing.Spring))([Spring](http://docs.google.com/javax/swing/Spring.html) y)            Sets the y property, which controls the y value of a component's location. |

| **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** |
| --- |

### SpringLayout.Constraints

public **SpringLayout.Constraints**()

Creates an empty Constraints object.

### SpringLayout.Constraints

public **SpringLayout.Constraints**([Spring](http://docs.google.com/javax/swing/Spring.html) x,  
 [Spring](http://docs.google.com/javax/swing/Spring.html) y)

Creates a Constraints object with the specified values for its x and y properties. The height and width springs have null values.

**Parameters:**x - the spring controlling the component's *x* valuey - the spring controlling the component's *y* value

### SpringLayout.Constraints

public **SpringLayout.Constraints**([Spring](http://docs.google.com/javax/swing/Spring.html) x,  
 [Spring](http://docs.google.com/javax/swing/Spring.html) y,  
 [Spring](http://docs.google.com/javax/swing/Spring.html) width,  
 [Spring](http://docs.google.com/javax/swing/Spring.html) height)

Creates a Constraints object with the specified values for its x, y, width, and height properties. Note: If the SpringLayout class encounters null values in the Constraints object of a given component, it replaces them with suitable defaults.

**Parameters:**x - the spring value for the x propertyy - the spring value for the y propertywidth - the spring value for the width propertyheight - the spring value for the height property

### SpringLayout.Constraints

public **SpringLayout.Constraints**([Component](http://docs.google.com/java/awt/Component.html) c)

Creates a Constraints object with suitable x, y, width and height springs for component, c. The x and y springs are constant springs initialised with the component's location at the time this method is called. The width and height springs are special springs, created by the Spring.width() and Spring.height() methods, which track the size characteristics of the component when they change.

**Parameters:**c - the component whose characteristics will be reflected by this Constraints object **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if c is null.**Since:** 1.5

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

### setX

public void **setX**([Spring](http://docs.google.com/javax/swing/Spring.html) x)

Sets the x property, which controls the x value of a component's location.

**Parameters:**x - the spring controlling the x value of a component's location**See Also:**[getX()](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getX()), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### getX

public [Spring](http://docs.google.com/javax/swing/Spring.html) **getX**()

Returns the value of the x property.

**Returns:**the spring controlling the x value of a component's location**See Also:**[setX(javax.swing.Spring)](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setX(javax.swing.Spring)), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### setY

public void **setY**([Spring](http://docs.google.com/javax/swing/Spring.html) y)

Sets the y property, which controls the y value of a component's location.

**Parameters:**y - the spring controlling the y value of a component's location**See Also:**[getY()](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getY()), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### getY

public [Spring](http://docs.google.com/javax/swing/Spring.html) **getY**()

Returns the value of the y property.

**Returns:**the spring controlling the y value of a component's location**See Also:**[setY(javax.swing.Spring)](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setY(javax.swing.Spring)), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### setWidth

public void **setWidth**([Spring](http://docs.google.com/javax/swing/Spring.html) width)

Sets the width property, which controls the width of a component.

**Parameters:**width - the spring controlling the width of this Constraints object**See Also:**[getWidth()](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getWidth()), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### getWidth

public [Spring](http://docs.google.com/javax/swing/Spring.html) **getWidth**()

Returns the value of the width property.

**Returns:**the spring controlling the width of a component**See Also:**[setWidth(javax.swing.Spring)](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setWidth(javax.swing.Spring)), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### setHeight

public void **setHeight**([Spring](http://docs.google.com/javax/swing/Spring.html) height)

Sets the height property, which controls the height of a component.

**Parameters:**height - the spring controlling the height of this Constraints object**See Also:**[getHeight()](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getHeight()), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### getHeight

public [Spring](http://docs.google.com/javax/swing/Spring.html) **getHeight**()

Returns the value of the height property.

**Returns:**the spring controlling the height of a component**See Also:**[setHeight(javax.swing.Spring)](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setHeight(javax.swing.Spring)), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### setConstraint

public void **setConstraint**([String](http://docs.google.com/java/lang/String.html) edgeName,  
 [Spring](http://docs.google.com/javax/swing/Spring.html) s)

Sets the spring controlling the specified edge. The edge must have one of the following values: SpringLayout.NORTH, SpringLayout.SOUTH, SpringLayout.EAST, SpringLayout.WEST, SpringLayout.HORIZONTAL\_CENTER, SpringLayout.VERTICAL\_CENTER, SpringLayout.BASELINE, SpringLayout.WIDTH or SpringLayout.HEIGHT. For any other String value passed as the edge, no action is taken. For a null edge, a NullPointerException is thrown.

**Parameters:**edgeName - the edge to be sets - the spring controlling the specified edge **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if edgeName is null**See Also:**[getConstraint(java.lang.String)](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#getConstraint(java.lang.String)), [SpringLayout.NORTH](http://docs.google.com/javax/swing/SpringLayout.html#NORTH), [SpringLayout.SOUTH](http://docs.google.com/javax/swing/SpringLayout.html#SOUTH), [SpringLayout.EAST](http://docs.google.com/javax/swing/SpringLayout.html#EAST), [SpringLayout.WEST](http://docs.google.com/javax/swing/SpringLayout.html#WEST), [SpringLayout.HORIZONTAL\_CENTER](http://docs.google.com/javax/swing/SpringLayout.html#HORIZONTAL_CENTER), [SpringLayout.VERTICAL\_CENTER](http://docs.google.com/javax/swing/SpringLayout.html#VERTICAL_CENTER), [SpringLayout.BASELINE](http://docs.google.com/javax/swing/SpringLayout.html#BASELINE), [SpringLayout.WIDTH](http://docs.google.com/javax/swing/SpringLayout.html#WIDTH), [SpringLayout.HEIGHT](http://docs.google.com/javax/swing/SpringLayout.html#HEIGHT), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

### getConstraint

public [Spring](http://docs.google.com/javax/swing/Spring.html) **getConstraint**([String](http://docs.google.com/java/lang/String.html) edgeName)

Returns the value of the specified edge, which may be a derived value, or even null. The edge must have one of the following values: SpringLayout.NORTH, SpringLayout.SOUTH, SpringLayout.EAST, SpringLayout.WEST, SpringLayout.HORIZONTAL\_CENTER, SpringLayout.VERTICAL\_CENTER, SpringLayout.BASELINE, SpringLayout.WIDTH or SpringLayout.HEIGHT. For any other String value passed as the edge, null will be returned. Throws NullPointerException for a null edge.

**Parameters:**edgeName - the edge whose value is to be returned **Returns:**the spring controlling the specified edge, may be null **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if edgeName is null**See Also:**[setConstraint(java.lang.String, javax.swing.Spring)](http://docs.google.com/javax/swing/SpringLayout.Constraints.html#setConstraint(java.lang.String,%20javax.swing.Spring)), [SpringLayout.NORTH](http://docs.google.com/javax/swing/SpringLayout.html#NORTH), [SpringLayout.SOUTH](http://docs.google.com/javax/swing/SpringLayout.html#SOUTH), [SpringLayout.EAST](http://docs.google.com/javax/swing/SpringLayout.html#EAST), [SpringLayout.WEST](http://docs.google.com/javax/swing/SpringLayout.html#WEST), [SpringLayout.HORIZONTAL\_CENTER](http://docs.google.com/javax/swing/SpringLayout.html#HORIZONTAL_CENTER), [SpringLayout.VERTICAL\_CENTER](http://docs.google.com/javax/swing/SpringLayout.html#VERTICAL_CENTER), [SpringLayout.BASELINE](http://docs.google.com/javax/swing/SpringLayout.html#BASELINE), [SpringLayout.WIDTH](http://docs.google.com/javax/swing/SpringLayout.html#WIDTH), [SpringLayout.HEIGHT](http://docs.google.com/javax/swing/SpringLayout.html#HEIGHT), [SpringLayout.Constraints](http://docs.google.com/javax/swing/SpringLayout.Constraints.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SpringLayout.Constraints.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/SpringLayout.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/SwingConstants.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/SpringLayout.Constraints.html)    [**NO FRAMES**](http://docs.google.com/SpringLayout.Constraints.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

[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).