| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SizeRequirements.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/SingleSelectionModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/SizeSequence.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/SizeRequirements.html)    [**NO FRAMES**](http://docs.google.com/SizeRequirements.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#1ksv4uv) |

## **javax.swing**

Class SizeRequirements

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

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

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

For the convenience of layout managers, calculates information about the size and position of components. All size and position calculation methods are class methods that take arrays of SizeRequirements as arguments. The SizeRequirements class supports two types of layout:

tiled The components are placed end-to-end, starting either at coordinate 0 (the leftmost or topmost position) or at the coordinate representing the end of the allocated span (the rightmost or bottommost position). aligned The components are aligned as specified by each component's X or Y alignment value.

Each SizeRequirements object contains information about either the width (and X alignment) or height (and Y alignment) of a single component or a group of components:

minimum The smallest reasonable width/height of the component or component group, in pixels. preferred The natural width/height of the component or component group, in pixels. maximum The largest reasonable width/height of the component or component group, in pixels. alignment The X/Y alignment of the component or component group.

**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).

**See Also:**[Component.getMinimumSize()](http://docs.google.com/java/awt/Component.html#getMinimumSize()), [Component.getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize()), [Component.getMaximumSize()](http://docs.google.com/java/awt/Component.html#getMaximumSize()), [Component.getAlignmentX()](http://docs.google.com/java/awt/Component.html#getAlignmentX()), [Component.getAlignmentY()](http://docs.google.com/java/awt/Component.html#getAlignmentY())

| **Field Summary** | |
| --- | --- |
| float | [**alignment**](http://docs.google.com/javax/swing/SizeRequirements.html#alignment)            The alignment, specified as a value between 0.0 and 1.0, inclusive. |
| int | [**maximum**](http://docs.google.com/javax/swing/SizeRequirements.html#maximum)            The maximum size allowed. |
| int | [**minimum**](http://docs.google.com/javax/swing/SizeRequirements.html#minimum)            The minimum size required. |
| int | [**preferred**](http://docs.google.com/javax/swing/SizeRequirements.html#preferred)            The preferred (natural) size. |

| **Constructor Summary** | |
| --- | --- |
| [**SizeRequirements**](http://docs.google.com/javax/swing/SizeRequirements.html#SizeRequirements())()            Creates a SizeRequirements object with the minimum, preferred, and maximum sizes set to zero and an alignment value of 0.5 (centered). |
| [**SizeRequirements**](http://docs.google.com/javax/swing/SizeRequirements.html#SizeRequirements(int,%20int,%20int,%20float))(int min, int pref, int max, float a)            Creates a SizeRequirements object with the specified minimum, preferred, and maximum sizes and the specified alignment. |

| **Method Summary** | |
| --- | --- |
| static int[] | [**adjustSizes**](http://docs.google.com/javax/swing/SizeRequirements.html#adjustSizes(int,%20javax.swing.SizeRequirements%5B%5D))(int delta, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children)            Adjust a specified array of sizes by a given amount. |
| static void | [**calculateAlignedPositions**](http://docs.google.com/javax/swing/SizeRequirements.html#calculateAlignedPositions(int,%20javax.swing.SizeRequirements,%20javax.swing.SizeRequirements%5B%5D,%20int%5B%5D,%20int%5B%5D))(int allocated, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) total, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children, int[] offsets, int[] spans)            Creates a bunch of offset/span pairs specifying how to lay out a set of components with the specified alignments. |
| static void | [**calculateAlignedPositions**](http://docs.google.com/javax/swing/SizeRequirements.html#calculateAlignedPositions(int,%20javax.swing.SizeRequirements,%20javax.swing.SizeRequirements%5B%5D,%20int%5B%5D,%20int%5B%5D,%20boolean))(int allocated, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) total, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children, int[] offsets, int[] spans, boolean normal)            Creates a set of offset/span pairs specifying how to lay out a set of components with the specified alignments. |
| static void | [**calculateTiledPositions**](http://docs.google.com/javax/swing/SizeRequirements.html#calculateTiledPositions(int,%20javax.swing.SizeRequirements,%20javax.swing.SizeRequirements%5B%5D,%20int%5B%5D,%20int%5B%5D))(int allocated, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) total, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children, int[] offsets, int[] spans)            Creates a set of offset/span pairs representing how to lay out a set of components end-to-end. |
| static void | [**calculateTiledPositions**](http://docs.google.com/javax/swing/SizeRequirements.html#calculateTiledPositions(int,%20javax.swing.SizeRequirements,%20javax.swing.SizeRequirements%5B%5D,%20int%5B%5D,%20int%5B%5D,%20boolean))(int allocated, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) total, [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children, int[] offsets, int[] spans, boolean forward)            Creates a set of offset/span pairs representing how to lay out a set of components end-to-end. |
| static [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) | [**getAlignedSizeRequirements**](http://docs.google.com/javax/swing/SizeRequirements.html#getAlignedSizeRequirements(javax.swing.SizeRequirements%5B%5D))([SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children)            Determines the total space necessary to align a set of components. |
| static [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) | [**getTiledSizeRequirements**](http://docs.google.com/javax/swing/SizeRequirements.html#getTiledSizeRequirements(javax.swing.SizeRequirements%5B%5D))([SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children)            Determines the total space necessary to place a set of components end-to-end. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/swing/SizeRequirements.html#toString())()            Returns a string describing the minimum, preferred, and maximum size requirements, along with the alignment. |

| **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** |
| --- |

### minimum

public int **minimum**

The minimum size required. For a component comp, this should be equal to either comp.getMinimumSize().width or comp.getMinimumSize().height.

### preferred

public int **preferred**

The preferred (natural) size. For a component comp, this should be equal to either comp.getPreferredSize().width or comp.getPreferredSize().height.

### maximum

public int **maximum**

The maximum size allowed. For a component comp, this should be equal to either comp.getMaximumSize().width or comp.getMaximumSize().height.

### alignment

public float **alignment**

The alignment, specified as a value between 0.0 and 1.0, inclusive. To specify centering, the alignment should be 0.5.

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

### SizeRequirements

public **SizeRequirements**()

Creates a SizeRequirements object with the minimum, preferred, and maximum sizes set to zero and an alignment value of 0.5 (centered).

### SizeRequirements

public **SizeRequirements**(int min,  
 int pref,  
 int max,  
 float a)

Creates a SizeRequirements object with the specified minimum, preferred, and maximum sizes and the specified alignment.

**Parameters:**min - the minimum size >= 0pref - the preferred size >= 0max - the maximum size >= 0a - the alignment >= 0.0f && <= 1.0f

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

### toString

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

Returns a string describing the minimum, preferred, and maximum size requirements, along with the alignment.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the string

### getTiledSizeRequirements

public static [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) **getTiledSizeRequirements**([SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children)

Determines the total space necessary to place a set of components end-to-end. The needs of each component in the set are represented by an entry in the passed-in SizeRequirements array. The returned SizeRequirements object has an alignment of 0.5 (centered). The space requirement is never more than Integer.MAX\_VALUE.

**Parameters:**children - the space requirements for a set of components. The vector may be of zero length, which will result in a default SizeRequirements object instance being passed back. **Returns:**the total space requirements.

### getAlignedSizeRequirements

public static [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) **getAlignedSizeRequirements**([SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children)

Determines the total space necessary to align a set of components. The needs of each component in the set are represented by an entry in the passed-in SizeRequirements array. The total space required will never be more than Integer.MAX\_VALUE.

**Parameters:**children - the set of child requirements. If of zero length, the returns result will be a default instance of SizeRequirements. **Returns:**the total space requirements.

### calculateTiledPositions

public static void **calculateTiledPositions**(int allocated,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) total,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children,  
 int[] offsets,  
 int[] spans)

Creates a set of offset/span pairs representing how to lay out a set of components end-to-end. This method requires that you specify the total amount of space to be allocated, the size requirements for each component to be placed (specified as an array of SizeRequirements), and the total size requirement of the set of components. You can get the total size requirement by invoking the getTiledSizeRequirements method. The components will be tiled in the forward direction with offsets increasing from 0.

**Parameters:**allocated - the total span to be allocated >= 0.total - the total of the children requests. This argument is optional and may be null.children - the size requirements for each component.offsets - the offset from 0 for each child where the spans were allocated (determines placement of the span).spans - the span allocated for each child to make the total target span.

### calculateTiledPositions

public static void **calculateTiledPositions**(int allocated,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) total,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children,  
 int[] offsets,  
 int[] spans,  
 boolean forward)

Creates a set of offset/span pairs representing how to lay out a set of components end-to-end. This method requires that you specify the total amount of space to be allocated, the size requirements for each component to be placed (specified as an array of SizeRequirements), and the total size requirement of the set of components. You can get the total size requirement by invoking the getTiledSizeRequirements method. This method also requires a flag indicating whether components should be tiled in the forward direction (offsets increasing from 0) or reverse direction (offsets decreasing from the end of the allocated space). The forward direction represents components tiled from left to right or top to bottom. The reverse direction represents components tiled from right to left or bottom to top.

**Parameters:**allocated - the total span to be allocated >= 0.total - the total of the children requests. This argument is optional and may be null.children - the size requirements for each component.offsets - the offset from 0 for each child where the spans were allocated (determines placement of the span).spans - the span allocated for each child to make the total target span.forward - tile with offsets increasing from 0 if true and with offsets decreasing from the end of the allocated space if false.**Since:** 1.4

### calculateAlignedPositions

public static void **calculateAlignedPositions**(int allocated,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) total,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children,  
 int[] offsets,  
 int[] spans)

Creates a bunch of offset/span pairs specifying how to lay out a set of components with the specified alignments. The resulting span allocations will overlap, with each one fitting as well as possible into the given total allocation. This method requires that you specify the total amount of space to be allocated, the size requirements for each component to be placed (specified as an array of SizeRequirements), and the total size requirements of the set of components (only the alignment field of which is actually used). You can get the total size requirement by invoking getAlignedSizeRequirements. Normal alignment will be done with an alignment value of 0.0f representing the left/top edge of a component.

**Parameters:**allocated - the total span to be allocated >= 0.total - the total of the children requests.children - the size requirements for each component.offsets - the offset from 0 for each child where the spans were allocated (determines placement of the span).spans - the span allocated for each child to make the total target span.

### calculateAlignedPositions

public static void **calculateAlignedPositions**(int allocated,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html) total,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children,  
 int[] offsets,  
 int[] spans,  
 boolean normal)

Creates a set of offset/span pairs specifying how to lay out a set of components with the specified alignments. The resulting span allocations will overlap, with each one fitting as well as possible into the given total allocation. This method requires that you specify the total amount of space to be allocated, the size requirements for each component to be placed (specified as an array of SizeRequirements), and the total size requirements of the set of components (only the alignment field of which is actually used) You can get the total size requirement by invoking getAlignedSizeRequirements. This method also requires a flag indicating whether normal or reverse alignment should be performed. With normal alignment the value 0.0f represents the left/top edge of the component to be aligned. With reverse alignment, 0.0f represents the right/bottom edge.

**Parameters:**allocated - the total span to be allocated >= 0.total - the total of the children requests.children - the size requirements for each component.offsets - the offset from 0 for each child where the spans were allocated (determines placement of the span).spans - the span allocated for each child to make the total target span.normal - when true, the alignment value 0.0f means left/top; when false, it means right/bottom.**Since:** 1.4

### adjustSizes

public static int[] **adjustSizes**(int delta,  
 [SizeRequirements](http://docs.google.com/javax/swing/SizeRequirements.html)[] children)

Adjust a specified array of sizes by a given amount.

**Parameters:**delta - an int specifying the size differencechildren - an array of SizeRequirements objects **Returns:**an array of ints containing the final size for each item

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SizeRequirements.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/SingleSelectionModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/SizeSequence.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/SizeRequirements.html)    [**NO FRAMES**](http://docs.google.com/SizeRequirements.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#1ksv4uv) |

[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).