| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GlyphJustificationInfo.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/font/FontRenderContext.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/font/GlyphMetrics.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/font/GlyphJustificationInfo.html)    [**NO FRAMES**](http://docs.google.com/GlyphJustificationInfo.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#4i7ojhp) | METHOD |

## **java.awt.font**

Class GlyphJustificationInfo

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

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

The GlyphJustificationInfo class represents information about the justification properties of a glyph. A glyph is the visual representation of one or more characters. Many different glyphs can be used to represent a single character or combination of characters. The four justification properties represented by GlyphJustificationInfo are weight, priority, absorb and limit.

Weight is the overall 'weight' of the glyph in the line. Generally it is proportional to the size of the font. Glyphs with larger weight are allocated a correspondingly larger amount of the change in space.

Priority determines the justification phase in which this glyph is used. All glyphs of the same priority are examined before glyphs of the next priority. If all the change in space can be allocated to these glyphs without exceeding their limits, then glyphs of the next priority are not examined. There are four priorities, kashida, whitespace, interchar, and none. KASHIDA is the first priority examined. NONE is the last priority examined.

Absorb determines whether a glyph absorbs all change in space. Within a given priority, some glyphs may absorb all the change in space. If any of these glyphs are present, no glyphs of later priority are examined.

Limit determines the maximum or minimum amount by which the glyph can change. Left and right sides of the glyph can have different limits.

Each GlyphJustificationInfo represents two sets of metrics, which are *growing* and *shrinking*. Growing metrics are used when the glyphs on a line are to be spread apart to fit a larger width. Shrinking metrics are used when the glyphs are to be moved together to fit a smaller width.

| **Field Summary** | |
| --- | --- |
| boolean | [**growAbsorb**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#growAbsorb)            If true, this glyph absorbs all extra space at this and lower priority levels when it grows. |
| float | [**growLeftLimit**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#growLeftLimit)            The maximum amount by which the left side of this glyph can grow. |
| int | [**growPriority**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#growPriority)            The priority level of this glyph as it is growing. |
| float | [**growRightLimit**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#growRightLimit)            The maximum amount by which the right side of this glyph can grow. |
| static int | [**PRIORITY\_INTERCHAR**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#PRIORITY_INTERCHAR)            The second lowest justification priority. |
| static int | [**PRIORITY\_KASHIDA**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#PRIORITY_KASHIDA)            The highest justification priority. |
| static int | [**PRIORITY\_NONE**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#PRIORITY_NONE)            The lowest justification priority. |
| static int | [**PRIORITY\_WHITESPACE**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#PRIORITY_WHITESPACE)            The second highest justification priority. |
| boolean | [**shrinkAbsorb**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#shrinkAbsorb)            If true,this glyph absorbs all remaining shrinkage at this and lower priority levels as it shrinks. |
| float | [**shrinkLeftLimit**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#shrinkLeftLimit)            The maximum amount by which the left side of this glyph can shrink (a positive number). |
| int | [**shrinkPriority**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#shrinkPriority)            The priority level of this glyph as it is shrinking. |
| float | [**shrinkRightLimit**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#shrinkRightLimit)            The maximum amount by which the right side of this glyph can shrink (a positive number). |
| float | [**weight**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#weight)            The weight of this glyph. |

| **Constructor Summary** | |
| --- | --- |
| [**GlyphJustificationInfo**](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html#GlyphJustificationInfo(float,%20boolean,%20int,%20float,%20float,%20boolean,%20int,%20float,%20float))(float weight, boolean growAbsorb, int growPriority, float growLeftLimit, float growRightLimit, boolean shrinkAbsorb, int shrinkPriority, float shrinkLeftLimit, float shrinkRightLimit)            Constructs information about the justification properties of a glyph. |

| **Method Summary** | |
| --- | --- |

| **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)) |

| **Field Detail** |
| --- |

### PRIORITY\_KASHIDA

public static final int **PRIORITY\_KASHIDA**

The highest justification priority.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphJustificationInfo.PRIORITY_KASHIDA)

### PRIORITY\_WHITESPACE

public static final int **PRIORITY\_WHITESPACE**

The second highest justification priority.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphJustificationInfo.PRIORITY_WHITESPACE)

### PRIORITY\_INTERCHAR

public static final int **PRIORITY\_INTERCHAR**

The second lowest justification priority.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphJustificationInfo.PRIORITY_INTERCHAR)

### PRIORITY\_NONE

public static final int **PRIORITY\_NONE**

The lowest justification priority.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphJustificationInfo.PRIORITY_NONE)

### weight

public final float **weight**

The weight of this glyph.

### growPriority

public final int **growPriority**

The priority level of this glyph as it is growing.

### growAbsorb

public final boolean **growAbsorb**

If true, this glyph absorbs all extra space at this and lower priority levels when it grows.

### growLeftLimit

public final float **growLeftLimit**

The maximum amount by which the left side of this glyph can grow.

### growRightLimit

public final float **growRightLimit**

The maximum amount by which the right side of this glyph can grow.

### shrinkPriority

public final int **shrinkPriority**

The priority level of this glyph as it is shrinking.

### shrinkAbsorb

public final boolean **shrinkAbsorb**

If true,this glyph absorbs all remaining shrinkage at this and lower priority levels as it shrinks.

### shrinkLeftLimit

public final float **shrinkLeftLimit**

The maximum amount by which the left side of this glyph can shrink (a positive number).

### shrinkRightLimit

public final float **shrinkRightLimit**

The maximum amount by which the right side of this glyph can shrink (a positive number).

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

### GlyphJustificationInfo

public **GlyphJustificationInfo**(float weight,  
 boolean growAbsorb,  
 int growPriority,  
 float growLeftLimit,  
 float growRightLimit,  
 boolean shrinkAbsorb,  
 int shrinkPriority,  
 float shrinkLeftLimit,  
 float shrinkRightLimit)

Constructs information about the justification properties of a glyph.

**Parameters:**weight - the weight of this glyph when allocating space. Must be non-negative.growAbsorb - if true this glyph absorbs all extra space at this priority and lower priority levels when it growsgrowPriority - the priority level of this glyph when it growsgrowLeftLimit - the maximum amount by which the left side of this glyph can grow. Must be non-negative.growRightLimit - the maximum amount by which the right side of this glyph can grow. Must be non-negative.shrinkAbsorb - if true, this glyph absorbs all remaining shrinkage at this and lower priority levels when it shrinksshrinkPriority - the priority level of this glyph when it shrinksshrinkLeftLimit - the maximum amount by which the left side of this glyph can shrink. Must be non-negative.shrinkRightLimit - the maximum amount by which the right side of this glyph can shrink. Must be non-negative.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GlyphJustificationInfo.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/font/FontRenderContext.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/font/GlyphMetrics.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/font/GlyphJustificationInfo.html)    [**NO FRAMES**](http://docs.google.com/GlyphJustificationInfo.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#4i7ojhp) | METHOD |

[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).