| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GlyphVector.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/GlyphMetrics.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/font/GraphicAttribute.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/font/GlyphVector.html)    [**NO FRAMES**](http://docs.google.com/GlyphVector.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

## **java.awt.font**

Class GlyphVector

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

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

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

A GlyphVector object is a collection of glyphs containing geometric information for the placement of each glyph in a transformed coordinate space which corresponds to the device on which the GlyphVector is ultimately displayed.

The GlyphVector does not attempt any interpretation of the sequence of glyphs it contains. Relationships between adjacent glyphs in sequence are solely used to determine the placement of the glyphs in the visual coordinate space.

Instances of GlyphVector are created by a [Font](http://docs.google.com/java/awt/Font.html).

In a text processing application that can cache intermediate representations of text, creation and subsequent caching of a GlyphVector for use during rendering is the fastest method to present the visual representation of characters to a user.

A GlyphVector is associated with exactly one Font, and can provide data useful only in relation to this Font. In addition, metrics obtained from a GlyphVector are not generally geometrically scaleable since the pixelization and spacing are dependent on grid-fitting algorithms within a Font. To facilitate accurate measurement of a GlyphVector and its component glyphs, you must specify a scaling transform, anti-alias mode, and fractional metrics mode when creating the GlyphVector. These characteristics can be derived from the destination device.

For each glyph in the GlyphVector, you can obtain:

* the position of the glyph
* the transform associated with the glyph
* the metrics of the glyph in the context of the GlyphVector. The metrics of the glyph may be different under different transforms, application specified rendering hints, and the specific instance of the glyph within the GlyphVector.

Altering the data used to create the GlyphVector does not alter the state of the GlyphVector.

Methods are provided to adjust the positions of the glyphs within the GlyphVector. These methods are most appropriate for applications that are performing justification operations for the presentation of the glyphs.

Methods are provided to transform individual glyphs within the GlyphVector. These methods are primarily useful for special effects.

Methods are provided to return both the visual, logical, and pixel bounds of the entire GlyphVector or of individual glyphs within the GlyphVector.

Methods are provided to return a [Shape](http://docs.google.com/java/awt/Shape.html) for the GlyphVector, and for individual glyphs within the GlyphVector.

**See Also:**[Font](http://docs.google.com/java/awt/Font.html), [GlyphMetrics](http://docs.google.com/java/awt/font/GlyphMetrics.html), [TextLayout](http://docs.google.com/java/awt/font/TextLayout.html)

| **Field Summary** | |
| --- | --- |
| static int | [**FLAG\_COMPLEX\_GLYPHS**](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_COMPLEX_GLYPHS)            A flag used with getLayoutFlags that indicates that this GlyphVector has a complex glyph-to-char mapping (one that does not map glyphs to chars one-to-one in strictly ascending or descending order matching the run direction). |
| static int | [**FLAG\_HAS\_POSITION\_ADJUSTMENTS**](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_HAS_POSITION_ADJUSTMENTS)            A flag used with getLayoutFlags that indicates that this GlyphVector has position adjustments. |
| static int | [**FLAG\_HAS\_TRANSFORMS**](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_HAS_TRANSFORMS)            A flag used with getLayoutFlags that indicates that this GlyphVector has per-glyph transforms. |
| static int | [**FLAG\_MASK**](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_MASK)            A mask for supported flags from getLayoutFlags. |
| static int | [**FLAG\_RUN\_RTL**](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_RUN_RTL)            A flag used with getLayoutFlags that indicates that this GlyphVector has a right-to-left run direction. |

| **Constructor Summary** | |
| --- | --- |
| [**GlyphVector**](http://docs.google.com/java/awt/font/GlyphVector.html#GlyphVector())() |

| **Method Summary** | |
| --- | --- |
| abstract  boolean | [**equals**](http://docs.google.com/java/awt/font/GlyphVector.html#equals(java.awt.font.GlyphVector))([GlyphVector](http://docs.google.com/java/awt/font/GlyphVector.html) set)            Tests if the specified GlyphVector exactly equals this GlyphVector. |
| abstract  [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/java/awt/font/GlyphVector.html#getFont())()            Returns the Font associated with this GlyphVector. |
| abstract  [FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) | [**getFontRenderContext**](http://docs.google.com/java/awt/font/GlyphVector.html#getFontRenderContext())()            Returns the [FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) associated with this GlyphVector. |
| int | [**getGlyphCharIndex**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphCharIndex(int))(int glyphIndex)            Returns the character index of the specified glyph. |
| int[] | [**getGlyphCharIndices**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphCharIndices(int,%20int,%20int%5B%5D))(int beginGlyphIndex, int numEntries, int[] codeReturn)            Returns the character indices of the specified glyphs. |
| abstract  int | [**getGlyphCode**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphCode(int))(int glyphIndex)            Returns the glyphcode of the specified glyph. |
| abstract  int[] | [**getGlyphCodes**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphCodes(int,%20int,%20int%5B%5D))(int beginGlyphIndex, int numEntries, int[] codeReturn)            Returns an array of glyphcodes for the specified glyphs. |
| abstract  [GlyphJustificationInfo](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html) | [**getGlyphJustificationInfo**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphJustificationInfo(int))(int glyphIndex)            Returns the justification information for the glyph at the specified index into this GlyphVector. |
| abstract  [Shape](http://docs.google.com/java/awt/Shape.html) | [**getGlyphLogicalBounds**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphLogicalBounds(int))(int glyphIndex)            Returns the logical bounds of the specified glyph within this GlyphVector. |
| abstract  [GlyphMetrics](http://docs.google.com/java/awt/font/GlyphMetrics.html) | [**getGlyphMetrics**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphMetrics(int))(int glyphIndex)            Returns the metrics of the glyph at the specified index into this GlyphVector. |
| abstract  [Shape](http://docs.google.com/java/awt/Shape.html) | [**getGlyphOutline**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphOutline(int))(int glyphIndex)            Returns a Shape whose interior corresponds to the visual representation of the specified glyph within this GlyphVector. |
| [Shape](http://docs.google.com/java/awt/Shape.html) | [**getGlyphOutline**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphOutline(int,%20float,%20float))(int glyphIndex, float x, float y)            Returns a Shape whose interior corresponds to the visual representation of the specified glyph within this GlyphVector, offset to x, y. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getGlyphPixelBounds**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphPixelBounds(int,%20java.awt.font.FontRenderContext,%20float,%20float))(int index, [FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) renderFRC, float x, float y)            Returns the pixel bounds of the glyph at index when this GlyphVector is rendered in a Graphics with the given FontRenderContext at the given location. |
| abstract  [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getGlyphPosition**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphPosition(int))(int glyphIndex)            Returns the position of the specified glyph relative to the origin of this GlyphVector. |
| abstract  float[] | [**getGlyphPositions**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphPositions(int,%20int,%20float%5B%5D))(int beginGlyphIndex, int numEntries, float[] positionReturn)            Returns an array of glyph positions for the specified glyphs. |
| abstract  [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) | [**getGlyphTransform**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphTransform(int))(int glyphIndex)            Returns the transform of the specified glyph within this GlyphVector. |
| abstract  [Shape](http://docs.google.com/java/awt/Shape.html) | [**getGlyphVisualBounds**](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphVisualBounds(int))(int glyphIndex)            Returns the visual bounds of the specified glyph within the GlyphVector. |
| int | [**getLayoutFlags**](http://docs.google.com/java/awt/font/GlyphVector.html#getLayoutFlags())()            Returns flags describing the global state of the GlyphVector. |
| abstract  [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getLogicalBounds**](http://docs.google.com/java/awt/font/GlyphVector.html#getLogicalBounds())()            Returns the logical bounds of this GlyphVector. |
| abstract  int | [**getNumGlyphs**](http://docs.google.com/java/awt/font/GlyphVector.html#getNumGlyphs())()            Returns the number of glyphs in this GlyphVector. |
| abstract  [Shape](http://docs.google.com/java/awt/Shape.html) | [**getOutline**](http://docs.google.com/java/awt/font/GlyphVector.html#getOutline())()            Returns a Shape whose interior corresponds to the visual representation of this GlyphVector. |
| abstract  [Shape](http://docs.google.com/java/awt/Shape.html) | [**getOutline**](http://docs.google.com/java/awt/font/GlyphVector.html#getOutline(float,%20float))(float x, float y)            Returns a Shape whose interior corresponds to the visual representation of this GlyphVector when rendered at x, y. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getPixelBounds**](http://docs.google.com/java/awt/font/GlyphVector.html#getPixelBounds(java.awt.font.FontRenderContext,%20float,%20float))([FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) renderFRC, float x, float y)            Returns the pixel bounds of this GlyphVector when rendered in a graphics with the given FontRenderContext at the given location. |
| abstract  [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getVisualBounds**](http://docs.google.com/java/awt/font/GlyphVector.html#getVisualBounds())()            Returns the visual bounds of this GlyphVector The visual bounds is the bounding box of the outline of this GlyphVector. |
| abstract  void | [**performDefaultLayout**](http://docs.google.com/java/awt/font/GlyphVector.html#performDefaultLayout())()            Assigns default positions to each glyph in this GlyphVector. |
| abstract  void | [**setGlyphPosition**](http://docs.google.com/java/awt/font/GlyphVector.html#setGlyphPosition(int,%20java.awt.geom.Point2D))(int glyphIndex, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) newPos)            Sets the position of the specified glyph within this GlyphVector. |
| abstract  void | [**setGlyphTransform**](http://docs.google.com/java/awt/font/GlyphVector.html#setGlyphTransform(int,%20java.awt.geom.AffineTransform))(int glyphIndex, [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) newTX)            Sets the transform of the specified glyph within this GlyphVector. |

| **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** |
| --- |

### FLAG\_HAS\_TRANSFORMS

public static final int **FLAG\_HAS\_TRANSFORMS**

A flag used with getLayoutFlags that indicates that this GlyphVector has per-glyph transforms.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphVector.FLAG_HAS_TRANSFORMS)

### FLAG\_HAS\_POSITION\_ADJUSTMENTS

public static final int **FLAG\_HAS\_POSITION\_ADJUSTMENTS**

A flag used with getLayoutFlags that indicates that this GlyphVector has position adjustments. When this is true, the glyph positions don't match the accumulated default advances of the glyphs (for example, if kerning has been done).

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphVector.FLAG_HAS_POSITION_ADJUSTMENTS)

### FLAG\_RUN\_RTL

public static final int **FLAG\_RUN\_RTL**

A flag used with getLayoutFlags that indicates that this GlyphVector has a right-to-left run direction. This refers to the glyph-to-char mapping and does not imply that the visual locations of the glyphs are necessarily in this order, although generally they will be.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphVector.FLAG_RUN_RTL)

### FLAG\_COMPLEX\_GLYPHS

public static final int **FLAG\_COMPLEX\_GLYPHS**

A flag used with getLayoutFlags that indicates that this GlyphVector has a complex glyph-to-char mapping (one that does not map glyphs to chars one-to-one in strictly ascending or descending order matching the run direction).

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphVector.FLAG_COMPLEX_GLYPHS)

### FLAG\_MASK

public static final int **FLAG\_MASK**

A mask for supported flags from getLayoutFlags. Only bits covered by the mask should be tested.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.font.GlyphVector.FLAG_MASK)

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

### GlyphVector

public **GlyphVector**()

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

### getFont

public abstract [Font](http://docs.google.com/java/awt/Font.html) **getFont**()

Returns the Font associated with this GlyphVector.

**Returns:**Font used to create this GlyphVector.**See Also:**[Font](http://docs.google.com/java/awt/Font.html)

### getFontRenderContext

public abstract [FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) **getFontRenderContext**()

Returns the [FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) associated with this GlyphVector.

**Returns:**FontRenderContext used to create this GlyphVector.**See Also:**[FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html), [Font](http://docs.google.com/java/awt/Font.html)

### performDefaultLayout

public abstract void **performDefaultLayout**()

Assigns default positions to each glyph in this GlyphVector. This can destroy information generated during initial layout of this GlyphVector.

### getNumGlyphs

public abstract int **getNumGlyphs**()

Returns the number of glyphs in this GlyphVector.

**Returns:**number of glyphs in this GlyphVector.

### getGlyphCode

public abstract int **getGlyphCode**(int glyphIndex)

Returns the glyphcode of the specified glyph. This return value is meaningless to anything other than the Font object that created this GlyphVector.

**Parameters:**glyphIndex - the index into this GlyphVector that corresponds to the glyph from which to retrieve the glyphcode. **Returns:**the glyphcode of the glyph at the specified glyphIndex. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector

### getGlyphCodes

public abstract int[] **getGlyphCodes**(int beginGlyphIndex,  
 int numEntries,  
 int[] codeReturn)

Returns an array of glyphcodes for the specified glyphs. The contents of this return value are meaningless to anything other than the Font used to create this GlyphVector. This method is used for convenience and performance when processing glyphcodes. If no array is passed in, a new array is created.

**Parameters:**beginGlyphIndex - the index into this GlyphVector at which to start retrieving glyphcodesnumEntries - the number of glyphcodes to retrievecodeReturn - the array that receives the glyphcodes and is then returned **Returns:**an array of glyphcodes for the specified glyphs. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if numEntries is less than 0 [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if beginGlyphIndex is less than 0 [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if the sum of beginGlyphIndex and numEntries is greater than the number of glyphs in this GlyphVector

### getGlyphCharIndex

public int **getGlyphCharIndex**(int glyphIndex)

Returns the character index of the specified glyph. The character index is the index of the first logical character represented by the glyph. The default implementation assumes a one-to-one, left-to-right mapping of glyphs to characters.

**Parameters:**glyphIndex - the index of the glyph **Returns:**the index of the first character represented by the glyph**Since:** 1.4

### getGlyphCharIndices

public int[] **getGlyphCharIndices**(int beginGlyphIndex,  
 int numEntries,  
 int[] codeReturn)

Returns the character indices of the specified glyphs. The character index is the index of the first logical character represented by the glyph. Indices are returned in glyph order. The default implementation invokes getGlyphCharIndex for each glyph, and subclassers will probably want to override this implementation for performance reasons. Use this method for convenience and performance in processing of glyphcodes. If no array is passed in, a new array is created.

**Parameters:**beginGlyphIndex - the index of the first glyphnumEntries - the number of glyph indicescodeReturn - the array into which to return the character indices **Returns:**an array of character indices, one per glyph.**Since:** 1.4

### getLogicalBounds

public abstract [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **getLogicalBounds**()

Returns the logical bounds of this GlyphVector. This method is used when positioning this GlyphVector in relation to visually adjacent GlyphVector objects.

**Returns:**a [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) that is the logical bounds of this GlyphVector.

### getVisualBounds

public abstract [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **getVisualBounds**()

Returns the visual bounds of this GlyphVector The visual bounds is the bounding box of the outline of this GlyphVector. Because of rasterization and alignment of pixels, it is possible that this box does not enclose all pixels affected by rendering this GlyphVector.

**Returns:**a Rectangle2D that is the bounding box of this GlyphVector.

### getPixelBounds

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getPixelBounds**([FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) renderFRC,  
 float x,  
 float y)

Returns the pixel bounds of this GlyphVector when rendered in a graphics with the given FontRenderContext at the given location. The renderFRC need not be the same as the FontRenderContext of this GlyphVector, and can be null. If it is null, the FontRenderContext of this GlyphVector is used. The default implementation returns the visual bounds, offset to x, y and rounded out to the next integer value (i.e. returns an integer rectangle which encloses the visual bounds) and ignores the FRC. Subclassers should override this method.

**Parameters:**renderFRC - the FontRenderContext of the Graphics.x - the x-coordinate at which to render this GlyphVector.y - the y-coordinate at which to render this GlyphVector. **Returns:**a Rectangle bounding the pixels that would be affected.**Since:** 1.4

### getOutline

public abstract [Shape](http://docs.google.com/java/awt/Shape.html) **getOutline**()

Returns a Shape whose interior corresponds to the visual representation of this GlyphVector.

**Returns:**a Shape that is the outline of this GlyphVector.

### getOutline

public abstract [Shape](http://docs.google.com/java/awt/Shape.html) **getOutline**(float x,  
 float y)

Returns a Shape whose interior corresponds to the visual representation of this GlyphVector when rendered at x, y.

**Parameters:**x - the X coordinate of this GlyphVector.y - the Y coordinate of this GlyphVector. **Returns:**a Shape that is the outline of this GlyphVector when rendered at the specified coordinates.

### getGlyphOutline

public abstract [Shape](http://docs.google.com/java/awt/Shape.html) **getGlyphOutline**(int glyphIndex)

Returns a Shape whose interior corresponds to the visual representation of the specified glyph within this GlyphVector. The outline returned by this method is positioned around the origin of each individual glyph.

**Parameters:**glyphIndex - the index into this GlyphVector **Returns:**a Shape that is the outline of the glyph at the specified glyphIndex of this GlyphVector. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector

### getGlyphOutline

public [Shape](http://docs.google.com/java/awt/Shape.html) **getGlyphOutline**(int glyphIndex,  
 float x,  
 float y)

Returns a Shape whose interior corresponds to the visual representation of the specified glyph within this GlyphVector, offset to x, y. The outline returned by this method is positioned around the origin of each individual glyph.

**Parameters:**glyphIndex - the index into this GlyphVectorx - the X coordinate of the location of this GlyphVectory - the Y coordinate of the location of this GlyphVector **Returns:**a Shape that is the outline of the glyph at the specified glyphIndex of this GlyphVector when rendered at the specified coordinates. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector**Since:** 1.4

### getGlyphPosition

public abstract [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) **getGlyphPosition**(int glyphIndex)

Returns the position of the specified glyph relative to the origin of this GlyphVector. If glyphIndex equals the number of of glyphs in this GlyphVector, this method returns the position after the last glyph. This position is used to define the advance of the entire GlyphVector.

**Parameters:**glyphIndex - the index into this GlyphVector **Returns:**a [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) object that is the position of the glyph at the specified glyphIndex. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than the number of glyphs in this GlyphVector**See Also:**[setGlyphPosition(int, java.awt.geom.Point2D)](http://docs.google.com/java/awt/font/GlyphVector.html#setGlyphPosition(int,%20java.awt.geom.Point2D))

### setGlyphPosition

public abstract void **setGlyphPosition**(int glyphIndex,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) newPos)

Sets the position of the specified glyph within this GlyphVector. If glyphIndex equals the number of of glyphs in this GlyphVector, this method sets the position after the last glyph. This position is used to define the advance of the entire GlyphVector.

**Parameters:**glyphIndex - the index into this GlyphVectornewPos - the Point2D at which to position the glyph at the specified glyphIndex **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than the number of glyphs in this GlyphVector**See Also:**[getGlyphPosition(int)](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphPosition(int))

### getGlyphTransform

public abstract [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) **getGlyphTransform**(int glyphIndex)

Returns the transform of the specified glyph within this GlyphVector. The transform is relative to the glyph position. If no special transform has been applied, null can be returned. A null return indicates an identity transform.

**Parameters:**glyphIndex - the index into this GlyphVector **Returns:**an [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) that is the transform of the glyph at the specified glyphIndex. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector**See Also:**[setGlyphTransform(int, java.awt.geom.AffineTransform)](http://docs.google.com/java/awt/font/GlyphVector.html#setGlyphTransform(int,%20java.awt.geom.AffineTransform))

### setGlyphTransform

public abstract void **setGlyphTransform**(int glyphIndex,  
 [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) newTX)

Sets the transform of the specified glyph within this GlyphVector. The transform is relative to the glyph position. A null argument for newTX indicates that no special transform is applied for the specified glyph. This method can be used to rotate, mirror, translate and scale the glyph. Adding a transform can result in signifant performance changes.

**Parameters:**glyphIndex - the index into this GlyphVectornewTX - the new transform of the glyph at glyphIndex **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector**See Also:**[getGlyphTransform(int)](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphTransform(int))

### getLayoutFlags

public int **getLayoutFlags**()

Returns flags describing the global state of the GlyphVector. Flags not described below are reserved. The default implementation returns 0 (meaning false) for the position adjustments, transforms, rtl, and complex flags. Subclassers should override this method, and make sure it correctly describes the GlyphVector and corresponds to the results of related calls.

**Returns:**an int containing the flags describing the state**Since:** 1.4 **See Also:**[FLAG\_HAS\_POSITION\_ADJUSTMENTS](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_HAS_POSITION_ADJUSTMENTS), [FLAG\_HAS\_TRANSFORMS](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_HAS_TRANSFORMS), [FLAG\_RUN\_RTL](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_RUN_RTL), [FLAG\_COMPLEX\_GLYPHS](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_COMPLEX_GLYPHS), [FLAG\_MASK](http://docs.google.com/java/awt/font/GlyphVector.html#FLAG_MASK)

### getGlyphPositions

public abstract float[] **getGlyphPositions**(int beginGlyphIndex,  
 int numEntries,  
 float[] positionReturn)

Returns an array of glyph positions for the specified glyphs. This method is used for convenience and performance when processing glyph positions. If no array is passed in, a new array is created. Even numbered array entries beginning with position zero are the X coordinates of the glyph numbered beginGlyphIndex + position/2. Odd numbered array entries beginning with position one are the Y coordinates of the glyph numbered beginGlyphIndex + (position-1)/2. If beginGlyphIndex equals the number of of glyphs in this GlyphVector, this method gets the position after the last glyph and this position is used to define the advance of the entire GlyphVector.

**Parameters:**beginGlyphIndex - the index at which to begin retrieving glyph positionsnumEntries - the number of glyphs to retrievepositionReturn - the array that receives the glyph positions and is then returned. **Returns:**an array of glyph positions specified by beginGlyphIndex and numEntries. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if numEntries is less than 0 [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if beginGlyphIndex is less than 0 [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if the sum of beginGlyphIndex and numEntries is greater than the number of glyphs in this GlyphVector plus one

### getGlyphLogicalBounds

public abstract [Shape](http://docs.google.com/java/awt/Shape.html) **getGlyphLogicalBounds**(int glyphIndex)

Returns the logical bounds of the specified glyph within this GlyphVector. These logical bounds have a total of four edges, with two edges parallel to the baseline under the glyph's transform and the other two edges are shared with adjacent glyphs if they are present. This method is useful for hit-testing of the specified glyph, positioning of a caret at the leading or trailing edge of a glyph, and for drawing a highlight region around the specified glyph.

**Parameters:**glyphIndex - the index into this GlyphVector that corresponds to the glyph from which to retrieve its logical bounds **Returns:**a Shape that is the logical bounds of the glyph at the specified glyphIndex. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector**See Also:**[getGlyphVisualBounds(int)](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphVisualBounds(int))

### getGlyphVisualBounds

public abstract [Shape](http://docs.google.com/java/awt/Shape.html) **getGlyphVisualBounds**(int glyphIndex)

Returns the visual bounds of the specified glyph within the GlyphVector. The bounds returned by this method is positioned around the origin of each individual glyph.

**Parameters:**glyphIndex - the index into this GlyphVector that corresponds to the glyph from which to retrieve its visual bounds **Returns:**a Shape that is the visual bounds of the glyph at the specified glyphIndex. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector**See Also:**[getGlyphLogicalBounds(int)](http://docs.google.com/java/awt/font/GlyphVector.html#getGlyphLogicalBounds(int))

### getGlyphPixelBounds

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getGlyphPixelBounds**(int index,  
 [FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) renderFRC,  
 float x,  
 float y)

Returns the pixel bounds of the glyph at index when this GlyphVector is rendered in a Graphics with the given FontRenderContext at the given location. The renderFRC need not be the same as the FontRenderContext of this GlyphVector, and can be null. If it is null, the FontRenderContext of this GlyphVector is used. The default implementation returns the visual bounds of the glyph, offset to x, y and rounded out to the next integer value, and ignores the FRC. Subclassers should override this method.

**Parameters:**index - the index of the glyph.renderFRC - the FontRenderContext of the Graphics.x - the X position at which to render this GlyphVector.y - the Y position at which to render this GlyphVector. **Returns:**a Rectangle bounding the pixels that would be affected.**Since:** 1.4

### getGlyphMetrics

public abstract [GlyphMetrics](http://docs.google.com/java/awt/font/GlyphMetrics.html) **getGlyphMetrics**(int glyphIndex)

Returns the metrics of the glyph at the specified index into this GlyphVector.

**Parameters:**glyphIndex - the index into this GlyphVector that corresponds to the glyph from which to retrieve its metrics **Returns:**a [GlyphMetrics](http://docs.google.com/java/awt/font/GlyphMetrics.html) object that represents the metrics of the glyph at the specified glyphIndex into this GlyphVector. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector

### getGlyphJustificationInfo

public abstract [GlyphJustificationInfo](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html) **getGlyphJustificationInfo**(int glyphIndex)

Returns the justification information for the glyph at the specified index into this GlyphVector.

**Parameters:**glyphIndex - the index into this GlyphVector that corresponds to the glyph from which to retrieve its justification properties **Returns:**a [GlyphJustificationInfo](http://docs.google.com/java/awt/font/GlyphJustificationInfo.html) object that represents the justification properties of the glyph at the specified glyphIndex into this GlyphVector. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - if glyphIndex is less than 0 or greater than or equal to the number of glyphs in this GlyphVector

### equals

public abstract boolean **equals**([GlyphVector](http://docs.google.com/java/awt/font/GlyphVector.html) set)

Tests if the specified GlyphVector exactly equals this GlyphVector.

**Parameters:**set - the specified GlyphVector to test **Returns:**true if the specified GlyphVector equals this GlyphVector; false otherwise.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GlyphVector.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/GlyphMetrics.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/font/GraphicAttribute.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/font/GlyphVector.html)    [**NO FRAMES**](http://docs.google.com/GlyphVector.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#lnxbz9) | [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).