| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SynthGraphicsUtils.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/plaf/synth/SynthContext.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/synth/SynthLookAndFeel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/synth/SynthGraphicsUtils.html)    [**NO FRAMES**](http://docs.google.com/SynthGraphicsUtils.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **javax.swing.plaf.synth**

Class SynthGraphicsUtils

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

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

Wrapper for primitive graphics calls.

**Since:** 1.5

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

| **Method Summary** | |
| --- | --- |
| int | [**computeStringWidth**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#computeStringWidth(javax.swing.plaf.synth.SynthContext,%20java.awt.Font,%20java.awt.FontMetrics,%20java.lang.String))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss, [Font](http://docs.google.com/java/awt/Font.html) font, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics, [String](http://docs.google.com/java/lang/String.html) text)            Returns the size of the passed in string. |
| void | [**drawLine**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#drawLine(javax.swing.plaf.synth.SynthContext,%20java.lang.Object,%20java.awt.Graphics,%20int,%20int,%20int,%20int))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) context, [Object](http://docs.google.com/java/lang/Object.html) paintKey, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x1, int y1, int x2, int y2)            Draws a line between the two end points. |
| void | [**drawLine**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#drawLine(javax.swing.plaf.synth.SynthContext,%20java.lang.Object,%20java.awt.Graphics,%20int,%20int,%20int,%20int,%20java.lang.Object))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) context, [Object](http://docs.google.com/java/lang/Object.html) paintKey, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, int x1, int y1, int x2, int y2, [Object](http://docs.google.com/java/lang/Object.html) styleKey)            Draws a line between the two end points. |
| int | [**getMaximumCharHeight**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#getMaximumCharHeight(javax.swing.plaf.synth.SynthContext))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) context)            Returns the maximum height of the the Font from the passed in SynthContext. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#getMaximumSize(javax.swing.plaf.synth.SynthContext,%20java.awt.Font,%20java.lang.String,%20javax.swing.Icon,%20int,%20int,%20int,%20int,%20int,%20int))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss, [Font](http://docs.google.com/java/awt/Font.html) font, [String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int hAlign, int vAlign, int hTextPosition, int vTextPosition, int iconTextGap, int mnemonicIndex)            Returns the maximum size needed to properly render an icon and text. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#getMinimumSize(javax.swing.plaf.synth.SynthContext,%20java.awt.Font,%20java.lang.String,%20javax.swing.Icon,%20int,%20int,%20int,%20int,%20int,%20int))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss, [Font](http://docs.google.com/java/awt/Font.html) font, [String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int hAlign, int vAlign, int hTextPosition, int vTextPosition, int iconTextGap, int mnemonicIndex)            Returns the minimum size needed to properly render an icon and text. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#getPreferredSize(javax.swing.plaf.synth.SynthContext,%20java.awt.Font,%20java.lang.String,%20javax.swing.Icon,%20int,%20int,%20int,%20int,%20int,%20int))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss, [Font](http://docs.google.com/java/awt/Font.html) font, [String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int hAlign, int vAlign, int hTextPosition, int vTextPosition, int iconTextGap, int mnemonicIndex)            Returns the preferred size needed to properly render an icon and text. |
| [String](http://docs.google.com/java/lang/String.html) | [**layoutText**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#layoutText(javax.swing.plaf.synth.SynthContext,%20java.awt.FontMetrics,%20java.lang.String,%20javax.swing.Icon,%20int,%20int,%20int,%20int,%20java.awt.Rectangle,%20java.awt.Rectangle,%20java.awt.Rectangle,%20int))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fm, [String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int hAlign, int vAlign, int hTextPosition, int vTextPosition, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewR, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconR, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textR, int iconTextGap)            Lays out text and an icon returning, by reference, the location to place the icon and text. |
| void | [**paintText**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#paintText(javax.swing.plaf.synth.SynthContext,%20java.awt.Graphics,%20java.lang.String,%20javax.swing.Icon,%20int,%20int,%20int,%20int,%20int,%20int,%20int))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, [String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int hAlign, int vAlign, int hTextPosition, int vTextPosition, int iconTextGap, int mnemonicIndex, int textOffset)            Paints an icon and text. |
| void | [**paintText**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#paintText(javax.swing.plaf.synth.SynthContext,%20java.awt.Graphics,%20java.lang.String,%20int,%20int,%20int))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, [String](http://docs.google.com/java/lang/String.html) text, int x, int y, int mnemonicIndex)            Paints text at the specified location. |
| void | [**paintText**](http://docs.google.com/javax/swing/plaf/synth/SynthGraphicsUtils.html#paintText(javax.swing.plaf.synth.SynthContext,%20java.awt.Graphics,%20java.lang.String,%20java.awt.Rectangle,%20int))([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss, [Graphics](http://docs.google.com/java/awt/Graphics.html) g, [String](http://docs.google.com/java/lang/String.html) text, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds, int mnemonicIndex)            Paints text at the specified 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** |
| --- |

### SynthGraphicsUtils

public **SynthGraphicsUtils**()

Creates a SynthGraphicsUtils.

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

### drawLine

public void **drawLine**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) context,  
 [Object](http://docs.google.com/java/lang/Object.html) paintKey,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x1,  
 int y1,  
 int x2,  
 int y2)

Draws a line between the two end points.

**Parameters:**context - Identifies hosting region.paintKey - Identifies the portion of the component being asked to paint, may be null.g - Graphics object to paint tox1 - x originy1 - y originx2 - x destinationy2 - y destination

### drawLine

public void **drawLine**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) context,  
 [Object](http://docs.google.com/java/lang/Object.html) paintKey,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int x1,  
 int y1,  
 int x2,  
 int y2,  
 [Object](http://docs.google.com/java/lang/Object.html) styleKey)

Draws a line between the two end points.

This implementation supports only one line style key, "dashed". The "dashed" line style is applied only to vertical and horizontal lines.

Specifying null or any key different from "dashed" will draw solid lines.

**Parameters:**context - identifies hosting regionpaintKey - identifies the portion of the component being asked to paint, may be nullg - Graphics object to paint tox1 - x originy1 - y originx2 - x destinationy2 - y destinationstyleKey - identifies the requested style of the line (e.g. "dashed")**Since:** 1.6

### layoutText

public [String](http://docs.google.com/java/lang/String.html) **layoutText**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss,  
 [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fm,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int hAlign,  
 int vAlign,  
 int hTextPosition,  
 int vTextPosition,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewR,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconR,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textR,  
 int iconTextGap)

Lays out text and an icon returning, by reference, the location to place the icon and text.

**Parameters:**ss - SynthContextfm - FontMetrics for the Font to use, this may be ignoredtext - Text to layouticon - Icon to layouthAlign - horizontal alignmentvAlign - vertical alignmenthTextPosition - horizontal text positionvTextPosition - vertical text positionviewR - Rectangle to layout text and icon in.iconR - Rectangle to place icon bounds intextR - Rectangle to place text iniconTextGap - gap between icon and text

### computeStringWidth

public int **computeStringWidth**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss,  
 [Font](http://docs.google.com/java/awt/Font.html) font,  
 [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) metrics,  
 [String](http://docs.google.com/java/lang/String.html) text)

Returns the size of the passed in string.

**Parameters:**ss - SynthContextfont - Font to usemetrics - FontMetrics, may be ignoredtext - Text to get size of.

### getMinimumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMinimumSize**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss,  
 [Font](http://docs.google.com/java/awt/Font.html) font,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int hAlign,  
 int vAlign,  
 int hTextPosition,  
 int vTextPosition,  
 int iconTextGap,  
 int mnemonicIndex)

Returns the minimum size needed to properly render an icon and text.

**Parameters:**ss - SynthContextfont - Font to usetext - Text to layouticon - Icon to layouthAlign - horizontal alignmentvAlign - vertical alignmenthTextPosition - horizontal text positionvTextPosition - vertical text positioniconTextGap - gap between icon and textmnemonicIndex - Index into text to render the mnemonic at, -1 indicates no mnemonic.

### getMaximumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMaximumSize**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss,  
 [Font](http://docs.google.com/java/awt/Font.html) font,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int hAlign,  
 int vAlign,  
 int hTextPosition,  
 int vTextPosition,  
 int iconTextGap,  
 int mnemonicIndex)

Returns the maximum size needed to properly render an icon and text.

**Parameters:**ss - SynthContextfont - Font to usetext - Text to layouticon - Icon to layouthAlign - horizontal alignmentvAlign - vertical alignmenthTextPosition - horizontal text positionvTextPosition - vertical text positioniconTextGap - gap between icon and textmnemonicIndex - Index into text to render the mnemonic at, -1 indicates no mnemonic.

### getMaximumCharHeight

public int **getMaximumCharHeight**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) context)

Returns the maximum height of the the Font from the passed in SynthContext.

**Parameters:**context - SynthContext used to determine font. **Returns:**maximum height of the characters for the font from the passed in context.

### getPreferredSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getPreferredSize**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss,  
 [Font](http://docs.google.com/java/awt/Font.html) font,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int hAlign,  
 int vAlign,  
 int hTextPosition,  
 int vTextPosition,  
 int iconTextGap,  
 int mnemonicIndex)

Returns the preferred size needed to properly render an icon and text.

**Parameters:**ss - SynthContextfont - Font to usetext - Text to layouticon - Icon to layouthAlign - horizontal alignmentvAlign - vertical alignmenthTextPosition - horizontal text positionvTextPosition - vertical text positioniconTextGap - gap between icon and textmnemonicIndex - Index into text to render the mnemonic at, -1 indicates no mnemonic.

### paintText

public void **paintText**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds,  
 int mnemonicIndex)

Paints text at the specified location. This will not attempt to render the text as html nor will it offset by the insets of the component.

**Parameters:**ss - SynthContextg - Graphics used to render string in.text - Text to renderbounds - Bounds of the text to be drawn.mnemonicIndex - Index to draw string at.

### paintText

public void **paintText**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 int x,  
 int y,  
 int mnemonicIndex)

Paints text at the specified location. This will not attempt to render the text as html nor will it offset by the insets of the component.

**Parameters:**ss - SynthContextg - Graphics used to render string in.text - Text to renderx - X location to draw text at.y - Upper left corner to draw text at.mnemonicIndex - Index to draw string at.

### paintText

public void **paintText**([SynthContext](http://docs.google.com/javax/swing/plaf/synth/SynthContext.html) ss,  
 [Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int hAlign,  
 int vAlign,  
 int hTextPosition,  
 int vTextPosition,  
 int iconTextGap,  
 int mnemonicIndex,  
 int textOffset)

Paints an icon and text. This will render the text as html, if necessary, and offset the location by the insets of the component.

**Parameters:**ss - SynthContextg - Graphics to render string and icon intotext - Text to layouticon - Icon to layouthAlign - horizontal alignmentvAlign - vertical alignmenthTextPosition - horizontal text positionvTextPosition - vertical text positioniconTextGap - gap between icon and textmnemonicIndex - Index into text to render the mnemonic at, -1 indicates no mnemonic.textOffset - Amount to offset the text when painting

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SynthGraphicsUtils.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/plaf/synth/SynthContext.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/synth/SynthLookAndFeel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/synth/SynthGraphicsUtils.html)    [**NO FRAMES**](http://docs.google.com/SynthGraphicsUtils.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).