| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DebugGraphics.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/ComponentInputMap.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/DefaultBoundedRangeModel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/DebugGraphics.html)    [**NO FRAMES**](http://docs.google.com/DebugGraphics.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#lnxbz9) | [METHOD](#2jxsxqh) |

## **javax.swing**

Class DebugGraphics

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

public class **DebugGraphics**extends [Graphics](http://docs.google.com/java/awt/Graphics.html)

Graphics subclass supporting graphics debugging. Overrides most methods from Graphics. DebugGraphics objects are rarely created by hand. They are most frequently created automatically when a JComponent's debugGraphicsOptions are changed using the setDebugGraphicsOptions() method.

NOTE: You must turn off double buffering to use DebugGraphics: RepaintManager repaintManager = RepaintManager.currentManager(component); repaintManager.setDoubleBufferingEnabled(false);

**See Also:**[JComponent.setDebugGraphicsOptions(int)](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [RepaintManager.currentManager(java.awt.Component)](http://docs.google.com/javax/swing/RepaintManager.html#currentManager(java.awt.Component)), [RepaintManager.setDoubleBufferingEnabled(boolean)](http://docs.google.com/javax/swing/RepaintManager.html#setDoubleBufferingEnabled(boolean))

| **Field Summary** | |
| --- | --- |
| static int | [**BUFFERED\_OPTION**](http://docs.google.com/javax/swing/DebugGraphics.html#BUFFERED_OPTION)            Show buffered operations in a separate Frame. |
| static int | [**FLASH\_OPTION**](http://docs.google.com/javax/swing/DebugGraphics.html#FLASH_OPTION)            Flash graphics operations. |
| static int | [**LOG\_OPTION**](http://docs.google.com/javax/swing/DebugGraphics.html#LOG_OPTION)            Log graphics operations. |
| static int | [**NONE\_OPTION**](http://docs.google.com/javax/swing/DebugGraphics.html#NONE_OPTION)            Don't debug graphics operations. |

| **Constructor Summary** | |
| --- | --- |
| [**DebugGraphics**](http://docs.google.com/javax/swing/DebugGraphics.html#DebugGraphics())()            Constructs a new debug graphics context that supports slowed down drawing. |
| [**DebugGraphics**](http://docs.google.com/javax/swing/DebugGraphics.html#DebugGraphics(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) graphics)            Constructs a debug graphics context from an existing graphics context that supports slowed down drawing. |
| [**DebugGraphics**](http://docs.google.com/javax/swing/DebugGraphics.html#DebugGraphics(java.awt.Graphics,%20javax.swing.JComponent))([Graphics](http://docs.google.com/java/awt/Graphics.html) graphics, [JComponent](http://docs.google.com/javax/swing/JComponent.html) component)            Constructs a debug graphics context from an existing graphics context that slows down drawing for the specified component. |

| **Method Summary** | |
| --- | --- |
| void | [**clearRect**](http://docs.google.com/javax/swing/DebugGraphics.html#clearRect(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Overrides Graphics.clearRect. |
| void | [**clipRect**](http://docs.google.com/javax/swing/DebugGraphics.html#clipRect(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Overrides Graphics.clipRect. |
| void | [**copyArea**](http://docs.google.com/javax/swing/DebugGraphics.html#copyArea(int,%20int,%20int,%20int,%20int,%20int))(int x, int y, int width, int height, int destX, int destY)            Overrides Graphics.copyArea. |
| [Graphics](http://docs.google.com/java/awt/Graphics.html) | [**create**](http://docs.google.com/javax/swing/DebugGraphics.html#create())()            Overrides Graphics.create to return a DebugGraphics object. |
| [Graphics](http://docs.google.com/java/awt/Graphics.html) | [**create**](http://docs.google.com/javax/swing/DebugGraphics.html#create(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Overrides Graphics.create to return a DebugGraphics object. |
| void | [**dispose**](http://docs.google.com/javax/swing/DebugGraphics.html#dispose())()            Overrides Graphics.dispose. |
| void | [**draw3DRect**](http://docs.google.com/javax/swing/DebugGraphics.html#draw3DRect(int,%20int,%20int,%20int,%20boolean))(int x, int y, int width, int height, boolean raised)            Overrides Graphics.draw3DRect. |
| void | [**drawArc**](http://docs.google.com/javax/swing/DebugGraphics.html#drawArc(int,%20int,%20int,%20int,%20int,%20int))(int x, int y, int width, int height, int startAngle, int arcAngle)            Overrides Graphics.drawArc. |
| void | [**drawBytes**](http://docs.google.com/javax/swing/DebugGraphics.html#drawBytes(byte%5B%5D,%20int,%20int,%20int,%20int))(byte[] data, int offset, int length, int x, int y)            Overrides Graphics.drawBytes. |
| void | [**drawChars**](http://docs.google.com/javax/swing/DebugGraphics.html#drawChars(char%5B%5D,%20int,%20int,%20int,%20int))(char[] data, int offset, int length, int x, int y)            Overrides Graphics.drawChars. |
| boolean | [**drawImage**](http://docs.google.com/javax/swing/DebugGraphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.Color,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) img, int x, int y, [Color](http://docs.google.com/java/awt/Color.html) bgcolor, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Overrides Graphics.drawImage. |
| boolean | [**drawImage**](http://docs.google.com/javax/swing/DebugGraphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) img, int x, int y, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Overrides Graphics.drawImage. |
| boolean | [**drawImage**](http://docs.google.com/javax/swing/DebugGraphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20java.awt.Color,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) img, int x, int y, int width, int height, [Color](http://docs.google.com/java/awt/Color.html) bgcolor, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Overrides Graphics.drawImage. |
| boolean | [**drawImage**](http://docs.google.com/javax/swing/DebugGraphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) img, int x, int y, int width, int height, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Overrides Graphics.drawImage. |
| boolean | [**drawImage**](http://docs.google.com/javax/swing/DebugGraphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20java.awt.Color,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) img, int dx1, int dy1, int dx2, int dy2, int sx1, int sy1, int sx2, int sy2, [Color](http://docs.google.com/java/awt/Color.html) bgcolor, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Overrides Graphics.drawImage. |
| boolean | [**drawImage**](http://docs.google.com/javax/swing/DebugGraphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) img, int dx1, int dy1, int dx2, int dy2, int sx1, int sy1, int sx2, int sy2, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Overrides Graphics.drawImage. |
| void | [**drawLine**](http://docs.google.com/javax/swing/DebugGraphics.html#drawLine(int,%20int,%20int,%20int))(int x1, int y1, int x2, int y2)            Overrides Graphics.drawLine. |
| void | [**drawOval**](http://docs.google.com/javax/swing/DebugGraphics.html#drawOval(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Overrides Graphics.drawOval. |
| void | [**drawPolygon**](http://docs.google.com/javax/swing/DebugGraphics.html#drawPolygon(int%5B%5D,%20int%5B%5D,%20int))(int[] xPoints, int[] yPoints, int nPoints)            Overrides Graphics.drawPolygon. |
| void | [**drawPolyline**](http://docs.google.com/javax/swing/DebugGraphics.html#drawPolyline(int%5B%5D,%20int%5B%5D,%20int))(int[] xPoints, int[] yPoints, int nPoints)            Overrides Graphics.drawPolyline. |
| void | [**drawRect**](http://docs.google.com/javax/swing/DebugGraphics.html#drawRect(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Overrides Graphics.drawRect. |
| void | [**drawRoundRect**](http://docs.google.com/javax/swing/DebugGraphics.html#drawRoundRect(int,%20int,%20int,%20int,%20int,%20int))(int x, int y, int width, int height, int arcWidth, int arcHeight)            Overrides Graphics.drawRoundRect. |
| void | [**drawString**](http://docs.google.com/javax/swing/DebugGraphics.html#drawString(java.text.AttributedCharacterIterator,%20int,%20int))([AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) iterator, int x, int y)            Overrides Graphics.drawString. |
| void | [**drawString**](http://docs.google.com/javax/swing/DebugGraphics.html#drawString(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) aString, int x, int y)            Overrides Graphics.drawString. |
| void | [**fill3DRect**](http://docs.google.com/javax/swing/DebugGraphics.html#fill3DRect(int,%20int,%20int,%20int,%20boolean))(int x, int y, int width, int height, boolean raised)            Overrides Graphics.fill3DRect. |
| void | [**fillArc**](http://docs.google.com/javax/swing/DebugGraphics.html#fillArc(int,%20int,%20int,%20int,%20int,%20int))(int x, int y, int width, int height, int startAngle, int arcAngle)            Overrides Graphics.fillArc. |
| void | [**fillOval**](http://docs.google.com/javax/swing/DebugGraphics.html#fillOval(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Overrides Graphics.fillOval. |
| void | [**fillPolygon**](http://docs.google.com/javax/swing/DebugGraphics.html#fillPolygon(int%5B%5D,%20int%5B%5D,%20int))(int[] xPoints, int[] yPoints, int nPoints)            Overrides Graphics.fillPolygon. |
| void | [**fillRect**](http://docs.google.com/javax/swing/DebugGraphics.html#fillRect(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Overrides Graphics.fillRect. |
| void | [**fillRoundRect**](http://docs.google.com/javax/swing/DebugGraphics.html#fillRoundRect(int,%20int,%20int,%20int,%20int,%20int))(int x, int y, int width, int height, int arcWidth, int arcHeight)            Overrides Graphics.fillRoundRect. |
| static [Color](http://docs.google.com/java/awt/Color.html) | [**flashColor**](http://docs.google.com/javax/swing/DebugGraphics.html#flashColor())()            Returns the Color used to flash drawing operations. |
| static int | [**flashCount**](http://docs.google.com/javax/swing/DebugGraphics.html#flashCount())()            Returns the number of times that drawing operations will flash. |
| static int | [**flashTime**](http://docs.google.com/javax/swing/DebugGraphics.html#flashTime())()            Returns the time delay of drawing operation flashing. |
| [Shape](http://docs.google.com/java/awt/Shape.html) | [**getClip**](http://docs.google.com/javax/swing/DebugGraphics.html#getClip())()            Overrides Graphics.getClip. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getClipBounds**](http://docs.google.com/javax/swing/DebugGraphics.html#getClipBounds())()            Overrides Graphics.getClipBounds. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getColor**](http://docs.google.com/javax/swing/DebugGraphics.html#getColor())()            Returns the Color used for text drawing operations. |
| int | [**getDebugOptions**](http://docs.google.com/javax/swing/DebugGraphics.html#getDebugOptions())()            Returns the current debugging options for this DebugGraphics. |
| [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/javax/swing/DebugGraphics.html#getFont())()            Returns the Font used for text drawing operations. |
| [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) | [**getFontMetrics**](http://docs.google.com/javax/swing/DebugGraphics.html#getFontMetrics())()            Overrides Graphics.getFontMetrics. |
| [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) | [**getFontMetrics**](http://docs.google.com/javax/swing/DebugGraphics.html#getFontMetrics(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) f)            Overrides Graphics.getFontMetrics. |
| boolean | [**isDrawingBuffer**](http://docs.google.com/javax/swing/DebugGraphics.html#isDrawingBuffer())()            Returns the drawingBuffer value. |
| static [PrintStream](http://docs.google.com/java/io/PrintStream.html) | [**logStream**](http://docs.google.com/javax/swing/DebugGraphics.html#logStream())()            Returns the stream to which the DebugGraphics logs drawing operations. |
| void | [**setClip**](http://docs.google.com/javax/swing/DebugGraphics.html#setClip(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Overrides Graphics.setClip. |
| void | [**setClip**](http://docs.google.com/javax/swing/DebugGraphics.html#setClip(java.awt.Shape))([Shape](http://docs.google.com/java/awt/Shape.html) clip)            Overrides Graphics.setClip. |
| void | [**setColor**](http://docs.google.com/javax/swing/DebugGraphics.html#setColor(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) aColor)            Sets the color to be used for drawing and filling lines and shapes. |
| void | [**setDebugOptions**](http://docs.google.com/javax/swing/DebugGraphics.html#setDebugOptions(int))(int options)            Enables/disables diagnostic information about every graphics operation. |
| static void | [**setFlashColor**](http://docs.google.com/javax/swing/DebugGraphics.html#setFlashColor(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) flashColor)            Sets the Color used to flash drawing operations. |
| static void | [**setFlashCount**](http://docs.google.com/javax/swing/DebugGraphics.html#setFlashCount(int))(int flashCount)            Sets the number of times that drawing operations will flash. |
| static void | [**setFlashTime**](http://docs.google.com/javax/swing/DebugGraphics.html#setFlashTime(int))(int flashTime)            Sets the time delay of drawing operation flashing. |
| void | [**setFont**](http://docs.google.com/javax/swing/DebugGraphics.html#setFont(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) aFont)            Sets the Font used for text drawing operations. |
| static void | [**setLogStream**](http://docs.google.com/javax/swing/DebugGraphics.html#setLogStream(java.io.PrintStream))([PrintStream](http://docs.google.com/java/io/PrintStream.html) stream)            Sets the stream to which the DebugGraphics logs drawing operations. |
| void | [**setPaintMode**](http://docs.google.com/javax/swing/DebugGraphics.html#setPaintMode())()            Overrides Graphics.setPaintMode. |
| void | [**setXORMode**](http://docs.google.com/javax/swing/DebugGraphics.html#setXORMode(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) aColor)            Overrides Graphics.setXORMode. |
| void | [**translate**](http://docs.google.com/javax/swing/DebugGraphics.html#translate(int,%20int))(int x, int y)            Overrides Graphics.translate. |

| **Methods inherited from class java.awt.**[**Graphics**](http://docs.google.com/java/awt/Graphics.html) |
| --- |
| [drawPolygon](http://docs.google.com/java/awt/Graphics.html#drawPolygon(java.awt.Polygon)), [fillPolygon](http://docs.google.com/java/awt/Graphics.html#fillPolygon(java.awt.Polygon)), [finalize](http://docs.google.com/java/awt/Graphics.html#finalize()), [getClipBounds](http://docs.google.com/java/awt/Graphics.html#getClipBounds(java.awt.Rectangle)), [getClipRect](http://docs.google.com/java/awt/Graphics.html#getClipRect()), [hitClip](http://docs.google.com/java/awt/Graphics.html#hitClip(int,%20int,%20int,%20int)), [toString](http://docs.google.com/java/awt/Graphics.html#toString()) |

| **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)), [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** |
| --- |

### LOG\_OPTION

public static final int **LOG\_OPTION**

Log graphics operations.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.DebugGraphics.LOG_OPTION)

### FLASH\_OPTION

public static final int **FLASH\_OPTION**

Flash graphics operations.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.DebugGraphics.FLASH_OPTION)

### BUFFERED\_OPTION

public static final int **BUFFERED\_OPTION**

Show buffered operations in a separate Frame.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.DebugGraphics.BUFFERED_OPTION)

### NONE\_OPTION

public static final int **NONE\_OPTION**

Don't debug graphics operations.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.DebugGraphics.NONE_OPTION)

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

### DebugGraphics

public **DebugGraphics**()

Constructs a new debug graphics context that supports slowed down drawing.

### DebugGraphics

public **DebugGraphics**([Graphics](http://docs.google.com/java/awt/Graphics.html) graphics,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) component)

Constructs a debug graphics context from an existing graphics context that slows down drawing for the specified component.

**Parameters:**graphics - the Graphics context to slow downcomponent - the JComponent to draw slowly

### DebugGraphics

public **DebugGraphics**([Graphics](http://docs.google.com/java/awt/Graphics.html) graphics)

Constructs a debug graphics context from an existing graphics context that supports slowed down drawing.

**Parameters:**graphics - the Graphics context to slow down

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

### create

public [Graphics](http://docs.google.com/java/awt/Graphics.html) **create**()

Overrides Graphics.create to return a DebugGraphics object.

**Specified by:**[create](http://docs.google.com/java/awt/Graphics.html#create()) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Returns:**a new graphics context that is a copy of this graphics context.

### create

public [Graphics](http://docs.google.com/java/awt/Graphics.html) **create**(int x,  
 int y,  
 int width,  
 int height)

Overrides Graphics.create to return a DebugGraphics object.

**Overrides:**[create](http://docs.google.com/java/awt/Graphics.html#create(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate.y - the *y* coordinate.width - the width of the clipping rectangle.height - the height of the clipping rectangle. **Returns:**a new graphics context.**See Also:**[Graphics.translate(int, int)](http://docs.google.com/java/awt/Graphics.html#translate(int,%20int)), [Graphics.clipRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#clipRect(int,%20int,%20int,%20int))

### setFlashColor

public static void **setFlashColor**([Color](http://docs.google.com/java/awt/Color.html) flashColor)

Sets the Color used to flash drawing operations.

### flashColor

public static [Color](http://docs.google.com/java/awt/Color.html) **flashColor**()

Returns the Color used to flash drawing operations.

**See Also:**[setFlashColor(java.awt.Color)](http://docs.google.com/javax/swing/DebugGraphics.html#setFlashColor(java.awt.Color))

### setFlashTime

public static void **setFlashTime**(int flashTime)

Sets the time delay of drawing operation flashing.

### flashTime

public static int **flashTime**()

Returns the time delay of drawing operation flashing.

**See Also:**[setFlashTime(int)](http://docs.google.com/javax/swing/DebugGraphics.html#setFlashTime(int))

### setFlashCount

public static void **setFlashCount**(int flashCount)

Sets the number of times that drawing operations will flash.

### flashCount

public static int **flashCount**()

Returns the number of times that drawing operations will flash.

**See Also:**[setFlashCount(int)](http://docs.google.com/javax/swing/DebugGraphics.html#setFlashCount(int))

### setLogStream

public static void **setLogStream**([PrintStream](http://docs.google.com/java/io/PrintStream.html) stream)

Sets the stream to which the DebugGraphics logs drawing operations.

### logStream

public static [PrintStream](http://docs.google.com/java/io/PrintStream.html) **logStream**()

Returns the stream to which the DebugGraphics logs drawing operations.

**See Also:**[setLogStream(java.io.PrintStream)](http://docs.google.com/javax/swing/DebugGraphics.html#setLogStream(java.io.PrintStream))

### setFont

public void **setFont**([Font](http://docs.google.com/java/awt/Font.html) aFont)

Sets the Font used for text drawing operations.

**Specified by:**[setFont](http://docs.google.com/java/awt/Graphics.html#setFont(java.awt.Font)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**aFont - the font.**See Also:**[Graphics.getFont()](http://docs.google.com/java/awt/Graphics.html#getFont()), [Graphics.drawString(java.lang.String, int, int)](http://docs.google.com/java/awt/Graphics.html#drawString(java.lang.String,%20int,%20int)), [Graphics.drawBytes(byte[], int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawBytes(byte%5B%5D,%20int,%20int,%20int,%20int)), [Graphics.drawChars(char[], int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawChars(char%5B%5D,%20int,%20int,%20int,%20int))

### getFont

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

Returns the Font used for text drawing operations.

**Specified by:**[getFont](http://docs.google.com/java/awt/Graphics.html#getFont()) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Returns:**this graphics context's current font.**See Also:**[setFont(java.awt.Font)](http://docs.google.com/javax/swing/DebugGraphics.html#setFont(java.awt.Font))

### setColor

public void **setColor**([Color](http://docs.google.com/java/awt/Color.html) aColor)

Sets the color to be used for drawing and filling lines and shapes.

**Specified by:**[setColor](http://docs.google.com/java/awt/Graphics.html#setColor(java.awt.Color)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**aColor - the new rendering color.**See Also:**[Color](http://docs.google.com/java/awt/Color.html), [Graphics.getColor()](http://docs.google.com/java/awt/Graphics.html#getColor())

### getColor

public [Color](http://docs.google.com/java/awt/Color.html) **getColor**()

Returns the Color used for text drawing operations.

**Specified by:**[getColor](http://docs.google.com/java/awt/Graphics.html#getColor()) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Returns:**this graphics context's current color.**See Also:**[setColor(java.awt.Color)](http://docs.google.com/javax/swing/DebugGraphics.html#setColor(java.awt.Color))

### getFontMetrics

public [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) **getFontMetrics**()

Overrides Graphics.getFontMetrics.

**Overrides:**[getFontMetrics](http://docs.google.com/java/awt/Graphics.html#getFontMetrics()) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Returns:**the font metrics of this graphics context's current font.**See Also:**[Graphics.getFont()](http://docs.google.com/java/awt/Graphics.html#getFont()), [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html), [Graphics.getFontMetrics(Font)](http://docs.google.com/java/awt/Graphics.html#getFontMetrics(java.awt.Font))

### getFontMetrics

public [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) **getFontMetrics**([Font](http://docs.google.com/java/awt/Font.html) f)

Overrides Graphics.getFontMetrics.

**Specified by:**[getFontMetrics](http://docs.google.com/java/awt/Graphics.html#getFontMetrics(java.awt.Font)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**f - the specified font **Returns:**the font metrics for the specified font.**See Also:**[Graphics.getFont()](http://docs.google.com/java/awt/Graphics.html#getFont()), [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html), [Graphics.getFontMetrics()](http://docs.google.com/java/awt/Graphics.html#getFontMetrics())

### translate

public void **translate**(int x,  
 int y)

Overrides Graphics.translate.

**Specified by:**[translate](http://docs.google.com/java/awt/Graphics.html#translate(int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate.y - the *y* coordinate.

### setPaintMode

public void **setPaintMode**()

Overrides Graphics.setPaintMode.

**Specified by:**[setPaintMode](http://docs.google.com/java/awt/Graphics.html#setPaintMode()) in class [Graphics](http://docs.google.com/java/awt/Graphics.html)

### setXORMode

public void **setXORMode**([Color](http://docs.google.com/java/awt/Color.html) aColor)

Overrides Graphics.setXORMode.

**Specified by:**[setXORMode](http://docs.google.com/java/awt/Graphics.html#setXORMode(java.awt.Color)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**aColor - the XOR alternation color

### getClipBounds

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getClipBounds**()

Overrides Graphics.getClipBounds.

**Specified by:**[getClipBounds](http://docs.google.com/java/awt/Graphics.html#getClipBounds()) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Returns:**the bounding rectangle of the current clipping area, or null if no clip is set.**See Also:**[Graphics.getClip()](http://docs.google.com/java/awt/Graphics.html#getClip()), [Graphics.clipRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#clipRect(int,%20int,%20int,%20int)), [Graphics.setClip(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#setClip(int,%20int,%20int,%20int)), [Graphics.setClip(Shape)](http://docs.google.com/java/awt/Graphics.html#setClip(java.awt.Shape))

### clipRect

public void **clipRect**(int x,  
 int y,  
 int width,  
 int height)

Overrides Graphics.clipRect.

**Specified by:**[clipRect](http://docs.google.com/java/awt/Graphics.html#clipRect(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the x coordinate of the rectangle to intersect the clip withy - the y coordinate of the rectangle to intersect the clip withwidth - the width of the rectangle to intersect the clip withheight - the height of the rectangle to intersect the clip with**See Also:**[Graphics.setClip(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#setClip(int,%20int,%20int,%20int)), [Graphics.setClip(Shape)](http://docs.google.com/java/awt/Graphics.html#setClip(java.awt.Shape))

### setClip

public void **setClip**(int x,  
 int y,  
 int width,  
 int height)

Overrides Graphics.setClip.

**Specified by:**[setClip](http://docs.google.com/java/awt/Graphics.html#setClip(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the new clip rectangle.y - the *y* coordinate of the new clip rectangle.width - the width of the new clip rectangle.height - the height of the new clip rectangle.**See Also:**[Graphics.clipRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#clipRect(int,%20int,%20int,%20int)), [Graphics.setClip(Shape)](http://docs.google.com/java/awt/Graphics.html#setClip(java.awt.Shape)), [Graphics.getClip()](http://docs.google.com/java/awt/Graphics.html#getClip())

### getClip

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

Overrides Graphics.getClip.

**Specified by:**[getClip](http://docs.google.com/java/awt/Graphics.html#getClip()) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Returns:**a Shape object representing the current clipping area, or null if no clip is set.**See Also:**[Graphics.getClipBounds()](http://docs.google.com/java/awt/Graphics.html#getClipBounds()), [Graphics.clipRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#clipRect(int,%20int,%20int,%20int)), [Graphics.setClip(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#setClip(int,%20int,%20int,%20int)), [Graphics.setClip(Shape)](http://docs.google.com/java/awt/Graphics.html#setClip(java.awt.Shape))

### setClip

public void **setClip**([Shape](http://docs.google.com/java/awt/Shape.html) clip)

Overrides Graphics.setClip.

**Specified by:**[setClip](http://docs.google.com/java/awt/Graphics.html#setClip(java.awt.Shape)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**clip - the Shape to use to set the clip**See Also:**[Graphics.getClip()](http://docs.google.com/java/awt/Graphics.html#getClip()), [Graphics.clipRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#clipRect(int,%20int,%20int,%20int)), [Graphics.setClip(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#setClip(int,%20int,%20int,%20int))

### drawRect

public void **drawRect**(int x,  
 int y,  
 int width,  
 int height)

Overrides Graphics.drawRect.

**Overrides:**[drawRect](http://docs.google.com/java/awt/Graphics.html#drawRect(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the rectangle to be drawn.y - the *y* coordinate of the rectangle to be drawn.width - the width of the rectangle to be drawn.height - the height of the rectangle to be drawn.**See Also:**[Graphics.fillRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#fillRect(int,%20int,%20int,%20int)), [Graphics.clearRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#clearRect(int,%20int,%20int,%20int))

### fillRect

public void **fillRect**(int x,  
 int y,  
 int width,  
 int height)

Overrides Graphics.fillRect.

**Specified by:**[fillRect](http://docs.google.com/java/awt/Graphics.html#fillRect(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the rectangle to be filled.y - the *y* coordinate of the rectangle to be filled.width - the width of the rectangle to be filled.height - the height of the rectangle to be filled.**See Also:**[Graphics.clearRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#clearRect(int,%20int,%20int,%20int)), [Graphics.drawRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawRect(int,%20int,%20int,%20int))

### clearRect

public void **clearRect**(int x,  
 int y,  
 int width,  
 int height)

Overrides Graphics.clearRect.

**Specified by:**[clearRect](http://docs.google.com/java/awt/Graphics.html#clearRect(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the rectangle to clear.y - the *y* coordinate of the rectangle to clear.width - the width of the rectangle to clear.height - the height of the rectangle to clear.**See Also:**[Graphics.fillRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#fillRect(int,%20int,%20int,%20int)), [Graphics.drawRect(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawRect(int,%20int,%20int,%20int)), [Graphics.setColor(java.awt.Color)](http://docs.google.com/java/awt/Graphics.html#setColor(java.awt.Color)), [Graphics.setPaintMode()](http://docs.google.com/java/awt/Graphics.html#setPaintMode()), [Graphics.setXORMode(java.awt.Color)](http://docs.google.com/java/awt/Graphics.html#setXORMode(java.awt.Color))

### drawRoundRect

public void **drawRoundRect**(int x,  
 int y,  
 int width,  
 int height,  
 int arcWidth,  
 int arcHeight)

Overrides Graphics.drawRoundRect.

**Specified by:**[drawRoundRect](http://docs.google.com/java/awt/Graphics.html#drawRoundRect(int,%20int,%20int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the rectangle to be drawn.y - the *y* coordinate of the rectangle to be drawn.width - the width of the rectangle to be drawn.height - the height of the rectangle to be drawn.arcWidth - the horizontal diameter of the arc at the four corners.arcHeight - the vertical diameter of the arc at the four corners.**See Also:**[Graphics.fillRoundRect(int, int, int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#fillRoundRect(int,%20int,%20int,%20int,%20int,%20int))

### fillRoundRect

public void **fillRoundRect**(int x,  
 int y,  
 int width,  
 int height,  
 int arcWidth,  
 int arcHeight)

Overrides Graphics.fillRoundRect.

**Specified by:**[fillRoundRect](http://docs.google.com/java/awt/Graphics.html#fillRoundRect(int,%20int,%20int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the rectangle to be filled.y - the *y* coordinate of the rectangle to be filled.width - the width of the rectangle to be filled.height - the height of the rectangle to be filled.arcWidth - the horizontal diameter of the arc at the four corners.arcHeight - the vertical diameter of the arc at the four corners.**See Also:**[Graphics.drawRoundRect(int, int, int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawRoundRect(int,%20int,%20int,%20int,%20int,%20int))

### drawLine

public void **drawLine**(int x1,  
 int y1,  
 int x2,  
 int y2)

Overrides Graphics.drawLine.

**Specified by:**[drawLine](http://docs.google.com/java/awt/Graphics.html#drawLine(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x1 - the first point's *x* coordinate.y1 - the first point's *y* coordinate.x2 - the second point's *x* coordinate.y2 - the second point's *y* coordinate.

### draw3DRect

public void **draw3DRect**(int x,  
 int y,  
 int width,  
 int height,  
 boolean raised)

Overrides Graphics.draw3DRect.

**Overrides:**[draw3DRect](http://docs.google.com/java/awt/Graphics.html#draw3DRect(int,%20int,%20int,%20int,%20boolean)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the rectangle to be drawn.y - the *y* coordinate of the rectangle to be drawn.width - the width of the rectangle to be drawn.height - the height of the rectangle to be drawn.raised - a boolean that determines whether the rectangle appears to be raised above the surface or sunk into the surface.**See Also:**[Graphics.fill3DRect(int, int, int, int, boolean)](http://docs.google.com/java/awt/Graphics.html#fill3DRect(int,%20int,%20int,%20int,%20boolean))

### fill3DRect

public void **fill3DRect**(int x,  
 int y,  
 int width,  
 int height,  
 boolean raised)

Overrides Graphics.fill3DRect.

**Overrides:**[fill3DRect](http://docs.google.com/java/awt/Graphics.html#fill3DRect(int,%20int,%20int,%20int,%20boolean)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the rectangle to be filled.y - the *y* coordinate of the rectangle to be filled.width - the width of the rectangle to be filled.height - the height of the rectangle to be filled.raised - a boolean value that determines whether the rectangle appears to be raised above the surface or etched into the surface.**See Also:**[Graphics.draw3DRect(int, int, int, int, boolean)](http://docs.google.com/java/awt/Graphics.html#draw3DRect(int,%20int,%20int,%20int,%20boolean))

### drawOval

public void **drawOval**(int x,  
 int y,  
 int width,  
 int height)

Overrides Graphics.drawOval.

**Specified by:**[drawOval](http://docs.google.com/java/awt/Graphics.html#drawOval(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the upper left corner of the oval to be drawn.y - the *y* coordinate of the upper left corner of the oval to be drawn.width - the width of the oval to be drawn.height - the height of the oval to be drawn.**See Also:**[Graphics.fillOval(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#fillOval(int,%20int,%20int,%20int))

### fillOval

public void **fillOval**(int x,  
 int y,  
 int width,  
 int height)

Overrides Graphics.fillOval.

**Specified by:**[fillOval](http://docs.google.com/java/awt/Graphics.html#fillOval(int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the upper left corner of the oval to be filled.y - the *y* coordinate of the upper left corner of the oval to be filled.width - the width of the oval to be filled.height - the height of the oval to be filled.**See Also:**[Graphics.drawOval(int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawOval(int,%20int,%20int,%20int))

### drawArc

public void **drawArc**(int x,  
 int y,  
 int width,  
 int height,  
 int startAngle,  
 int arcAngle)

Overrides Graphics.drawArc.

**Specified by:**[drawArc](http://docs.google.com/java/awt/Graphics.html#drawArc(int,%20int,%20int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the upper-left corner of the arc to be drawn.y - the *y* coordinate of the upper-left corner of the arc to be drawn.width - the width of the arc to be drawn.height - the height of the arc to be drawn.startAngle - the beginning angle.arcAngle - the angular extent of the arc, relative to the start angle.**See Also:**[Graphics.fillArc(int, int, int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#fillArc(int,%20int,%20int,%20int,%20int,%20int))

### fillArc

public void **fillArc**(int x,  
 int y,  
 int width,  
 int height,  
 int startAngle,  
 int arcAngle)

Overrides Graphics.fillArc.

**Specified by:**[fillArc](http://docs.google.com/java/awt/Graphics.html#fillArc(int,%20int,%20int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the upper-left corner of the arc to be filled.y - the *y* coordinate of the upper-left corner of the arc to be filled.width - the width of the arc to be filled.height - the height of the arc to be filled.startAngle - the beginning angle.arcAngle - the angular extent of the arc, relative to the start angle.**See Also:**[Graphics.drawArc(int, int, int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawArc(int,%20int,%20int,%20int,%20int,%20int))

### drawPolyline

public void **drawPolyline**(int[] xPoints,  
 int[] yPoints,  
 int nPoints)

Overrides Graphics.drawPolyline.

**Specified by:**[drawPolyline](http://docs.google.com/java/awt/Graphics.html#drawPolyline(int%5B%5D,%20int%5B%5D,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**xPoints - an array of *x* pointsyPoints - an array of *y* pointsnPoints - the total number of points**See Also:**[Graphics.drawPolygon(int[], int[], int)](http://docs.google.com/java/awt/Graphics.html#drawPolygon(int%5B%5D,%20int%5B%5D,%20int))

### drawPolygon

public void **drawPolygon**(int[] xPoints,  
 int[] yPoints,  
 int nPoints)

Overrides Graphics.drawPolygon.

**Specified by:**[drawPolygon](http://docs.google.com/java/awt/Graphics.html#drawPolygon(int%5B%5D,%20int%5B%5D,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**xPoints - a an array of x coordinates.yPoints - a an array of y coordinates.nPoints - a the total number of points.**See Also:**[Graphics.fillPolygon(int[], int[], int)](http://docs.google.com/java/awt/Graphics.html#fillPolygon(int%5B%5D,%20int%5B%5D,%20int)), [Graphics.drawPolyline(int[], int[], int)](http://docs.google.com/java/awt/Graphics.html#drawPolyline(int%5B%5D,%20int%5B%5D,%20int))

### fillPolygon

public void **fillPolygon**(int[] xPoints,  
 int[] yPoints,  
 int nPoints)

Overrides Graphics.fillPolygon.

**Specified by:**[fillPolygon](http://docs.google.com/java/awt/Graphics.html#fillPolygon(int%5B%5D,%20int%5B%5D,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**xPoints - a an array of x coordinates.yPoints - a an array of y coordinates.nPoints - a the total number of points.**See Also:**[Graphics.drawPolygon(int[], int[], int)](http://docs.google.com/java/awt/Graphics.html#drawPolygon(int%5B%5D,%20int%5B%5D,%20int))

### drawString

public void **drawString**([String](http://docs.google.com/java/lang/String.html) aString,  
 int x,  
 int y)

Overrides Graphics.drawString.

**Specified by:**[drawString](http://docs.google.com/java/awt/Graphics.html#drawString(java.lang.String,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**aString - the string to be drawn.x - the *x* coordinate.y - the *y* coordinate.**See Also:**[Graphics.drawBytes(byte[], int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawBytes(byte%5B%5D,%20int,%20int,%20int,%20int)), [Graphics.drawChars(char[], int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawChars(char%5B%5D,%20int,%20int,%20int,%20int))

### drawString

public void **drawString**([AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) iterator,  
 int x,  
 int y)

Overrides Graphics.drawString.

**Specified by:**[drawString](http://docs.google.com/java/awt/Graphics.html#drawString(java.text.AttributedCharacterIterator,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**iterator - the iterator whose text is to be drawnx - the *x* coordinate.y - the *y* coordinate.**See Also:**[Graphics.drawBytes(byte[], int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawBytes(byte%5B%5D,%20int,%20int,%20int,%20int)), [Graphics.drawChars(char[], int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawChars(char%5B%5D,%20int,%20int,%20int,%20int))

### drawBytes

public void **drawBytes**(byte[] data,  
 int offset,  
 int length,  
 int x,  
 int y)

Overrides Graphics.drawBytes.

**Overrides:**[drawBytes](http://docs.google.com/java/awt/Graphics.html#drawBytes(byte%5B%5D,%20int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**data - the data to be drawnoffset - the start offset in the datalength - the number of bytes that are drawnx - the *x* coordinate of the baseline of the texty - the *y* coordinate of the baseline of the text**See Also:**[Graphics.drawChars(char[], int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawChars(char%5B%5D,%20int,%20int,%20int,%20int)), [Graphics.drawString(java.lang.String, int, int)](http://docs.google.com/java/awt/Graphics.html#drawString(java.lang.String,%20int,%20int))

### drawChars

public void **drawChars**(char[] data,  
 int offset,  
 int length,  
 int x,  
 int y)

Overrides Graphics.drawChars.

**Overrides:**[drawChars](http://docs.google.com/java/awt/Graphics.html#drawChars(char%5B%5D,%20int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**data - the array of characters to be drawnoffset - the start offset in the datalength - the number of characters to be drawnx - the *x* coordinate of the baseline of the texty - the *y* coordinate of the baseline of the text**See Also:**[Graphics.drawBytes(byte[], int, int, int, int)](http://docs.google.com/java/awt/Graphics.html#drawBytes(byte%5B%5D,%20int,%20int,%20int,%20int)), [Graphics.drawString(java.lang.String, int, int)](http://docs.google.com/java/awt/Graphics.html#drawString(java.lang.String,%20int,%20int))

### drawImage

public boolean **drawImage**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int x,  
 int y,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Overrides Graphics.drawImage.

**Specified by:**[drawImage](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**img - the specified image to be drawn. This method does nothing if img is null.x - the *x* coordinate.y - the *y* coordinate.observer - object to be notified as more of the image is converted. **Returns:**false if the image pixels are still changing; true otherwise.**See Also:**[Image](http://docs.google.com/java/awt/Image.html), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [ImageObserver.imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))

### drawImage

public boolean **drawImage**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int x,  
 int y,  
 int width,  
 int height,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Overrides Graphics.drawImage.

**Specified by:**[drawImage](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20java.awt.image.ImageObserver)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**img - the specified image to be drawn. This method does nothing if img is null.x - the *x* coordinate.y - the *y* coordinate.width - the width of the rectangle.height - the height of the rectangle.observer - object to be notified as more of the image is converted. **Returns:**false if the image pixels are still changing; true otherwise.**See Also:**[Image](http://docs.google.com/java/awt/Image.html), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [ImageObserver.imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))

### drawImage

public boolean **drawImage**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int x,  
 int y,  
 [Color](http://docs.google.com/java/awt/Color.html) bgcolor,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Overrides Graphics.drawImage.

**Specified by:**[drawImage](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.Color,%20java.awt.image.ImageObserver)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**img - the specified image to be drawn. This method does nothing if img is null.x - the *x* coordinate.y - the *y* coordinate.bgcolor - the background color to paint under the non-opaque portions of the image.observer - object to be notified as more of the image is converted. **Returns:**false if the image pixels are still changing; true otherwise.**See Also:**[Image](http://docs.google.com/java/awt/Image.html), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [ImageObserver.imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))

### drawImage

public boolean **drawImage**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int x,  
 int y,  
 int width,  
 int height,  
 [Color](http://docs.google.com/java/awt/Color.html) bgcolor,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Overrides Graphics.drawImage.

**Specified by:**[drawImage](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20java.awt.Color,%20java.awt.image.ImageObserver)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**img - the specified image to be drawn. This method does nothing if img is null.x - the *x* coordinate.y - the *y* coordinate.width - the width of the rectangle.height - the height of the rectangle.bgcolor - the background color to paint under the non-opaque portions of the image.observer - object to be notified as more of the image is converted. **Returns:**false if the image pixels are still changing; true otherwise.**See Also:**[Image](http://docs.google.com/java/awt/Image.html), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [ImageObserver.imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))

### drawImage

public boolean **drawImage**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int dx1,  
 int dy1,  
 int dx2,  
 int dy2,  
 int sx1,  
 int sy1,  
 int sx2,  
 int sy2,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Overrides Graphics.drawImage.

**Specified by:**[drawImage](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20java.awt.image.ImageObserver)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**img - the specified image to be drawn. This method does nothing if img is null.dx1 - the *x* coordinate of the first corner of the destination rectangle.dy1 - the *y* coordinate of the first corner of the destination rectangle.dx2 - the *x* coordinate of the second corner of the destination rectangle.dy2 - the *y* coordinate of the second corner of the destination rectangle.sx1 - the *x* coordinate of the first corner of the source rectangle.sy1 - the *y* coordinate of the first corner of the source rectangle.sx2 - the *x* coordinate of the second corner of the source rectangle.sy2 - the *y* coordinate of the second corner of the source rectangle.observer - object to be notified as more of the image is scaled and converted. **Returns:**false if the image pixels are still changing; true otherwise.**See Also:**[Image](http://docs.google.com/java/awt/Image.html), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [ImageObserver.imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))

### drawImage

public boolean **drawImage**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int dx1,  
 int dy1,  
 int dx2,  
 int dy2,  
 int sx1,  
 int sy1,  
 int sx2,  
 int sy2,  
 [Color](http://docs.google.com/java/awt/Color.html) bgcolor,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Overrides Graphics.drawImage.

**Specified by:**[drawImage](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20int,%20java.awt.Color,%20java.awt.image.ImageObserver)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**img - the specified image to be drawn. This method does nothing if img is null.dx1 - the *x* coordinate of the first corner of the destination rectangle.dy1 - the *y* coordinate of the first corner of the destination rectangle.dx2 - the *x* coordinate of the second corner of the destination rectangle.dy2 - the *y* coordinate of the second corner of the destination rectangle.sx1 - the *x* coordinate of the first corner of the source rectangle.sy1 - the *y* coordinate of the first corner of the source rectangle.sx2 - the *x* coordinate of the second corner of the source rectangle.sy2 - the *y* coordinate of the second corner of the source rectangle.bgcolor - the background color to paint under the non-opaque portions of the image.observer - object to be notified as more of the image is scaled and converted. **Returns:**false if the image pixels are still changing; true otherwise.**See Also:**[Image](http://docs.google.com/java/awt/Image.html), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [ImageObserver.imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))

### copyArea

public void **copyArea**(int x,  
 int y,  
 int width,  
 int height,  
 int destX,  
 int destY)

Overrides Graphics.copyArea.

**Specified by:**[copyArea](http://docs.google.com/java/awt/Graphics.html#copyArea(int,%20int,%20int,%20int,%20int,%20int)) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **Parameters:**x - the *x* coordinate of the source rectangle.y - the *y* coordinate of the source rectangle.width - the width of the source rectangle.height - the height of the source rectangle.destX - the horizontal distance to copy the pixels.destY - the vertical distance to copy the pixels.

### dispose

public void **dispose**()

Overrides Graphics.dispose.

**Specified by:**[dispose](http://docs.google.com/java/awt/Graphics.html#dispose()) in class [Graphics](http://docs.google.com/java/awt/Graphics.html) **See Also:**[Graphics.finalize()](http://docs.google.com/java/awt/Graphics.html#finalize()), [Component.paint(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics)), [Component.update(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics)), [Component.getGraphics()](http://docs.google.com/java/awt/Component.html#getGraphics()), [Graphics.create()](http://docs.google.com/java/awt/Graphics.html#create())

### isDrawingBuffer

public boolean **isDrawingBuffer**()

Returns the drawingBuffer value.

**Returns:**true if this object is drawing from a Buffer

### setDebugOptions

public void **setDebugOptions**(int options)

Enables/disables diagnostic information about every graphics operation. The value of **options** indicates how this information should be displayed. LOG\_OPTION causes a text message to be printed. FLASH\_OPTION causes the drawing to flash several times. BUFFERED\_OPTION creates a new Frame that shows each operation on an offscreen buffer. The value of **options** is bitwise OR'd into the current value. To disable debugging use NONE\_OPTION.

### getDebugOptions

public int **getDebugOptions**()

Returns the current debugging options for this DebugGraphics.

**See Also:**[setDebugOptions(int)](http://docs.google.com/javax/swing/DebugGraphics.html#setDebugOptions(int))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DebugGraphics.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/ComponentInputMap.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/DefaultBoundedRangeModel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/DebugGraphics.html)    [**NO FRAMES**](http://docs.google.com/DebugGraphics.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#lnxbz9) | [METHOD](#2jxsxqh) |

[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).