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

## **java.awt**

Class GraphicsEnvironment

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

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

The GraphicsEnvironment class describes the collection of [GraphicsDevice](http://docs.google.com/java/awt/GraphicsDevice.html) objects and [Font](http://docs.google.com/java/awt/Font.html) objects available to a Java(tm) application on a particular platform. The resources in this GraphicsEnvironment might be local or on a remote machine. GraphicsDevice objects can be screens, printers or image buffers and are the destination of [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) drawing methods. Each GraphicsDevice has a number of [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) objects associated with it. These objects specify the different configurations in which the GraphicsDevice can be used.

**See Also:**[GraphicsDevice](http://docs.google.com/java/awt/GraphicsDevice.html), [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**GraphicsEnvironment**](http://docs.google.com/java/awt/GraphicsEnvironment.html#GraphicsEnvironment())()            This is an abstract class and cannot be instantiated directly. |

| **Method Summary** | |
| --- | --- |
| abstract  [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) | [**createGraphics**](http://docs.google.com/java/awt/GraphicsEnvironment.html#createGraphics(java.awt.image.BufferedImage))([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) img)            Returns a Graphics2D object for rendering into the specified [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html). |
| abstract  [Font](http://docs.google.com/java/awt/Font.html)[] | [**getAllFonts**](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAllFonts())()            Returns an array containing a one-point size instance of all fonts available in this GraphicsEnvironment. |
| abstract  [String](http://docs.google.com/java/lang/String.html)[] | [**getAvailableFontFamilyNames**](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAvailableFontFamilyNames())()            Returns an array containing the names of all font families in this GraphicsEnvironment localized for the default locale, as returned by Locale.getDefault(). |
| abstract  [String](http://docs.google.com/java/lang/String.html)[] | [**getAvailableFontFamilyNames**](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAvailableFontFamilyNames(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) l)            Returns an array containing the names of all font families in this GraphicsEnvironment localized for the specified locale. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getCenterPoint**](http://docs.google.com/java/awt/GraphicsEnvironment.html#getCenterPoint())()            Returns the Point where Windows should be centered. |
| abstract  [GraphicsDevice](http://docs.google.com/java/awt/GraphicsDevice.html) | [**getDefaultScreenDevice**](http://docs.google.com/java/awt/GraphicsEnvironment.html#getDefaultScreenDevice())()            Returns the default screen GraphicsDevice. |
| static [GraphicsEnvironment](http://docs.google.com/java/awt/GraphicsEnvironment.html) | [**getLocalGraphicsEnvironment**](http://docs.google.com/java/awt/GraphicsEnvironment.html#getLocalGraphicsEnvironment())()            Returns the local GraphicsEnvironment. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getMaximumWindowBounds**](http://docs.google.com/java/awt/GraphicsEnvironment.html#getMaximumWindowBounds())()            Returns the maximum bounds for centered Windows. |
| abstract  [GraphicsDevice](http://docs.google.com/java/awt/GraphicsDevice.html)[] | [**getScreenDevices**](http://docs.google.com/java/awt/GraphicsEnvironment.html#getScreenDevices())()            Returns an array of all of the screen GraphicsDevice objects. |
| static boolean | [**isHeadless**](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())()            Tests whether or not a display, keyboard, and mouse can be supported in this environment. |
| boolean | [**isHeadlessInstance**](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadlessInstance())()            Returns whether or not a display, keyboard, and mouse can be supported in this graphics environment. |
| void | [**preferLocaleFonts**](http://docs.google.com/java/awt/GraphicsEnvironment.html#preferLocaleFonts())()            Indicates a preference for locale-specific fonts in the mapping of logical fonts to physical fonts. |
| void | [**preferProportionalFonts**](http://docs.google.com/java/awt/GraphicsEnvironment.html#preferProportionalFonts())()            Indicates a preference for proportional over non-proportional (e.g. |
| boolean | [**registerFont**](http://docs.google.com/java/awt/GraphicsEnvironment.html#registerFont(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) font)            Registers a */created* Fontin this GraphicsEnvironment. |

| **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** |
| --- |

### GraphicsEnvironment

protected **GraphicsEnvironment**()

This is an abstract class and cannot be instantiated directly. Instances must be obtained from a suitable factory or query method.

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

### getLocalGraphicsEnvironment

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

Returns the local GraphicsEnvironment.

**Returns:**the local GraphicsEnvironment

### isHeadless

public static boolean **isHeadless**()

Tests whether or not a display, keyboard, and mouse can be supported in this environment. If this method returns true, a HeadlessException is thrown from areas of the Toolkit and GraphicsEnvironment that are dependent on a display, keyboard, or mouse.

**Returns:**true if this environment cannot support a display, keyboard, and mouse; false otherwise**Since:** 1.4 **See Also:**[HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

### isHeadlessInstance

public boolean **isHeadlessInstance**()

Returns whether or not a display, keyboard, and mouse can be supported in this graphics environment. If this returns true, HeadlessException will be thrown from areas of the graphics environment that are dependent on a display, keyboard, or mouse.

**Returns:**true if a display, keyboard, and mouse can be supported in this environment; false otherwise**Since:** 1.4 **See Also:**[HeadlessException](http://docs.google.com/java/awt/HeadlessException.html), [isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### getScreenDevices

public abstract [GraphicsDevice](http://docs.google.com/java/awt/GraphicsDevice.html)[] **getScreenDevices**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Returns an array of all of the screen GraphicsDevice objects.

**Returns:**an array containing all the GraphicsDevice objects that represent screen devices **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if isHeadless() returns true**See Also:**[isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### getDefaultScreenDevice

public abstract [GraphicsDevice](http://docs.google.com/java/awt/GraphicsDevice.html) **getDefaultScreenDevice**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Returns the default screen GraphicsDevice.

**Returns:**the GraphicsDevice that represents the default screen device **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if isHeadless() returns true**See Also:**[isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### createGraphics

public abstract [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) **createGraphics**([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) img)

Returns a Graphics2D object for rendering into the specified [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html).

**Parameters:**img - the specified BufferedImage **Returns:**a Graphics2D to be used for rendering into the specified BufferedImage **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if img is null

### getAllFonts

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

Returns an array containing a one-point size instance of all fonts available in this GraphicsEnvironment. Typical usage would be to allow a user to select a particular font. Then, the application can size the font and set various font attributes by calling the deriveFont method on the choosen instance.

This method provides for the application the most precise control over which Font instance is used to render text. If a font in this GraphicsEnvironment has multiple programmable variations, only one instance of that Font is returned in the array, and other variations must be derived by the application.

If a font in this environment has multiple programmable variations, such as Multiple-Master fonts, only one instance of that font is returned in the Font array. The other variations must be derived by the application.

**Returns:**an array of Font objects**Since:** 1.2 **See Also:**[getAvailableFontFamilyNames()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAvailableFontFamilyNames()), [Font](http://docs.google.com/java/awt/Font.html), [Font.deriveFont(int, float)](http://docs.google.com/java/awt/Font.html#deriveFont(int,%20float)), [Font.getFontName()](http://docs.google.com/java/awt/Font.html#getFontName())

### getAvailableFontFamilyNames

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

Returns an array containing the names of all font families in this GraphicsEnvironment localized for the default locale, as returned by Locale.getDefault().

Typical usage would be for presentation to a user for selection of a particular family name. An application can then specify this name when creating a font, in conjunction with a style, such as bold or italic, giving the font system flexibility in choosing its own best match among multiple fonts in the same font family.

**Returns:**an array of String containing font family names localized for the default locale, or a suitable alternative name if no name exists for this locale.**Since:** 1.2 **See Also:**[getAllFonts()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAllFonts()), [Font](http://docs.google.com/java/awt/Font.html), [Font.getFamily()](http://docs.google.com/java/awt/Font.html#getFamily())

### getAvailableFontFamilyNames

public abstract [String](http://docs.google.com/java/lang/String.html)[] **getAvailableFontFamilyNames**([Locale](http://docs.google.com/java/util/Locale.html) l)

Returns an array containing the names of all font families in this GraphicsEnvironment localized for the specified locale.

Typical usage would be for presentation to a user for selection of a particular family name. An application can then specify this name when creating a font, in conjunction with a style, such as bold or italic, giving the font system flexibility in choosing its own best match among multiple fonts in the same font family.

**Parameters:**l - a [Locale](http://docs.google.com/java/util/Locale.html) object that represents a particular geographical, political, or cultural region. Specifying null is equivalent to specifying Locale.getDefault(). **Returns:**an array of String containing font family names localized for the specified Locale, or a suitable alternative name if no name exists for the specified locale.**Since:** 1.2 **See Also:**[getAllFonts()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAllFonts()), [Font](http://docs.google.com/java/awt/Font.html), [Font.getFamily()](http://docs.google.com/java/awt/Font.html#getFamily())

### registerFont

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

Registers a */created* Fontin this GraphicsEnvironment. A created font is one that was returned from calling [Font.createFont(int, java.io.InputStream)](http://docs.google.com/java/awt/Font.html#createFont(int,%20java.io.InputStream)), or derived from a created font by calling [Font.deriveFont(int, float)](http://docs.google.com/java/awt/Font.html#deriveFont(int,%20float)). After calling this method for such a font, it is available to be used in constructing new Fonts by name or family name, and is enumerated by [getAvailableFontFamilyNames()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAvailableFontFamilyNames()) and [getAllFonts()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getAllFonts()) within the execution context of this application or applet. This means applets cannot register fonts in a way that they are visible to other applets.

Reasons that this method might not register the font and therefore return false are :

* The font is not a */created* Font.
* The font conflicts with a non-created Font already in this GraphicsEnvironment. For example if the name is that of a system font, or a logical font as described in the documentation of the [Font](http://docs.google.com/java/awt/Font.html) class. It is implementation dependent whether a font may also conflict if it has the same family name as a system font. The exception for created fonts means that an application can supersede the registration of an earlier created font with a new one.

**Returns:**true if the font is successfully registered in this GraphicsEnvironment.**Since:** 1.6

### preferLocaleFonts

public void **preferLocaleFonts**()

Indicates a preference for locale-specific fonts in the mapping of logical fonts to physical fonts. Calling this method indicates that font rendering should primarily use fonts specific to the primary writing system (the one indicated by the default encoding and the initial default locale). For example, if the primary writing system is Japanese, then characters should be rendered using a Japanese font if possible, and other fonts should only be used for characters for which the Japanese font doesn't have glyphs.

The actual change in font rendering behavior resulting from a call to this method is implementation dependent; it may have no effect at all, or the requested behavior may already match the default behavior. The behavior may differ between font rendering in lightweight and peered components. Since calling this method requests a different font, clients should expect different metrics, and may need to recalculate window sizes and layout. Therefore this method should be called before user interface initialisation.

**Since:** 1.5

### preferProportionalFonts

public void **preferProportionalFonts**()

Indicates a preference for proportional over non-proportional (e.g. dual-spaced CJK fonts) fonts in the mapping of logical fonts to physical fonts. If the default mapping contains fonts for which proportional and non-proportional variants exist, then calling this method indicates the mapping should use a proportional variant.

The actual change in font rendering behavior resulting from a call to this method is implementation dependent; it may have no effect at all. The behavior may differ between font rendering in lightweight and peered components. Since calling this method requests a different font, clients should expect different metrics, and may need to recalculate window sizes and layout. Therefore this method should be called before user interface initialisation.

**Since:** 1.5

### getCenterPoint

public [Point](http://docs.google.com/java/awt/Point.html) **getCenterPoint**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Returns the Point where Windows should be centered. It is recommended that centered Windows be checked to ensure they fit within the available display area using getMaximumWindowBounds().

**Returns:**the point where Windows should be centered **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if isHeadless() returns true**Since:** 1.4 **See Also:**[getMaximumWindowBounds()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getMaximumWindowBounds())

### getMaximumWindowBounds

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

Returns the maximum bounds for centered Windows. These bounds account for objects in the native windowing system such as task bars and menu bars. The returned bounds will reside on a single display with one exception: on multi-screen systems where Windows should be centered across all displays, this method returns the bounds of the entire display area.

To get the usable bounds of a single display, use GraphicsConfiguration.getBounds() and Toolkit.getScreenInsets().

**Returns:**the maximum bounds for centered Windows **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if isHeadless() returns true**Since:** 1.4 **See Also:**[getCenterPoint()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getCenterPoint()), [GraphicsConfiguration.getBounds()](http://docs.google.com/java/awt/GraphicsConfiguration.html#getBounds()), [Toolkit.getScreenInsets(java.awt.GraphicsConfiguration)](http://docs.google.com/java/awt/Toolkit.html#getScreenInsets(java.awt.GraphicsConfiguration))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GraphicsEnvironment.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/GraphicsDevice.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/GridBagConstraints.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/GraphicsEnvironment.html)    [**NO FRAMES**](http://docs.google.com/GraphicsEnvironment.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).