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

## **java.awt**

Class GraphicsDevice

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

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

The GraphicsDevice class describes the graphics devices that might be available in a particular graphics environment. These include screen and printer devices. Note that there can be many screens and many printers in an instance of [GraphicsEnvironment](http://docs.google.com/java/awt/GraphicsEnvironment.html). Each graphics device has one or more [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.

In a multi-screen environment, the GraphicsConfiguration objects can be used to render components on multiple screens. The following code sample demonstrates how to create a JFrame object for each GraphicsConfiguration on each screen device in the GraphicsEnvironment:

GraphicsEnvironment ge = GraphicsEnvironment.  
 getLocalGraphicsEnvironment();  
 GraphicsDevice[] gs = ge.getScreenDevices();  
 for (int j = 0; j < gs.length; j++) {   
 GraphicsDevice gd = gs[j];  
 GraphicsConfiguration[] gc =  
 gd.getConfigurations();  
 for (int i=0; i < gc.length; i++) {  
 JFrame f = new  
 JFrame(gs[j].getDefaultConfiguration());  
 Canvas c = new Canvas(gc[i]);   
 Rectangle gcBounds = gc[i].getBounds();  
 int xoffs = gcBounds.x;  
 int yoffs = gcBounds.y;  
 f.getContentPane().add(c);  
 f.setLocation((i\*50)+xoffs, (i\*60)+yoffs);  
 f.show();  
 }  
 }

For more information on full-screen exclusive mode API, see the  [Full-Screen Exclusive Mode API Tutorial](http://java.sun.com/docs/books/tutorial/extra/fullscreen/index.html).

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

| **Field Summary** | |
| --- | --- |
| static int | [**TYPE\_IMAGE\_BUFFER**](http://docs.google.com/java/awt/GraphicsDevice.html#TYPE_IMAGE_BUFFER)            Device is an image buffer. |
| static int | [**TYPE\_PRINTER**](http://docs.google.com/java/awt/GraphicsDevice.html#TYPE_PRINTER)            Device is a printer. |
| static int | [**TYPE\_RASTER\_SCREEN**](http://docs.google.com/java/awt/GraphicsDevice.html#TYPE_RASTER_SCREEN)            Device is a raster screen. |

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

| **Method Summary** | |
| --- | --- |
| int | [**getAvailableAcceleratedMemory**](http://docs.google.com/java/awt/GraphicsDevice.html#getAvailableAcceleratedMemory())()            This method returns the number of bytes available in accelerated memory on this device. |
| [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) | [**getBestConfiguration**](http://docs.google.com/java/awt/GraphicsDevice.html#getBestConfiguration(java.awt.GraphicsConfigTemplate))([GraphicsConfigTemplate](http://docs.google.com/java/awt/GraphicsConfigTemplate.html) gct)            Returns the "best" configuration possible that passes the criteria defined in the [GraphicsConfigTemplate](http://docs.google.com/java/awt/GraphicsConfigTemplate.html). |
| abstract  [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html)[] | [**getConfigurations**](http://docs.google.com/java/awt/GraphicsDevice.html#getConfigurations())()            Returns all of the GraphicsConfiguration objects associated with this GraphicsDevice. |
| abstract  [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) | [**getDefaultConfiguration**](http://docs.google.com/java/awt/GraphicsDevice.html#getDefaultConfiguration())()            Returns the default GraphicsConfiguration associated with this GraphicsDevice. |
| [DisplayMode](http://docs.google.com/java/awt/DisplayMode.html) | [**getDisplayMode**](http://docs.google.com/java/awt/GraphicsDevice.html#getDisplayMode())()            Returns the current display mode of this GraphicsDevice. |
| [DisplayMode](http://docs.google.com/java/awt/DisplayMode.html)[] | [**getDisplayModes**](http://docs.google.com/java/awt/GraphicsDevice.html#getDisplayModes())()            Returns all display modes available for this GraphicsDevice. |
| [Window](http://docs.google.com/java/awt/Window.html) | [**getFullScreenWindow**](http://docs.google.com/java/awt/GraphicsDevice.html#getFullScreenWindow())()            Returns the Window object representing the full-screen window if the device is in full-screen mode. |
| abstract  [String](http://docs.google.com/java/lang/String.html) | [**getIDstring**](http://docs.google.com/java/awt/GraphicsDevice.html#getIDstring())()            Returns the identification string associated with this GraphicsDevice. |
| abstract  int | [**getType**](http://docs.google.com/java/awt/GraphicsDevice.html#getType())()            Returns the type of this GraphicsDevice. |
| boolean | [**isDisplayChangeSupported**](http://docs.google.com/java/awt/GraphicsDevice.html#isDisplayChangeSupported())()            Returns true if this GraphicsDevice supports low-level display changes. |
| boolean | [**isFullScreenSupported**](http://docs.google.com/java/awt/GraphicsDevice.html#isFullScreenSupported())()            Returns true if this GraphicsDevice supports full-screen exclusive mode. |
| void | [**setDisplayMode**](http://docs.google.com/java/awt/GraphicsDevice.html#setDisplayMode(java.awt.DisplayMode))([DisplayMode](http://docs.google.com/java/awt/DisplayMode.html) dm)            Sets the display mode of this graphics device. |
| void | [**setFullScreenWindow**](http://docs.google.com/java/awt/GraphicsDevice.html#setFullScreenWindow(java.awt.Window))([Window](http://docs.google.com/java/awt/Window.html) w)            Enter full-screen mode, or return to windowed mode. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

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

### TYPE\_RASTER\_SCREEN

public static final int **TYPE\_RASTER\_SCREEN**

Device is a raster screen.

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

### TYPE\_PRINTER

public static final int **TYPE\_PRINTER**

Device is a printer.

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

### TYPE\_IMAGE\_BUFFER

public static final int **TYPE\_IMAGE\_BUFFER**

Device is an image buffer. This buffer can reside in device or system memory but it is not physically viewable by the user.

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

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

### GraphicsDevice

protected **GraphicsDevice**()

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

**See Also:**[GraphicsEnvironment.getScreenDevices()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getScreenDevices()), [GraphicsEnvironment.getDefaultScreenDevice()](http://docs.google.com/java/awt/GraphicsEnvironment.html#getDefaultScreenDevice()), [GraphicsConfiguration.getDevice()](http://docs.google.com/java/awt/GraphicsConfiguration.html#getDevice())

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

### getType

public abstract int **getType**()

Returns the type of this GraphicsDevice.

**Returns:**the type of this GraphicsDevice, which can either be TYPE\_RASTER\_SCREEN, TYPE\_PRINTER or TYPE\_IMAGE\_BUFFER.**See Also:**[TYPE\_RASTER\_SCREEN](http://docs.google.com/java/awt/GraphicsDevice.html#TYPE_RASTER_SCREEN), [TYPE\_PRINTER](http://docs.google.com/java/awt/GraphicsDevice.html#TYPE_PRINTER), [TYPE\_IMAGE\_BUFFER](http://docs.google.com/java/awt/GraphicsDevice.html#TYPE_IMAGE_BUFFER)

### getIDstring

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

Returns the identification string associated with this GraphicsDevice.

A particular program might use more than one GraphicsDevice in a GraphicsEnvironment. This method returns a String identifying a particular GraphicsDevice in the local GraphicsEnvironment. Although there is no public method to set this String, a programmer can use the String for debugging purposes. Vendors of the JavaTM Runtime Environment can format the return value of the String. To determine how to interpret the value of the String, contact the vendor of your Java Runtime. To find out who the vendor is, from your program, call the [getProperty](http://docs.google.com/java/lang/System.html#getProperty(java.lang.String)) method of the System class with "java.vendor".

**Returns:**a String that is the identification of this GraphicsDevice.

### getConfigurations

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

Returns all of the GraphicsConfiguration objects associated with this GraphicsDevice.

**Returns:**an array of GraphicsConfiguration objects that are associated with this GraphicsDevice.

### getDefaultConfiguration

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

Returns the default GraphicsConfiguration associated with this GraphicsDevice.

**Returns:**the default GraphicsConfiguration of this GraphicsDevice.

### getBestConfiguration

public [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) **getBestConfiguration**([GraphicsConfigTemplate](http://docs.google.com/java/awt/GraphicsConfigTemplate.html) gct)

Returns the "best" configuration possible that passes the criteria defined in the [GraphicsConfigTemplate](http://docs.google.com/java/awt/GraphicsConfigTemplate.html).

**Parameters:**gct - the GraphicsConfigTemplate object used to obtain a valid GraphicsConfiguration **Returns:**a GraphicsConfiguration that passes the criteria defined in the specified GraphicsConfigTemplate.**See Also:**[GraphicsConfigTemplate](http://docs.google.com/java/awt/GraphicsConfigTemplate.html)

### isFullScreenSupported

public boolean **isFullScreenSupported**()

Returns true if this GraphicsDevice supports full-screen exclusive mode. If a SecurityManager is installed, its checkPermission method will be called with AWTPermission("fullScreenExclusive"). isFullScreenSupported returns true only if that permission is granted.

**Returns:**whether full-screen exclusive mode is available for this graphics device**Since:** 1.4 **See Also:**[AWTPermission](http://docs.google.com/java/awt/AWTPermission.html)

### setFullScreenWindow

public void **setFullScreenWindow**([Window](http://docs.google.com/java/awt/Window.html) w)

Enter full-screen mode, or return to windowed mode. The entered full-screen mode may be either exclusive or simulated. Exclusive mode is only available if isFullScreenSupported returns true.

Exclusive mode implies:

* Windows cannot overlap the full-screen window. All other application windows will always appear beneath the full-screen window in the Z-order.
* There can be only one full-screen window on a device at any time, so calling this method while there is an existing full-screen Window will cause the existing full-screen window to return to windowed mode.
* Input method windows are disabled. It is advisable to call Component.enableInputMethods(false) to make a component a non-client of the input method framework.

Simulated full-screen mode resizes the window to the size of the screen and positions it at (0,0).

When entering full-screen mode, if the window to be used as the full-screen window is not visible, this method will make it visible. It will remain visible when returning to windowed mode.

When returning to windowed mode from an exclusive full-screen window, any display changes made by calling setDisplayMode are automatically restored to their original state.

**Parameters:**w - a window to use as the full-screen window; null if returning to windowed mode. Some platforms expect the fullscreen window to be a top-level component (i.e., a Frame); therefore it is preferable to use a Frame here rather than a Window.**Since:** 1.4 **See Also:**[isFullScreenSupported()](http://docs.google.com/java/awt/GraphicsDevice.html#isFullScreenSupported()), [getFullScreenWindow()](http://docs.google.com/java/awt/GraphicsDevice.html#getFullScreenWindow()), [setDisplayMode(java.awt.DisplayMode)](http://docs.google.com/java/awt/GraphicsDevice.html#setDisplayMode(java.awt.DisplayMode)), [Component.enableInputMethods(boolean)](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### getFullScreenWindow

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

Returns the Window object representing the full-screen window if the device is in full-screen mode.

**Returns:**the full-screen window, or null if the device is not in full-screen mode.**Since:** 1.4 **See Also:**[setFullScreenWindow(Window)](http://docs.google.com/java/awt/GraphicsDevice.html#setFullScreenWindow(java.awt.Window))

### isDisplayChangeSupported

public boolean **isDisplayChangeSupported**()

Returns true if this GraphicsDevice supports low-level display changes. On some platforms low-level display changes may only be allowed in full-screen exclusive mode (i.e., if [isFullScreenSupported()](http://docs.google.com/java/awt/GraphicsDevice.html#isFullScreenSupported()) returns true and the application has already entered full-screen mode using [setFullScreenWindow(java.awt.Window)](http://docs.google.com/java/awt/GraphicsDevice.html#setFullScreenWindow(java.awt.Window))).

**Returns:**whether low-level display changes are supported for this graphics device.**Since:** 1.4 **See Also:**[isFullScreenSupported()](http://docs.google.com/java/awt/GraphicsDevice.html#isFullScreenSupported()), [setDisplayMode(java.awt.DisplayMode)](http://docs.google.com/java/awt/GraphicsDevice.html#setDisplayMode(java.awt.DisplayMode)), [setFullScreenWindow(java.awt.Window)](http://docs.google.com/java/awt/GraphicsDevice.html#setFullScreenWindow(java.awt.Window))

### setDisplayMode

public void **setDisplayMode**([DisplayMode](http://docs.google.com/java/awt/DisplayMode.html) dm)

Sets the display mode of this graphics device. This is only allowed if [isDisplayChangeSupported()](http://docs.google.com/java/awt/GraphicsDevice.html#isDisplayChangeSupported()) returns true and may require first entering full-screen exclusive mode using [setFullScreenWindow(java.awt.Window)](http://docs.google.com/java/awt/GraphicsDevice.html#setFullScreenWindow(java.awt.Window)) providing that full-screen exclusive mode is supported (i.e., [isFullScreenSupported()](http://docs.google.com/java/awt/GraphicsDevice.html#isFullScreenSupported()) returns true).

The display mode must be one of the display modes returned by [getDisplayModes()](http://docs.google.com/java/awt/GraphicsDevice.html#getDisplayModes()), with one exception: passing a display mode with [DisplayMode.REFRESH\_RATE\_UNKNOWN](http://docs.google.com/java/awt/DisplayMode.html#REFRESH_RATE_UNKNOWN) refresh rate will result in selecting a display mode from the list of available display modes with matching width, height and bit depth. However, passing a display mode with [DisplayMode.BIT\_DEPTH\_MULTI](http://docs.google.com/java/awt/DisplayMode.html#BIT_DEPTH_MULTI) for bit depth is only allowed if such mode exists in the list returned by [getDisplayModes()](http://docs.google.com/java/awt/GraphicsDevice.html#getDisplayModes()).

Example code:

Frame frame;  
 DisplayMode newDisplayMode;  
 GraphicsDevice gd;  
 // create a Frame, select desired DisplayMode from the list of modes  
 // returned by gd.getDisplayModes() ...  
  
 if (gd.isFullScreenSupported()) {  
 gd.setFullScreenWindow(frame);  
 } else {  
 // proceed in non-full-screen mode  
 frame.setSize(...);  
 frame.setLocation(...);  
 frame.setVisible(true);  
 }  
  
 if (gd.isDisplayChangeSupported()) {  
 gd.setDisplayMode(newDisplayMode);  
 }

**Parameters:**dm - The new display mode of this graphics device. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the DisplayMode supplied is null, or is not available in the array returned by getDisplayModes [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if isDisplayChangeSupported returns false**Since:** 1.4 **See Also:**[getDisplayMode()](http://docs.google.com/java/awt/GraphicsDevice.html#getDisplayMode()), [getDisplayModes()](http://docs.google.com/java/awt/GraphicsDevice.html#getDisplayModes()), [isDisplayChangeSupported()](http://docs.google.com/java/awt/GraphicsDevice.html#isDisplayChangeSupported())

### getDisplayMode

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

Returns the current display mode of this GraphicsDevice. The returned display mode is allowed to have a refresh rate [DisplayMode.REFRESH\_RATE\_UNKNOWN](http://docs.google.com/java/awt/DisplayMode.html#REFRESH_RATE_UNKNOWN) if it is indeterminate. Likewise, the returned display mode is allowed to have a bit depth [DisplayMode.BIT\_DEPTH\_MULTI](http://docs.google.com/java/awt/DisplayMode.html#BIT_DEPTH_MULTI) if it is indeterminate or if multiple bit depths are supported.

**Returns:**the current display mode of this graphics device.**Since:** 1.4 **See Also:**[setDisplayMode(DisplayMode)](http://docs.google.com/java/awt/GraphicsDevice.html#setDisplayMode(java.awt.DisplayMode))

### getDisplayModes

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

Returns all display modes available for this GraphicsDevice. The returned display modes are allowed to have a refresh rate [DisplayMode.REFRESH\_RATE\_UNKNOWN](http://docs.google.com/java/awt/DisplayMode.html#REFRESH_RATE_UNKNOWN) if it is indeterminate. Likewise, the returned display modes are allowed to have a bit depth [DisplayMode.BIT\_DEPTH\_MULTI](http://docs.google.com/java/awt/DisplayMode.html#BIT_DEPTH_MULTI) if it is indeterminate or if multiple bit depths are supported.

**Returns:**all of the display modes available for this graphics device.**Since:** 1.4

### getAvailableAcceleratedMemory

public int **getAvailableAcceleratedMemory**()

This method returns the number of bytes available in accelerated memory on this device. Some images are created or cached in accelerated memory on a first-come, first-served basis. On some operating systems, this memory is a finite resource. Calling this method and scheduling the creation and flushing of images carefully may enable applications to make the most efficient use of that finite resource.

Note that the number returned is a snapshot of how much memory is available; some images may still have problems being allocated into that memory. For example, depending on operating system, driver, memory configuration, and thread situations, the full extent of the size reported may not be available for a given image. There are further inquiry methods on the [ImageCapabilities](http://docs.google.com/java/awt/ImageCapabilities.html) object associated with a VolatileImage that can be used to determine whether a particular VolatileImage has been created in accelerated memory.

**Returns:**number of bytes available in accelerated memory. A negative return value indicates that the amount of accelerated memory on this GraphicsDevice is indeterminate.**Since:** 1.4 **See Also:**[Image.flush()](http://docs.google.com/java/awt/Image.html#flush()), [ImageCapabilities.isAccelerated()](http://docs.google.com/java/awt/ImageCapabilities.html#isAccelerated())

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

[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).