| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DisplayMode.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/Dimension.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Event.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/DisplayMode.html)    [**NO FRAMES**](http://docs.google.com/DisplayMode.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **java.awt**

Class DisplayMode

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

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

The DisplayMode class encapsulates the bit depth, height, width, and refresh rate of a GraphicsDevice. The ability to change graphics device's display mode is platform- and configuration-dependent and may not always be available (see [GraphicsDevice.isDisplayChangeSupported()](http://docs.google.com/java/awt/GraphicsDevice.html#isDisplayChangeSupported())).

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).

**Since:** 1.4 **See Also:**[GraphicsDevice](http://docs.google.com/java/awt/GraphicsDevice.html), [GraphicsDevice.isDisplayChangeSupported()](http://docs.google.com/java/awt/GraphicsDevice.html#isDisplayChangeSupported()), [GraphicsDevice.getDisplayModes()](http://docs.google.com/java/awt/GraphicsDevice.html#getDisplayModes()), [GraphicsDevice.setDisplayMode(java.awt.DisplayMode)](http://docs.google.com/java/awt/GraphicsDevice.html#setDisplayMode(java.awt.DisplayMode))

| **Field Summary** | |
| --- | --- |
| static int | [**BIT\_DEPTH\_MULTI**](http://docs.google.com/java/awt/DisplayMode.html#BIT_DEPTH_MULTI)            Value of the bit depth if multiple bit depths are supported in this display mode. |
| static int | [**REFRESH\_RATE\_UNKNOWN**](http://docs.google.com/java/awt/DisplayMode.html#REFRESH_RATE_UNKNOWN)            Value of the refresh rate if not known. |

| **Constructor Summary** | |
| --- | --- |
| [**DisplayMode**](http://docs.google.com/java/awt/DisplayMode.html#DisplayMode(int,%20int,%20int,%20int))(int width, int height, int bitDepth, int refreshRate)            Create a new display mode object with the supplied parameters. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/awt/DisplayMode.html#equals(java.awt.DisplayMode))([DisplayMode](http://docs.google.com/java/awt/DisplayMode.html) dm)            Returns whether the two display modes are equal. |
| boolean | [**equals**](http://docs.google.com/java/awt/DisplayMode.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) dm)            Indicates whether some other object is "equal to" this one. |
| int | [**getBitDepth**](http://docs.google.com/java/awt/DisplayMode.html#getBitDepth())()            Returns the bit depth of the display, in bits per pixel. |
| int | [**getHeight**](http://docs.google.com/java/awt/DisplayMode.html#getHeight())()            Returns the height of the display, in pixels. |
| int | [**getRefreshRate**](http://docs.google.com/java/awt/DisplayMode.html#getRefreshRate())()            Returns the refresh rate of the display, in hertz. |
| int | [**getWidth**](http://docs.google.com/java/awt/DisplayMode.html#getWidth())()            Returns the width of the display, in pixels. |
| int | [**hashCode**](http://docs.google.com/java/awt/DisplayMode.html#hashCode())()            Returns a hash code value for the object. |

| **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()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [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** |
| --- |

### BIT\_DEPTH\_MULTI

public static final int **BIT\_DEPTH\_MULTI**

Value of the bit depth if multiple bit depths are supported in this display mode.

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

### REFRESH\_RATE\_UNKNOWN

public static final int **REFRESH\_RATE\_UNKNOWN**

Value of the refresh rate if not known.

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

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

### DisplayMode

public **DisplayMode**(int width,  
 int height,  
 int bitDepth,  
 int refreshRate)

Create a new display mode object with the supplied parameters.

**Parameters:**width - the width of the display, in pixelsheight - the height of the display, in pixelsbitDepth - the bit depth of the display, in bits per pixel. This can be BIT\_DEPTH\_MULTI if multiple bit depths are available.refreshRate - the refresh rate of the display, in hertz. This can be REFRESH\_RATE\_UNKNOWN if the information is not available.**See Also:**[BIT\_DEPTH\_MULTI](http://docs.google.com/java/awt/DisplayMode.html#BIT_DEPTH_MULTI), [REFRESH\_RATE\_UNKNOWN](http://docs.google.com/java/awt/DisplayMode.html#REFRESH_RATE_UNKNOWN)

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

### getHeight

public int **getHeight**()

Returns the height of the display, in pixels.

**Returns:**the height of the display, in pixels

### getWidth

public int **getWidth**()

Returns the width of the display, in pixels.

**Returns:**the width of the display, in pixels

### getBitDepth

public int **getBitDepth**()

Returns the bit depth of the display, in bits per pixel. This may be BIT\_DEPTH\_MULTI if multiple bit depths are supported in this display mode.

**Returns:**the bit depth of the display, in bits per pixel.**See Also:**[BIT\_DEPTH\_MULTI](http://docs.google.com/java/awt/DisplayMode.html#BIT_DEPTH_MULTI)

### getRefreshRate

public int **getRefreshRate**()

Returns the refresh rate of the display, in hertz. This may be REFRESH\_RATE\_UNKNOWN if the information is not available.

**Returns:**the refresh rate of the display, in hertz.**See Also:**[REFRESH\_RATE\_UNKNOWN](http://docs.google.com/java/awt/DisplayMode.html#REFRESH_RATE_UNKNOWN)

### equals

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

Returns whether the two display modes are equal.

**Returns:**whether the two display modes are equal

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) dm)

Indicates whether some other object is "equal to" this one.

The equals method implements an equivalence relation on non-null object references:

* It is *reflexive*: for any non-null reference value x, x.equals(x) should return true.
* It is *symmetric*: for any non-null reference values x and y, x.equals(y) should return true if and only if y.equals(x) returns true.
* It is *transitive*: for any non-null reference values x, y, and z, if x.equals(y) returns true and y.equals(z) returns true, then x.equals(z) should return true.
* It is *consistent*: for any non-null reference values x and y, multiple invocations of x.equals(y) consistently return true or consistently return false, provided no information used in equals comparisons on the objects is modified.
* For any non-null reference value x, x.equals(null) should return false.

The equals method for class Object implements the most discriminating possible equivalence relation on objects; that is, for any non-null reference values x and y, this method returns true if and only if x and y refer to the same object (x == y has the value true).

Note that it is generally necessary to override the hashCode method whenever this method is overridden, so as to maintain the general contract for the hashCode method, which states that equal objects must have equal hash codes.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**dm - the reference object with which to compare. **Returns:**true if this object is the same as the obj argument; false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns a hash code value for the object. This method is supported for the benefit of hashtables such as those provided by java.util.Hashtable.

The general contract of hashCode is:

* Whenever it is invoked on the same object more than once during an execution of a Java application, the hashCode method must consistently return the same integer, provided no information used in equals comparisons on the object is modified. This integer need not remain consistent from one execution of an application to another execution of the same application.
* If two objects are equal according to the equals(Object) method, then calling the hashCode method on each of the two objects must produce the same integer result.
* It is *not* required that if two objects are unequal according to the [Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) method, then calling the hashCode method on each of the two objects must produce distinct integer results. However, the programmer should be aware that producing distinct integer results for unequal objects may improve the performance of hashtables.

As much as is reasonably practical, the hashCode method defined by class Object does return distinct integers for distinct objects. (This is typically implemented by converting the internal address of the object into an integer, but this implementation technique is not required by the JavaTM programming language.)

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a hash code value for this object.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DisplayMode.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/Dimension.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Event.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/DisplayMode.html)    [**NO FRAMES**](http://docs.google.com/DisplayMode.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).