| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/VolatileImage.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/image/TileObserver.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/WritableRaster.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/VolatileImage.html)    [**NO FRAMES**](http://docs.google.com/VolatileImage.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#1ksv4uv) | [METHOD](#2jxsxqh) |

## **java.awt.image**

Class VolatileImage

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

**All Implemented Interfaces:** [Transparency](http://docs.google.com/java/awt/Transparency.html)

public abstract class **VolatileImage**extends [Image](http://docs.google.com/java/awt/Image.html)implements [Transparency](http://docs.google.com/java/awt/Transparency.html)

VolatileImage is an image which can lose its contents at any time due to circumstances beyond the control of the application (e.g., situations caused by the operating system or by other applications). Because of the potential for hardware acceleration, a VolatileImage object can have significant performance benefits on some platforms.

The drawing surface of an image (the memory where the image contents actually reside) can be lost or invalidated, causing the contents of that memory to go away. The drawing surface thus needs to be restored or recreated and the contents of that surface need to be re-rendered. VolatileImage provides an interface for allowing the user to detect these problems and fix them when they occur.

When a VolatileImage object is created, limited system resources such as video memory (VRAM) may be allocated in order to support the image. When a VolatileImage object is no longer used, it may be garbage-collected and those system resources will be returned, but this process does not happen at guaranteed times. Applications that create many VolatileImage objects (for example, a resizing window may force recreation of its back buffer as the size changes) may run out of optimal system resources for new VolatileImage objects simply because the old objects have not yet been removed from the system. (New VolatileImage objects may still be created, but they may not perform as well as those created in accelerated memory). The flush method may be called at any time to proactively release the resources used by a VolatileImage so that it does not prevent subsequent VolatileImage objects from being accelerated. In this way, applications can have more control over the state of the resources taken up by obsolete VolatileImage objects.

This image should not be subclassed directly but should be created by using the [Component.createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)) or [GraphicsConfiguration.createCompatibleVolatileImage(int, int)](http://docs.google.com/java/awt/GraphicsConfiguration.html#createCompatibleVolatileImage(int,%20int)) methods.

An example of using a VolatileImage object follows:

// image creation  
 VolatileImage vImg = createVolatileImage(w, h);  
  
   
 // rendering to the image  
 void renderOffscreen() {  
 do {  
 if (vImg.validate(getGraphicsConfiguration()) ==  
 VolatileImage.IMAGE\_INCOMPATIBLE)  
 {  
 // old vImg doesn't work with new GraphicsConfig; re-create it  
 vImg = createVolatileImage(w, h);  
 }  
 Graphics2D g = vImg.createGraphics();  
 //  
 // miscellaneous rendering commands...  
 //  
 g.dispose();  
 } while (vImg.contentsLost());  
 }  
  
  
 // copying from the image (here, gScreen is the Graphics  
 // object for the onscreen window)  
 do {  
 int returnCode = vImg.validate(getGraphicsConfiguration());  
 if (returnCode == VolatileImage.IMAGE\_RESTORED) {  
 // Contents need to be restored  
 renderOffscreen(); // restore contents  
 } else if (returnCode == VolatileImage.IMAGE\_INCOMPATIBLE) {  
 // old vImg doesn't work with new GraphicsConfig; re-create it  
 vImg = createVolatileImage(w, h);  
 renderOffscreen();  
 }  
 gScreen.drawImage(vImg, 0, 0, this);  
 } while (vImg.contentsLost());

Note that this class subclasses from the [Image](http://docs.google.com/java/awt/Image.html) class, which includes methods that take an [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) parameter for asynchronous notifications as information is received from a potential [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html). Since this VolatileImage is not loaded from an asynchronous source, the various methods that take an ImageObserver parameter will behave as if the data has already been obtained from the ImageProducer. Specifically, this means that the return values from such methods will never indicate that the information is not yet available and the ImageObserver used in such methods will never need to be recorded for an asynchronous callback notification.

**Since:** 1.4

| **Field Summary** | |
| --- | --- |
| static int | [**IMAGE\_INCOMPATIBLE**](http://docs.google.com/java/awt/image/VolatileImage.html#IMAGE_INCOMPATIBLE)            Validated image is incompatible with supplied GraphicsConfiguration object and should be re-created as appropriate. |
| static int | [**IMAGE\_OK**](http://docs.google.com/java/awt/image/VolatileImage.html#IMAGE_OK)            Validated image is ready to use as-is. |
| static int | [**IMAGE\_RESTORED**](http://docs.google.com/java/awt/image/VolatileImage.html#IMAGE_RESTORED)            Validated image has been restored and is now ready to use. |
| protected  int | [**transparency**](http://docs.google.com/java/awt/image/VolatileImage.html#transparency)            The transparency value with which this image was created. |

| **Fields inherited from class java.awt.**[**Image**](http://docs.google.com/java/awt/Image.html) |
| --- |
| [accelerationPriority](http://docs.google.com/java/awt/Image.html#accelerationPriority), [SCALE\_AREA\_AVERAGING](http://docs.google.com/java/awt/Image.html#SCALE_AREA_AVERAGING), [SCALE\_DEFAULT](http://docs.google.com/java/awt/Image.html#SCALE_DEFAULT), [SCALE\_FAST](http://docs.google.com/java/awt/Image.html#SCALE_FAST), [SCALE\_REPLICATE](http://docs.google.com/java/awt/Image.html#SCALE_REPLICATE), [SCALE\_SMOOTH](http://docs.google.com/java/awt/Image.html#SCALE_SMOOTH), [UndefinedProperty](http://docs.google.com/java/awt/Image.html#UndefinedProperty) |

| **Fields inherited from interface java.awt.**[**Transparency**](http://docs.google.com/java/awt/Transparency.html) |
| --- |
| [BITMASK](http://docs.google.com/java/awt/Transparency.html#BITMASK), [OPAQUE](http://docs.google.com/java/awt/Transparency.html#OPAQUE), [TRANSLUCENT](http://docs.google.com/java/awt/Transparency.html#TRANSLUCENT) |

| **Constructor Summary** | |
| --- | --- |
| [**VolatileImage**](http://docs.google.com/java/awt/image/VolatileImage.html#VolatileImage())() |

| **Method Summary** | |
| --- | --- |
| abstract  boolean | [**contentsLost**](http://docs.google.com/java/awt/image/VolatileImage.html#contentsLost())()            Returns true if rendering data was lost since last validate call. |
| abstract  [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) | [**createGraphics**](http://docs.google.com/java/awt/image/VolatileImage.html#createGraphics())()            Creates a Graphics2D, which can be used to draw into this VolatileImage. |
| abstract  [ImageCapabilities](http://docs.google.com/java/awt/ImageCapabilities.html) | [**getCapabilities**](http://docs.google.com/java/awt/image/VolatileImage.html#getCapabilities())()            Returns an ImageCapabilities object which can be inquired as to the specific capabilities of this VolatileImage. |
| [Graphics](http://docs.google.com/java/awt/Graphics.html) | [**getGraphics**](http://docs.google.com/java/awt/image/VolatileImage.html#getGraphics())()            This method returns a [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html), but is here for backwards compatibility. |
| abstract  int | [**getHeight**](http://docs.google.com/java/awt/image/VolatileImage.html#getHeight())()            Returns the height of the VolatileImage. |
| abstract  [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) | [**getSnapshot**](http://docs.google.com/java/awt/image/VolatileImage.html#getSnapshot())()            Returns a static snapshot image of this object. |
| [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) | [**getSource**](http://docs.google.com/java/awt/image/VolatileImage.html#getSource())()            This returns an ImageProducer for this VolatileImage. |
| int | [**getTransparency**](http://docs.google.com/java/awt/image/VolatileImage.html#getTransparency())()            Returns the transparency. |
| abstract  int | [**getWidth**](http://docs.google.com/java/awt/image/VolatileImage.html#getWidth())()            Returns the width of the VolatileImage. |
| abstract  int | [**validate**](http://docs.google.com/java/awt/image/VolatileImage.html#validate(java.awt.GraphicsConfiguration))([GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)            Attempts to restore the drawing surface of the image if the surface had been lost since the last validate call. |

| **Methods inherited from class java.awt.**[**Image**](http://docs.google.com/java/awt/Image.html) |
| --- |
| [flush](http://docs.google.com/java/awt/Image.html#flush()), [getAccelerationPriority](http://docs.google.com/java/awt/Image.html#getAccelerationPriority()), [getCapabilities](http://docs.google.com/java/awt/Image.html#getCapabilities(java.awt.GraphicsConfiguration)), [getHeight](http://docs.google.com/java/awt/Image.html#getHeight(java.awt.image.ImageObserver)), [getProperty](http://docs.google.com/java/awt/Image.html#getProperty(java.lang.String,%20java.awt.image.ImageObserver)), [getScaledInstance](http://docs.google.com/java/awt/Image.html#getScaledInstance(int,%20int,%20int)), [getWidth](http://docs.google.com/java/awt/Image.html#getWidth(java.awt.image.ImageObserver)), [setAccelerationPriority](http://docs.google.com/java/awt/Image.html#setAccelerationPriority(float)) |

| **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** |
| --- |

### IMAGE\_OK

public static final int **IMAGE\_OK**

Validated image is ready to use as-is.

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

### IMAGE\_RESTORED

public static final int **IMAGE\_RESTORED**

Validated image has been restored and is now ready to use. Note that restoration causes contents of the image to be lost.

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

### IMAGE\_INCOMPATIBLE

public static final int **IMAGE\_INCOMPATIBLE**

Validated image is incompatible with supplied GraphicsConfiguration object and should be re-created as appropriate. Usage of the image as-is after receiving this return code from validate is undefined.

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

### transparency

protected int **transparency**

The transparency value with which this image was created.

**Since:** 1.5 **See Also:**[GraphicsConfiguration.createCompatibleVolatileImage(int, int,int)](http://docs.google.com/java/awt/GraphicsConfiguration.html#createCompatibleVolatileImage(int,%20int,%20int)), [GraphicsConfiguration.createCompatibleVolatileImage(int, int,ImageCapabilities,int)](http://docs.google.com/java/awt/GraphicsConfiguration.html#createCompatibleVolatileImage(int,%20int,%20java.awt.ImageCapabilities,%20int)), [Transparency](http://docs.google.com/java/awt/Transparency.html)

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

### VolatileImage

public **VolatileImage**()

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

### getSnapshot

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

Returns a static snapshot image of this object. The BufferedImage returned is only current with the VolatileImage at the time of the request and will not be updated with any future changes to the VolatileImage.

**Returns:**a [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) representation of this VolatileImage**See Also:**[BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html)

### getWidth

public abstract int **getWidth**()

Returns the width of the VolatileImage.

**Returns:**the width of this VolatileImage.

### getHeight

public abstract int **getHeight**()

Returns the height of the VolatileImage.

**Returns:**the height of this VolatileImage.

### getSource

public [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) **getSource**()

This returns an ImageProducer for this VolatileImage. Note that the VolatileImage object is optimized for rendering operations and blitting to the screen or other VolatileImage objects, as opposed to reading back the pixels of the image. Therefore, operations such as getSource may not perform as fast as operations that do not rely on reading the pixels. Note also that the pixel values read from the image are current with those in the image only at the time that they are retrieved. This method takes a snapshot of the image at the time the request is made and the ImageProducer object returned works with that static snapshot image, not the original VolatileImage. Calling getSource() is equivalent to calling getSnapshot().getSource().

**Specified by:**[getSource](http://docs.google.com/java/awt/Image.html#getSource()) in class [Image](http://docs.google.com/java/awt/Image.html) **Returns:**an [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) that can be used to produce the pixels for a BufferedImage representation of this Image.**See Also:**[ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html), [getSnapshot()](http://docs.google.com/java/awt/image/VolatileImage.html#getSnapshot())

### getGraphics

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

This method returns a [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html), but is here for backwards compatibility. [createGraphics](http://docs.google.com/java/awt/image/VolatileImage.html#createGraphics()) is more convenient, since it is declared to return a Graphics2D.

**Specified by:**[getGraphics](http://docs.google.com/java/awt/Image.html#getGraphics()) in class [Image](http://docs.google.com/java/awt/Image.html) **Returns:**a Graphics2D, which can be used to draw into this image.**See Also:**[Graphics](http://docs.google.com/java/awt/Graphics.html), [Component.createImage(int, int)](http://docs.google.com/java/awt/Component.html#createImage(int,%20int))

### createGraphics

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

Creates a Graphics2D, which can be used to draw into this VolatileImage.

**Returns:**a Graphics2D, used for drawing into this image.

### validate

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

Attempts to restore the drawing surface of the image if the surface had been lost since the last validate call. Also validates this image against the given GraphicsConfiguration parameter to see whether operations from this image to the GraphicsConfiguration are compatible. An example of an incompatible combination might be a situation where a VolatileImage object was created on one graphics device and then was used to render to a different graphics device. Since VolatileImage objects tend to be very device-specific, this operation might not work as intended, so the return code from this validate call would note that incompatibility. A null or incorrect value for gc may cause incorrect values to be returned from validate and may cause later problems with rendering.

**Parameters:**gc - a GraphicsConfiguration object for this image to be validated against. A null gc implies that the validate method should skip the compatibility test. **Returns:**IMAGE\_OK if the image did not need validation

IMAGE\_RESTORED if the image needed restoration. Restoration implies that the contents of the image may have been affected and the image may need to be re-rendered.

IMAGE\_INCOMPATIBLE if the image is incompatible with the GraphicsConfiguration object passed into the validate method. Incompatibility implies that the image may need to be recreated with a new Component or GraphicsConfiguration in order to get an image that can be used successfully with this GraphicsConfiguration. An incompatible image is not checked for whether restoration was necessary, so the state of the image is unchanged after a return value of IMAGE\_INCOMPATIBLE and this return value implies nothing about whether the image needs to be restored.**See Also:**[GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html), [Component](http://docs.google.com/java/awt/Component.html), [IMAGE\_OK](http://docs.google.com/java/awt/image/VolatileImage.html#IMAGE_OK), [IMAGE\_RESTORED](http://docs.google.com/java/awt/image/VolatileImage.html#IMAGE_RESTORED), [IMAGE\_INCOMPATIBLE](http://docs.google.com/java/awt/image/VolatileImage.html#IMAGE_INCOMPATIBLE)

### contentsLost

public abstract boolean **contentsLost**()

Returns true if rendering data was lost since last validate call. This method should be called by the application at the end of any series of rendering operations to or from the image to see whether the image needs to be validated and the rendering redone.

**Returns:**true if the drawing surface needs to be restored; false otherwise.

### getCapabilities

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

Returns an ImageCapabilities object which can be inquired as to the specific capabilities of this VolatileImage. This would allow programmers to find out more runtime information on the specific VolatileImage object that they have created. For example, the user might create a VolatileImage but the system may have no video memory left for creating an image of that size, so although the object is a VolatileImage, it is not as accelerated as other VolatileImage objects on this platform might be. The user might want that information to find other solutions to their problem.

**Returns:**an ImageCapabilities object that contains the capabilities of this VolatileImage.**Since:** 1.4

### getTransparency

public int **getTransparency**()

Returns the transparency. Returns either OPAQUE, BITMASK, or TRANSLUCENT.

**Specified by:**[getTransparency](http://docs.google.com/java/awt/Transparency.html#getTransparency()) in interface [Transparency](http://docs.google.com/java/awt/Transparency.html) **Returns:**the transparency of this VolatileImage.**Since:** 1.5 **See Also:**[Transparency.OPAQUE](http://docs.google.com/java/awt/Transparency.html#OPAQUE), [Transparency.BITMASK](http://docs.google.com/java/awt/Transparency.html#BITMASK), [Transparency.TRANSLUCENT](http://docs.google.com/java/awt/Transparency.html#TRANSLUCENT)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/VolatileImage.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/image/TileObserver.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/WritableRaster.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/VolatileImage.html)    [**NO FRAMES**](http://docs.google.com/VolatileImage.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#1ksv4uv) | [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).