| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Image.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/IllegalComponentStateException.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/ImageCapabilities.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Image.html)    [**NO FRAMES**](http://docs.google.com/Image.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#1ksv4uv) | [METHOD](#2jxsxqh) |

## **java.awt**

Class Image

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

**Direct Known Subclasses:** [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html), [VolatileImage](http://docs.google.com/java/awt/image/VolatileImage.html)

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

The abstract class Image is the superclass of all classes that represent graphical images. The image must be obtained in a platform-specific manner.

**Since:** JDK1.0

| **Field Summary** | |
| --- | --- |
| protected  float | [**accelerationPriority**](http://docs.google.com/java/awt/Image.html#accelerationPriority)            Priority for accelerating this image. |
| static int | [**SCALE\_AREA\_AVERAGING**](http://docs.google.com/java/awt/Image.html#SCALE_AREA_AVERAGING)            Use the Area Averaging image scaling algorithm. |
| static int | [**SCALE\_DEFAULT**](http://docs.google.com/java/awt/Image.html#SCALE_DEFAULT)            Use the default image-scaling algorithm. |
| static int | [**SCALE\_FAST**](http://docs.google.com/java/awt/Image.html#SCALE_FAST)            Choose an image-scaling algorithm that gives higher priority to scaling speed than smoothness of the scaled image. |
| static int | [**SCALE\_REPLICATE**](http://docs.google.com/java/awt/Image.html#SCALE_REPLICATE)            Use the image scaling algorithm embodied in the ReplicateScaleFilter class. |
| static int | [**SCALE\_SMOOTH**](http://docs.google.com/java/awt/Image.html#SCALE_SMOOTH)            Choose an image-scaling algorithm that gives higher priority to image smoothness than scaling speed. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**UndefinedProperty**](http://docs.google.com/java/awt/Image.html#UndefinedProperty)            The UndefinedProperty object should be returned whenever a property which was not defined for a particular image is fetched. |

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

| **Method Summary** | |
| --- | --- |
| void | [**flush**](http://docs.google.com/java/awt/Image.html#flush())()            Flushes all reconstructable resources being used by this Image object. |
| float | [**getAccelerationPriority**](http://docs.google.com/java/awt/Image.html#getAccelerationPriority())()            Returns the current value of the acceleration priority hint. |
| [ImageCapabilities](http://docs.google.com/java/awt/ImageCapabilities.html) | [**getCapabilities**](http://docs.google.com/java/awt/Image.html#getCapabilities(java.awt.GraphicsConfiguration))([GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)            Returns an ImageCapabilities object which can be inquired as to the capabilities of this Image on the specified GraphicsConfiguration. |
| abstract  [Graphics](http://docs.google.com/java/awt/Graphics.html) | [**getGraphics**](http://docs.google.com/java/awt/Image.html#getGraphics())()            Creates a graphics context for drawing to an off-screen image. |
| abstract  int | [**getHeight**](http://docs.google.com/java/awt/Image.html#getHeight(java.awt.image.ImageObserver))([ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Determines the height of the image. |
| abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/java/awt/Image.html#getProperty(java.lang.String,%20java.awt.image.ImageObserver))([String](http://docs.google.com/java/lang/String.html) name, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Gets a property of this image by name. |
| [Image](http://docs.google.com/java/awt/Image.html) | [**getScaledInstance**](http://docs.google.com/java/awt/Image.html#getScaledInstance(int,%20int,%20int))(int width, int height, int hints)            Creates a scaled version of this image. |
| abstract  [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) | [**getSource**](http://docs.google.com/java/awt/Image.html#getSource())()            Gets the object that produces the pixels for the image. |
| abstract  int | [**getWidth**](http://docs.google.com/java/awt/Image.html#getWidth(java.awt.image.ImageObserver))([ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Determines the width of the image. |
| void | [**setAccelerationPriority**](http://docs.google.com/java/awt/Image.html#setAccelerationPriority(float))(float priority)            Sets a hint for this image about how important acceleration is. |

| **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** |
| --- |

### accelerationPriority

protected float **accelerationPriority**

Priority for accelerating this image. Subclasses are free to set different default priorities and applications are free to set the priority for specific images via the setAccelerationPriority(float) method.

**Since:** 1.5

### UndefinedProperty

public static final [Object](http://docs.google.com/java/lang/Object.html) **UndefinedProperty**

The UndefinedProperty object should be returned whenever a property which was not defined for a particular image is fetched.

### SCALE\_DEFAULT

public static final int **SCALE\_DEFAULT**

Use the default image-scaling algorithm.

**Since:** JDK1.1 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Image.SCALE_DEFAULT)

### SCALE\_FAST

public static final int **SCALE\_FAST**

Choose an image-scaling algorithm that gives higher priority to scaling speed than smoothness of the scaled image.

**Since:** JDK1.1 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Image.SCALE_FAST)

### SCALE\_SMOOTH

public static final int **SCALE\_SMOOTH**

Choose an image-scaling algorithm that gives higher priority to image smoothness than scaling speed.

**Since:** JDK1.1 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Image.SCALE_SMOOTH)

### SCALE\_REPLICATE

public static final int **SCALE\_REPLICATE**

Use the image scaling algorithm embodied in the ReplicateScaleFilter class. The Image object is free to substitute a different filter that performs the same algorithm yet integrates more efficiently into the imaging infrastructure supplied by the toolkit.

**Since:** JDK1.1 **See Also:**[ReplicateScaleFilter](http://docs.google.com/java/awt/image/ReplicateScaleFilter.html), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Image.SCALE_REPLICATE)

### SCALE\_AREA\_AVERAGING

public static final int **SCALE\_AREA\_AVERAGING**

Use the Area Averaging image scaling algorithm. The image object is free to substitute a different filter that performs the same algorithm yet integrates more efficiently into the image infrastructure supplied by the toolkit.

**Since:** JDK1.1 **See Also:**[AreaAveragingScaleFilter](http://docs.google.com/java/awt/image/AreaAveragingScaleFilter.html), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Image.SCALE_AREA_AVERAGING)

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

### Image

public **Image**()

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

### getWidth

public abstract int **getWidth**([ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Determines the width of the image. If the width is not yet known, this method returns -1 and the specified ImageObserver object is notified later.

**Parameters:**observer - an object waiting for the image to be loaded. **Returns:**the width of this image, or -1 if the width is not yet known.**See Also:**[getHeight(java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Image.html#getHeight(java.awt.image.ImageObserver)), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html)

### getHeight

public abstract int **getHeight**([ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Determines the height of the image. If the height is not yet known, this method returns -1 and the specified ImageObserver object is notified later.

**Parameters:**observer - an object waiting for the image to be loaded. **Returns:**the height of this image, or -1 if the height is not yet known.**See Also:**[getWidth(java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Image.html#getWidth(java.awt.image.ImageObserver)), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html)

### getSource

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

Gets the object that produces the pixels for the image. This method is called by the image filtering classes and by methods that perform image conversion and scaling.

**Returns:**the image producer that produces the pixels for this image.**See Also:**[ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html)

### getGraphics

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

Creates a graphics context for drawing to an off-screen image. This method can only be called for off-screen images.

**Returns:**a graphics context to draw to the off-screen image. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if called for a non-off-screen 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))

### getProperty

public abstract [Object](http://docs.google.com/java/lang/Object.html) **getProperty**([String](http://docs.google.com/java/lang/String.html) name,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Gets a property of this image by name.

Individual property names are defined by the various image formats. If a property is not defined for a particular image, this method returns the UndefinedProperty object.

If the properties for this image are not yet known, this method returns null, and the ImageObserver object is notified later.

The property name "comment" should be used to store an optional comment which can be presented to the application as a description of the image, its source, or its author.

**Parameters:**name - a property name.observer - an object waiting for this image to be loaded. **Returns:**the value of the named property. **Throws:** NullPointerException - if the property name is null.**See Also:**[ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [UndefinedProperty](http://docs.google.com/java/awt/Image.html#UndefinedProperty)

### getScaledInstance

public [Image](http://docs.google.com/java/awt/Image.html) **getScaledInstance**(int width,  
 int height,  
 int hints)

Creates a scaled version of this image. A new Image object is returned which will render the image at the specified width and height by default. The new Image object may be loaded asynchronously even if the original source image has already been loaded completely.

If either width or height is a negative number then a value is substituted to maintain the aspect ratio of the original image dimensions. If both width and height are negative, then the original image dimensions are used.

**Parameters:**width - the width to which to scale the image.height - the height to which to scale the image.hints - flags to indicate the type of algorithm to use for image resampling. **Returns:**a scaled version of the image. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if width or height is zero.**Since:** JDK1.1 **See Also:**[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\_SMOOTH](http://docs.google.com/java/awt/Image.html#SCALE_SMOOTH), [SCALE\_REPLICATE](http://docs.google.com/java/awt/Image.html#SCALE_REPLICATE), [SCALE\_AREA\_AVERAGING](http://docs.google.com/java/awt/Image.html#SCALE_AREA_AVERAGING)

### flush

public void **flush**()

Flushes all reconstructable resources being used by this Image object. This includes any pixel data that is being cached for rendering to the screen as well as any system resources that are being used to store data or pixels for the image if they can be recreated. The image is reset to a state similar to when it was first created so that if it is again rendered, the image data will have to be recreated or fetched again from its source.

Examples of how this method affects specific types of Image object:

* BufferedImage objects leave the primary Raster which stores their pixels untouched, but flush any information cached about those pixels such as copies uploaded to the display hardware for accelerated blits.
* Image objects created by the Component methods which take a width and height leave their primary buffer of pixels untouched, but have all cached information released much like is done for BufferedImage objects.
* VolatileImage objects release all of their pixel resources including their primary copy which is typically stored on the display hardware where resources are scarce. These objects can later be restored using their [validate](http://docs.google.com/java/awt/image/VolatileImage.html#validate(java.awt.GraphicsConfiguration)) method.
* Image objects created by the Toolkit and Component classes which are loaded from files, URLs or produced by an [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) are unloaded and all local resources are released. These objects can later be reloaded from their original source as needed when they are rendered, just as when they were first created.

### getCapabilities

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

Returns an ImageCapabilities object which can be inquired as to the capabilities of this Image on the specified GraphicsConfiguration. This allows programmers to find out more runtime information on the specific Image object that they have created. For example, the user might create a BufferedImage but the system may have no video memory left for creating an image of that size on the given GraphicsConfiguration, so although the object may be acceleratable in general, it is does not have that capability on this GraphicsConfiguration.

**Parameters:**gc - a GraphicsConfiguration object. A value of null for this parameter will result in getting the image capabilities for the default GraphicsConfiguration. **Returns:**an ImageCapabilities object that contains the capabilities of this Image on the specified GraphicsConfiguration.**Since:** 1.5 **See Also:**[VolatileImage.getCapabilities()](http://docs.google.com/java/awt/image/VolatileImage.html#getCapabilities())

### setAccelerationPriority

public void **setAccelerationPriority**(float priority)

Sets a hint for this image about how important acceleration is. This priority hint is used to compare to the priorities of other Image objects when determining how to use scarce acceleration resources such as video memory. When and if it is possible to accelerate this Image, if there are not enough resources available to provide that acceleration but enough can be freed up by de-accelerating some other image of lower priority, then that other Image may be de-accelerated in deference to this one. Images that have the same priority take up resources on a first-come, first-served basis.

**Parameters:**priority - a value between 0 and 1, inclusive, where higher values indicate more importance for acceleration. A value of 0 means that this Image should never be accelerated. Other values are used simply to determine acceleration priority relative to other Images. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if priority is less than zero or greater than 1.**Since:** 1.5

### getAccelerationPriority

public float **getAccelerationPriority**()

Returns the current value of the acceleration priority hint.

**Returns:**value between 0 and 1, inclusive, which represents the current priority value**Since:** 1.5 **See Also:**[setAccelerationPriority](http://docs.google.com/java/awt/Image.html#setAccelerationPriority(float))

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