| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MediaTracker.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/List.AccessibleAWTList.AccessibleAWTListChild.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Menu.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/MediaTracker.html)    [**NO FRAMES**](http://docs.google.com/MediaTracker.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **java.awt**

Class MediaTracker

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

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

public class **MediaTracker**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Serializable](http://docs.google.com/java/io/Serializable.html)

The MediaTracker class is a utility class to track the status of a number of media objects. Media objects could include audio clips as well as images, though currently only images are supported.

To use a media tracker, create an instance of MediaTracker and call its addImage method for each image to be tracked. In addition, each image can be assigned a unique identifier. This identifier controls the priority order in which the images are fetched. It can also be used to identify unique subsets of the images that can be waited on independently. Images with a lower ID are loaded in preference to those with a higher ID number.

Tracking an animated image might not always be useful due to the multi-part nature of animated image loading and painting, but it is supported. MediaTracker treats an animated image as completely loaded when the first frame is completely loaded. At that point, the MediaTracker signals any waiters that the image is completely loaded. If no ImageObservers are observing the image when the first frame has finished loading, the image might flush itself to conserve resources (see [Image.flush()](http://docs.google.com/java/awt/Image.html#flush())).

Here is an example of using MediaTracker:

import java.applet.Applet;  
 import java.awt.Color;  
 import java.awt.Image;  
 import java.awt.Graphics;  
 import java.awt.MediaTracker;  
  
 public class ImageBlaster extends Applet implements Runnable {  
 MediaTracker tracker;  
 Image bg;  
 Image anim[] = new Image[5];  
 int index;  
 Thread animator;  
  
 // Get the images for the background (id == 0)   
 // and the animation frames (id == 1)   
 // and add them to the MediaTracker  
 public void init() {  
 tracker = new MediaTracker(this);  
 bg = getImage(getDocumentBase(),   
 "images/background.gif");  
 tracker.addImage(bg, 0);  
 for (int i = 0; i < 5; i++) {  
 anim[i] = getImage(getDocumentBase(),   
 "images/anim"+i+".gif");  
 tracker.addImage(anim[i], 1);  
 }  
 }  
  
 // Start the animation thread.  
 public void start() {  
 animator = new Thread(this);  
 animator.start();  
 }  
  
 // Stop the animation thread.  
 public void stop() {  
 animator = null;  
 }  
  
 // Run the animation thread.  
 // First wait for the background image to fully load   
 // and paint. Then wait for all of the animation   
 // frames to finish loading. Finally, loop and   
 // increment the animation frame index.  
 public void run() {  
 try {  
 tracker.waitForID(0);  
 tracker.waitForID(1);  
 } catch (InterruptedException e) {  
 return;  
 }  
 Thread me = Thread.currentThread();  
 while (animator == me) {  
 try {  
 Thread.sleep(100);  
 } catch (InterruptedException e) {  
 break;  
 }  
 synchronized (this) {  
 index++;  
 if (index >= anim.length) {  
 index = 0;  
 }  
 }  
 repaint();  
 }  
 }  
  
 // The background image fills the frame so we   
 // don't need to clear the applet on repaints.   
 // Just call the paint method.  
 public void update(Graphics g) {  
 paint(g);  
 }  
  
 // Paint a large red rectangle if there are any errors   
 // loading the images. Otherwise always paint the   
 // background so that it appears incrementally as it   
 // is loading. Finally, only paint the current animation   
 // frame if all of the frames (id == 1) are done loading,  
 // so that we don't get partial animations.  
 public void paint(Graphics g) {  
 if ((tracker.statusAll(false) & MediaTracker.ERRORED) != 0) {  
 g.setColor(Color.red);  
 g.fillRect(0, 0, size().width, size().height);  
 return;  
 }  
 g.drawImage(bg, 0, 0, this);  
 if (tracker.statusID(1, false) == MediaTracker.COMPLETE) {  
 g.drawImage(anim[index], 10, 10, this);  
 }  
 }  
 }

**Since:** JDK1.0 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.awt.MediaTracker)

| **Field Summary** | |
| --- | --- |
| static int | [**ABORTED**](http://docs.google.com/java/awt/MediaTracker.html#ABORTED)            Flag indicating that the downloading of media was aborted. |
| static int | [**COMPLETE**](http://docs.google.com/java/awt/MediaTracker.html#COMPLETE)            Flag indicating that the downloading of media was completed successfully. |
| static int | [**ERRORED**](http://docs.google.com/java/awt/MediaTracker.html#ERRORED)            Flag indicating that the downloading of media encountered an error. |
| static int | [**LOADING**](http://docs.google.com/java/awt/MediaTracker.html#LOADING)            Flag indicating that media is currently being loaded. |

| **Constructor Summary** | |
| --- | --- |
| [**MediaTracker**](http://docs.google.com/java/awt/MediaTracker.html#MediaTracker(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Creates a media tracker to track images for a given component. |

| **Method Summary** | |
| --- | --- |
| void | [**addImage**](http://docs.google.com/java/awt/MediaTracker.html#addImage(java.awt.Image,%20int))([Image](http://docs.google.com/java/awt/Image.html) image, int id)            Adds an image to the list of images being tracked by this media tracker. |
| void | [**addImage**](http://docs.google.com/java/awt/MediaTracker.html#addImage(java.awt.Image,%20int,%20int,%20int))([Image](http://docs.google.com/java/awt/Image.html) image, int id, int w, int h)            Adds a scaled image to the list of images being tracked by this media tracker. |
| boolean | [**checkAll**](http://docs.google.com/java/awt/MediaTracker.html#checkAll())()            Checks to see if all images being tracked by this media tracker have finished loading. |
| boolean | [**checkAll**](http://docs.google.com/java/awt/MediaTracker.html#checkAll(boolean))(boolean load)            Checks to see if all images being tracked by this media tracker have finished loading. |
| boolean | [**checkID**](http://docs.google.com/java/awt/MediaTracker.html#checkID(int))(int id)            Checks to see if all images tracked by this media tracker that are tagged with the specified identifier have finished loading. |
| boolean | [**checkID**](http://docs.google.com/java/awt/MediaTracker.html#checkID(int,%20boolean))(int id, boolean load)            Checks to see if all images tracked by this media tracker that are tagged with the specified identifier have finished loading. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getErrorsAny**](http://docs.google.com/java/awt/MediaTracker.html#getErrorsAny())()            Returns a list of all media that have encountered an error. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getErrorsID**](http://docs.google.com/java/awt/MediaTracker.html#getErrorsID(int))(int id)            Returns a list of media with the specified ID that have encountered an error. |
| boolean | [**isErrorAny**](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny())()            Checks the error status of all of the images. |
| boolean | [**isErrorID**](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))(int id)            Checks the error status of all of the images tracked by this media tracker with the specified identifier. |
| void | [**removeImage**](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image))([Image](http://docs.google.com/java/awt/Image.html) image)            Removes the specified image from this media tracker. |
| void | [**removeImage**](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image,%20int))([Image](http://docs.google.com/java/awt/Image.html) image, int id)            Removes the specified image from the specified tracking ID of this media tracker. |
| void | [**removeImage**](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image,%20int,%20int,%20int))([Image](http://docs.google.com/java/awt/Image.html) image, int id, int width, int height)            Removes the specified image with the specified width, height, and ID from this media tracker. |
| int | [**statusAll**](http://docs.google.com/java/awt/MediaTracker.html#statusAll(boolean))(boolean load)            Calculates and returns the bitwise inclusive **OR** of the status of all media that are tracked by this media tracker. |
| int | [**statusID**](http://docs.google.com/java/awt/MediaTracker.html#statusID(int,%20boolean))(int id, boolean load)            Calculates and returns the bitwise inclusive **OR** of the status of all media with the specified identifier that are tracked by this media tracker. |
| void | [**waitForAll**](http://docs.google.com/java/awt/MediaTracker.html#waitForAll())()            Starts loading all images tracked by this media tracker. |
| boolean | [**waitForAll**](http://docs.google.com/java/awt/MediaTracker.html#waitForAll(long))(long ms)            Starts loading all images tracked by this media tracker. |
| void | [**waitForID**](http://docs.google.com/java/awt/MediaTracker.html#waitForID(int))(int id)            Starts loading all images tracked by this media tracker with the specified identifier. |
| boolean | [**waitForID**](http://docs.google.com/java/awt/MediaTracker.html#waitForID(int,%20long))(int id, long ms)            Starts loading all images tracked by this media tracker with the specified identifier. |

| **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** |
| --- |

### LOADING

public static final int **LOADING**

Flag indicating that media is currently being loaded.

**See Also:**[statusAll(boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusAll(boolean)), [statusID(int, boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusID(int,%20boolean)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.MediaTracker.LOADING)

### ABORTED

public static final int **ABORTED**

Flag indicating that the downloading of media was aborted.

**See Also:**[statusAll(boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusAll(boolean)), [statusID(int, boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusID(int,%20boolean)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.MediaTracker.ABORTED)

### ERRORED

public static final int **ERRORED**

Flag indicating that the downloading of media encountered an error.

**See Also:**[statusAll(boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusAll(boolean)), [statusID(int, boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusID(int,%20boolean)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.MediaTracker.ERRORED)

### COMPLETE

public static final int **COMPLETE**

Flag indicating that the downloading of media was completed successfully.

**See Also:**[statusAll(boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusAll(boolean)), [statusID(int, boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusID(int,%20boolean)), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.MediaTracker.COMPLETE)

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

### MediaTracker

public **MediaTracker**([Component](http://docs.google.com/java/awt/Component.html) comp)

Creates a media tracker to track images for a given component.

**Parameters:**comp - the component on which the images will eventually be drawn

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

### addImage

public void **addImage**([Image](http://docs.google.com/java/awt/Image.html) image,  
 int id)

Adds an image to the list of images being tracked by this media tracker. The image will eventually be rendered at its default (unscaled) size.

**Parameters:**image - the image to be trackedid - an identifier used to track this image

### addImage

public void **addImage**([Image](http://docs.google.com/java/awt/Image.html) image,  
 int id,  
 int w,  
 int h)

Adds a scaled image to the list of images being tracked by this media tracker. The image will eventually be rendered at the indicated width and height.

**Parameters:**image - the image to be trackedid - an identifier that can be used to track this imagew - the width at which the image is renderedh - the height at which the image is rendered

### checkAll

public boolean **checkAll**()

Checks to see if all images being tracked by this media tracker have finished loading.

This method does not start loading the images if they are not already loading.

If there is an error while loading or scaling an image, then that image is considered to have finished loading. Use the isErrorAny or isErrorID methods to check for errors.

**Returns:**true if all images have finished loading, have been aborted, or have encountered an error; false otherwise**See Also:**[checkAll(boolean)](http://docs.google.com/java/awt/MediaTracker.html#checkAll(boolean)), [checkID(int)](http://docs.google.com/java/awt/MediaTracker.html#checkID(int)), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))

### checkAll

public boolean **checkAll**(boolean load)

Checks to see if all images being tracked by this media tracker have finished loading.

If the value of the load flag is true, then this method starts loading any images that are not yet being loaded.

If there is an error while loading or scaling an image, that image is considered to have finished loading. Use the isErrorAny and isErrorID methods to check for errors.

**Parameters:**load - if true, start loading any images that are not yet being loaded **Returns:**true if all images have finished loading, have been aborted, or have encountered an error; false otherwise**See Also:**[checkID(int)](http://docs.google.com/java/awt/MediaTracker.html#checkID(int)), [checkAll()](http://docs.google.com/java/awt/MediaTracker.html#checkAll()), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))

### isErrorAny

public boolean **isErrorAny**()

Checks the error status of all of the images.

**Returns:**true if any of the images tracked by this media tracker had an error during loading; false otherwise**See Also:**[isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int)), [getErrorsAny()](http://docs.google.com/java/awt/MediaTracker.html#getErrorsAny())

### getErrorsAny

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

Returns a list of all media that have encountered an error.

**Returns:**an array of media objects tracked by this media tracker that have encountered an error, or null if there are none with errors**See Also:**[isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [getErrorsID(int)](http://docs.google.com/java/awt/MediaTracker.html#getErrorsID(int))

### waitForAll

public void **waitForAll**()  
 throws [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html)

Starts loading all images tracked by this media tracker. This method waits until all the images being tracked have finished loading.

If there is an error while loading or scaling an image, then that image is considered to have finished loading. Use the isErrorAny or isErrorID methods to check for errors.

**Throws:** [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) - if any thread has interrupted this thread**See Also:**[waitForID(int)](http://docs.google.com/java/awt/MediaTracker.html#waitForID(int)), [waitForAll(long)](http://docs.google.com/java/awt/MediaTracker.html#waitForAll(long)), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))

### waitForAll

public boolean **waitForAll**(long ms)  
 throws [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html)

Starts loading all images tracked by this media tracker. This method waits until all the images being tracked have finished loading, or until the length of time specified in milliseconds by the ms argument has passed.

If there is an error while loading or scaling an image, then that image is considered to have finished loading. Use the isErrorAny or isErrorID methods to check for errors.

**Parameters:**ms - the number of milliseconds to wait for the loading to complete **Returns:**true if all images were successfully loaded; false otherwise **Throws:** [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) - if any thread has interrupted this thread.**See Also:**[waitForID(int)](http://docs.google.com/java/awt/MediaTracker.html#waitForID(int)), [waitForAll(long)](http://docs.google.com/java/awt/MediaTracker.html#waitForAll(long)), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))

### statusAll

public int **statusAll**(boolean load)

Calculates and returns the bitwise inclusive **OR** of the status of all media that are tracked by this media tracker.

Possible flags defined by the MediaTracker class are LOADING, ABORTED, ERRORED, and COMPLETE. An image that hasn't started loading has zero as its status.

If the value of load is true, then this method starts loading any images that are not yet being loaded.

**Parameters:**load - if true, start loading any images that are not yet being loaded **Returns:**the bitwise inclusive **OR** of the status of all of the media being tracked**See Also:**[statusID(int, boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusID(int,%20boolean)), [LOADING](http://docs.google.com/java/awt/MediaTracker.html#LOADING), [ABORTED](http://docs.google.com/java/awt/MediaTracker.html#ABORTED), [ERRORED](http://docs.google.com/java/awt/MediaTracker.html#ERRORED), [COMPLETE](http://docs.google.com/java/awt/MediaTracker.html#COMPLETE)

### checkID

public boolean **checkID**(int id)

Checks to see if all images tracked by this media tracker that are tagged with the specified identifier have finished loading.

This method does not start loading the images if they are not already loading.

If there is an error while loading or scaling an image, then that image is considered to have finished loading. Use the isErrorAny or isErrorID methods to check for errors.

**Parameters:**id - the identifier of the images to check **Returns:**true if all images have finished loading, have been aborted, or have encountered an error; false otherwise**See Also:**[checkID(int, boolean)](http://docs.google.com/java/awt/MediaTracker.html#checkID(int,%20boolean)), [checkAll()](http://docs.google.com/java/awt/MediaTracker.html#checkAll()), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))

### checkID

public boolean **checkID**(int id,  
 boolean load)

Checks to see if all images tracked by this media tracker that are tagged with the specified identifier have finished loading.

If the value of the load flag is true, then this method starts loading any images that are not yet being loaded.

If there is an error while loading or scaling an image, then that image is considered to have finished loading. Use the isErrorAny or isErrorID methods to check for errors.

**Parameters:**id - the identifier of the images to checkload - if true, start loading any images that are not yet being loaded **Returns:**true if all images have finished loading, have been aborted, or have encountered an error; false otherwise**See Also:**[checkID(int, boolean)](http://docs.google.com/java/awt/MediaTracker.html#checkID(int,%20boolean)), [checkAll()](http://docs.google.com/java/awt/MediaTracker.html#checkAll()), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))

### isErrorID

public boolean **isErrorID**(int id)

Checks the error status of all of the images tracked by this media tracker with the specified identifier.

**Parameters:**id - the identifier of the images to check **Returns:**true if any of the images with the specified identifier had an error during loading; false otherwise**See Also:**[isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [getErrorsID(int)](http://docs.google.com/java/awt/MediaTracker.html#getErrorsID(int))

### getErrorsID

public [Object](http://docs.google.com/java/lang/Object.html)[] **getErrorsID**(int id)

Returns a list of media with the specified ID that have encountered an error.

**Parameters:**id - the identifier of the images to check **Returns:**an array of media objects tracked by this media tracker with the specified identifier that have encountered an error, or null if there are none with errors**See Also:**[isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int)), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [getErrorsAny()](http://docs.google.com/java/awt/MediaTracker.html#getErrorsAny())

### waitForID

public void **waitForID**(int id)  
 throws [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html)

Starts loading all images tracked by this media tracker with the specified identifier. This method waits until all the images with the specified identifier have finished loading.

If there is an error while loading or scaling an image, then that image is considered to have finished loading. Use the isErrorAny and isErrorID methods to check for errors.

**Parameters:**id - the identifier of the images to check **Throws:** [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) - if any thread has interrupted this thread.**See Also:**[waitForAll()](http://docs.google.com/java/awt/MediaTracker.html#waitForAll()), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))

### waitForID

public boolean **waitForID**(int id,  
 long ms)  
 throws [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html)

Starts loading all images tracked by this media tracker with the specified identifier. This method waits until all the images with the specified identifier have finished loading, or until the length of time specified in milliseconds by the ms argument has passed.

If there is an error while loading or scaling an image, then that image is considered to have finished loading. Use the statusID, isErrorID, and isErrorAny methods to check for errors.

**Parameters:**id - the identifier of the images to checkms - the length of time, in milliseconds, to wait for the loading to complete **Throws:** [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) - if any thread has interrupted this thread.**See Also:**[waitForAll()](http://docs.google.com/java/awt/MediaTracker.html#waitForAll()), [waitForID(int)](http://docs.google.com/java/awt/MediaTracker.html#waitForID(int)), [statusID(int, boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusID(int,%20boolean)), [isErrorAny()](http://docs.google.com/java/awt/MediaTracker.html#isErrorAny()), [isErrorID(int)](http://docs.google.com/java/awt/MediaTracker.html#isErrorID(int))

### statusID

public int **statusID**(int id,  
 boolean load)

Calculates and returns the bitwise inclusive **OR** of the status of all media with the specified identifier that are tracked by this media tracker.

Possible flags defined by the MediaTracker class are LOADING, ABORTED, ERRORED, and COMPLETE. An image that hasn't started loading has zero as its status.

If the value of load is true, then this method starts loading any images that are not yet being loaded.

**Parameters:**id - the identifier of the images to checkload - if true, start loading any images that are not yet being loaded **Returns:**the bitwise inclusive **OR** of the status of all of the media with the specified identifier that are being tracked**See Also:**[statusAll(boolean)](http://docs.google.com/java/awt/MediaTracker.html#statusAll(boolean)), [LOADING](http://docs.google.com/java/awt/MediaTracker.html#LOADING), [ABORTED](http://docs.google.com/java/awt/MediaTracker.html#ABORTED), [ERRORED](http://docs.google.com/java/awt/MediaTracker.html#ERRORED), [COMPLETE](http://docs.google.com/java/awt/MediaTracker.html#COMPLETE)

### removeImage

public void **removeImage**([Image](http://docs.google.com/java/awt/Image.html) image)

Removes the specified image from this media tracker. All instances of the specified image are removed, regardless of scale or ID.

**Parameters:**image - the image to be removed**Since:** JDK1.1 **See Also:**[removeImage(java.awt.Image, int)](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image,%20int)), [removeImage(java.awt.Image, int, int, int)](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image,%20int,%20int,%20int))

### removeImage

public void **removeImage**([Image](http://docs.google.com/java/awt/Image.html) image,  
 int id)

Removes the specified image from the specified tracking ID of this media tracker. All instances of Image being tracked under the specified ID are removed regardless of scale.

**Parameters:**image - the image to be removedid - the tracking ID frrom which to remove the image**Since:** JDK1.1 **See Also:**[removeImage(java.awt.Image)](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image)), [removeImage(java.awt.Image, int, int, int)](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image,%20int,%20int,%20int))

### removeImage

public void **removeImage**([Image](http://docs.google.com/java/awt/Image.html) image,  
 int id,  
 int width,  
 int height)

Removes the specified image with the specified width, height, and ID from this media tracker. Only the specified instance (with any duplicates) is removed.

**Parameters:**image - the image to be removedid - the tracking ID from which to remove the imagewidth - the width to remove (-1 for unscaled)height - the height to remove (-1 for unscaled)**Since:** JDK1.1 **See Also:**[removeImage(java.awt.Image)](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image)), [removeImage(java.awt.Image, int)](http://docs.google.com/java/awt/MediaTracker.html#removeImage(java.awt.Image,%20int))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MediaTracker.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/List.AccessibleAWTList.AccessibleAWTListChild.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Menu.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/MediaTracker.html)    [**NO FRAMES**](http://docs.google.com/MediaTracker.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

[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).