| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MemoryImageSource.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/LookupTable.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/MultiPixelPackedSampleModel.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/MemoryImageSource.html)    [**NO FRAMES**](http://docs.google.com/MemoryImageSource.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#lnxbz9) |

## **java.awt.image**

Class MemoryImageSource

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

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

public class **MemoryImageSource**extends [Object](http://docs.google.com/java/lang/Object.html)implements [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html)

This class is an implementation of the ImageProducer interface which uses an array to produce pixel values for an Image. Here is an example which calculates a 100x100 image representing a fade from black to blue along the X axis and a fade from black to red along the Y axis:

int w = 100;  
 int h = 100;  
 int pix[] = new int[w \* h];  
 int index = 0;  
 for (int y = 0; y < h; y++) {  
 int red = (y \* 255) / (h - 1);  
 for (int x = 0; x < w; x++) {  
 int blue = (x \* 255) / (w - 1);  
 pix[index++] = (255 << 24) | (red << 16) | blue;  
 }  
 }  
 Image img = createImage(new MemoryImageSource(w, h, pix, 0, w));

The MemoryImageSource is also capable of managing a memory image which varies over time to allow animation or custom rendering. Here is an example showing how to set up the animation source and signal changes in the data (adapted from the MemoryAnimationSourceDemo by Garth Dickie):

int pixels[];  
 MemoryImageSource source;  
  
 public void init() {  
 int width = 50;  
 int height = 50;  
 int size = width \* height;  
 pixels = new int[size];  
  
 int value = getBackground().getRGB();  
 for (int i = 0; i < size; i++) {  
 pixels[i] = value;  
 }  
  
 source = new MemoryImageSource(width, height, pixels, 0, width);  
 source.setAnimated(true);  
 image = createImage(source);  
 }  
  
 public void run() {  
 Thread me = Thread.currentThread( );  
 me.setPriority(Thread.MIN\_PRIORITY);  
  
 while (true) {  
 try {  
 thread.sleep(10);  
 } catch( InterruptedException e ) {  
 return;  
 }  
  
 // Modify the values in the pixels array at (x, y, w, h)  
  
 // Send the new data to the interested ImageConsumers  
 source.newPixels(x, y, w, h);  
 }  
 }

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

| **Constructor Summary** | |
| --- | --- |
| [**MemoryImageSource**](http://docs.google.com/java/awt/image/MemoryImageSource.html#MemoryImageSource(int,%20int,%20java.awt.image.ColorModel,%20byte%5B%5D,%20int,%20int))(int w, int h, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm, byte[] pix, int off, int scan)            Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object. |
| [**MemoryImageSource**](http://docs.google.com/java/awt/image/MemoryImageSource.html#MemoryImageSource(int,%20int,%20java.awt.image.ColorModel,%20byte%5B%5D,%20int,%20int,%20java.util.Hashtable))(int w, int h, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm, byte[] pix, int off, int scan, [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> props)            Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object. |
| [**MemoryImageSource**](http://docs.google.com/java/awt/image/MemoryImageSource.html#MemoryImageSource(int,%20int,%20java.awt.image.ColorModel,%20int%5B%5D,%20int,%20int))(int w, int h, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm, int[] pix, int off, int scan)            Constructs an ImageProducer object which uses an array of integers to produce data for an Image object. |
| [**MemoryImageSource**](http://docs.google.com/java/awt/image/MemoryImageSource.html#MemoryImageSource(int,%20int,%20java.awt.image.ColorModel,%20int%5B%5D,%20int,%20int,%20java.util.Hashtable))(int w, int h, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm, int[] pix, int off, int scan, [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> props)            Constructs an ImageProducer object which uses an array of integers to produce data for an Image object. |
| [**MemoryImageSource**](http://docs.google.com/java/awt/image/MemoryImageSource.html#MemoryImageSource(int,%20int,%20int%5B%5D,%20int,%20int))(int w, int h, int[] pix, int off, int scan)            Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object. |
| [**MemoryImageSource**](http://docs.google.com/java/awt/image/MemoryImageSource.html#MemoryImageSource(int,%20int,%20int%5B%5D,%20int,%20int,%20java.util.Hashtable))(int w, int h, int[] pix, int off, int scan, [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> props)            Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object. |

| **Method Summary** | |
| --- | --- |
| void | [**addConsumer**](http://docs.google.com/java/awt/image/MemoryImageSource.html#addConsumer(java.awt.image.ImageConsumer))([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)            Adds an ImageConsumer to the list of consumers interested in data for this image. |
| boolean | [**isConsumer**](http://docs.google.com/java/awt/image/MemoryImageSource.html#isConsumer(java.awt.image.ImageConsumer))([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)            Determines if an ImageConsumer is on the list of consumers currently interested in data for this image. |
| void | [**newPixels**](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels())()            Sends a whole new buffer of pixels to any ImageConsumers that are currently interested in the data for this image and notify them that an animation frame is complete. |
| void | [**newPixels**](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels(byte%5B%5D,%20java.awt.image.ColorModel,%20int,%20int))(byte[] newpix, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) newmodel, int offset, int scansize)            Changes to a new byte array to hold the pixels for this image. |
| void | [**newPixels**](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels(int%5B%5D,%20java.awt.image.ColorModel,%20int,%20int))(int[] newpix, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) newmodel, int offset, int scansize)            Changes to a new int array to hold the pixels for this image. |
| void | [**newPixels**](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels(int,%20int,%20int,%20int))(int x, int y, int w, int h)            Sends a rectangular region of the buffer of pixels to any ImageConsumers that are currently interested in the data for this image and notify them that an animation frame is complete. |
| void | [**newPixels**](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels(int,%20int,%20int,%20int,%20boolean))(int x, int y, int w, int h, boolean framenotify)            Sends a rectangular region of the buffer of pixels to any ImageConsumers that are currently interested in the data for this image. |
| void | [**removeConsumer**](http://docs.google.com/java/awt/image/MemoryImageSource.html#removeConsumer(java.awt.image.ImageConsumer))([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)            Removes an ImageConsumer from the list of consumers interested in data for this image. |
| void | [**requestTopDownLeftRightResend**](http://docs.google.com/java/awt/image/MemoryImageSource.html#requestTopDownLeftRightResend(java.awt.image.ImageConsumer))([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)            Requests that a given ImageConsumer have the image data delivered one more time in top-down, left-right order. |
| void | [**setAnimated**](http://docs.google.com/java/awt/image/MemoryImageSource.html#setAnimated(boolean))(boolean animated)            Changes this memory image into a multi-frame animation or a single-frame static image depending on the animated parameter. |
| void | [**setFullBufferUpdates**](http://docs.google.com/java/awt/image/MemoryImageSource.html#setFullBufferUpdates(boolean))(boolean fullbuffers)            Specifies whether this animated memory image should always be updated by sending the complete buffer of pixels whenever there is a change. |
| void | [**startProduction**](http://docs.google.com/java/awt/image/MemoryImageSource.html#startProduction(java.awt.image.ImageConsumer))([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)            Adds an ImageConsumer to the list of consumers interested in data for this image and immediately starts delivery of the image data through the ImageConsumer interface. |

| **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)) |

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

### MemoryImageSource

public **MemoryImageSource**(int w,  
 int h,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm,  
 byte[] pix,  
 int off,  
 int scan)

Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object.

**Parameters:**w - the width of the rectangle of pixelsh - the height of the rectangle of pixelscm - the specified ColorModelpix - an array of pixelsoff - the offset into the array of where to store the first pixelscan - the distance from one row of pixels to the next in the array**See Also:**[Component.createImage(java.awt.image.ImageProducer)](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer))

### MemoryImageSource

public **MemoryImageSource**(int w,  
 int h,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm,  
 byte[] pix,  
 int off,  
 int scan,  
 [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> props)

Constructs an ImageProducer object which uses an array of bytes to produce data for an Image object.

**Parameters:**w - the width of the rectangle of pixelsh - the height of the rectangle of pixelscm - the specified ColorModelpix - an array of pixelsoff - the offset into the array of where to store the first pixelscan - the distance from one row of pixels to the next in the arrayprops - a list of properties that the ImageProducer uses to process an image**See Also:**[Component.createImage(java.awt.image.ImageProducer)](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer))

### MemoryImageSource

public **MemoryImageSource**(int w,  
 int h,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm,  
 int[] pix,  
 int off,  
 int scan)

Constructs an ImageProducer object which uses an array of integers to produce data for an Image object.

**Parameters:**w - the width of the rectangle of pixelsh - the height of the rectangle of pixelscm - the specified ColorModelpix - an array of pixelsoff - the offset into the array of where to store the first pixelscan - the distance from one row of pixels to the next in the array**See Also:**[Component.createImage(java.awt.image.ImageProducer)](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer))

### MemoryImageSource

public **MemoryImageSource**(int w,  
 int h,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm,  
 int[] pix,  
 int off,  
 int scan,  
 [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> props)

Constructs an ImageProducer object which uses an array of integers to produce data for an Image object.

**Parameters:**w - the width of the rectangle of pixelsh - the height of the rectangle of pixelscm - the specified ColorModelpix - an array of pixelsoff - the offset into the array of where to store the first pixelscan - the distance from one row of pixels to the next in the arrayprops - a list of properties that the ImageProducer uses to process an image**See Also:**[Component.createImage(java.awt.image.ImageProducer)](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer))

### MemoryImageSource

public **MemoryImageSource**(int w,  
 int h,  
 int[] pix,  
 int off,  
 int scan)

Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

**Parameters:**w - the width of the rectangle of pixelsh - the height of the rectangle of pixelspix - an array of pixelsoff - the offset into the array of where to store the first pixelscan - the distance from one row of pixels to the next in the array**See Also:**[Component.createImage(java.awt.image.ImageProducer)](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault())

### MemoryImageSource

public **MemoryImageSource**(int w,  
 int h,  
 int[] pix,  
 int off,  
 int scan,  
 [Hashtable](http://docs.google.com/java/util/Hashtable.html)<?,?> props)

Constructs an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

**Parameters:**w - the width of the rectangle of pixelsh - the height of the rectangle of pixelspix - an array of pixelsoff - the offset into the array of where to store the first pixelscan - the distance from one row of pixels to the next in the arrayprops - a list of properties that the ImageProducer uses to process an image**See Also:**[Component.createImage(java.awt.image.ImageProducer)](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault())

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

### addConsumer

public void **addConsumer**([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)

Adds an ImageConsumer to the list of consumers interested in data for this image.

**Specified by:**[addConsumer](http://docs.google.com/java/awt/image/ImageProducer.html#addConsumer(java.awt.image.ImageConsumer)) in interface [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) **Parameters:**ic - the specified ImageConsumer **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified ImageConsumer is null**See Also:**[ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html)

### isConsumer

public boolean **isConsumer**([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)

Determines if an ImageConsumer is on the list of consumers currently interested in data for this image.

**Specified by:**[isConsumer](http://docs.google.com/java/awt/image/ImageProducer.html#isConsumer(java.awt.image.ImageConsumer)) in interface [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) **Parameters:**ic - the specified ImageConsumer **Returns:**true if the ImageConsumer is on the list; false otherwise.**See Also:**[ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html)

### removeConsumer

public void **removeConsumer**([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)

Removes an ImageConsumer from the list of consumers interested in data for this image.

**Specified by:**[removeConsumer](http://docs.google.com/java/awt/image/ImageProducer.html#removeConsumer(java.awt.image.ImageConsumer)) in interface [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) **Parameters:**ic - the specified ImageConsumer**See Also:**[ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html)

### startProduction

public void **startProduction**([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)

Adds an ImageConsumer to the list of consumers interested in data for this image and immediately starts delivery of the image data through the ImageConsumer interface.

**Specified by:**[startProduction](http://docs.google.com/java/awt/image/ImageProducer.html#startProduction(java.awt.image.ImageConsumer)) in interface [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) **Parameters:**ic - the specified ImageConsumer image data through the ImageConsumer interface.**See Also:**[ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html)

### requestTopDownLeftRightResend

public void **requestTopDownLeftRightResend**([ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html) ic)

Requests that a given ImageConsumer have the image data delivered one more time in top-down, left-right order.

**Specified by:**[requestTopDownLeftRightResend](http://docs.google.com/java/awt/image/ImageProducer.html#requestTopDownLeftRightResend(java.awt.image.ImageConsumer)) in interface [ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) **Parameters:**ic - the specified ImageConsumer**See Also:**[ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html)

### setAnimated

public void **setAnimated**(boolean animated)

Changes this memory image into a multi-frame animation or a single-frame static image depending on the animated parameter.

This method should be called immediately after the MemoryImageSource is constructed and before an image is created with it to ensure that all ImageConsumers will receive the correct multi-frame data. If an ImageConsumer is added to this ImageProducer before this flag is set then that ImageConsumer will see only a snapshot of the pixel data that was available when it connected.

**Parameters:**animated - true if the image is a multi-frame animation

### setFullBufferUpdates

public void **setFullBufferUpdates**(boolean fullbuffers)

Specifies whether this animated memory image should always be updated by sending the complete buffer of pixels whenever there is a change. This flag is ignored if the animation flag is not turned on through the setAnimated() method.

This method should be called immediately after the MemoryImageSource is constructed and before an image is created with it to ensure that all ImageConsumers will receive the correct pixel delivery hints.

**Parameters:**fullbuffers - true if the complete pixel buffer should always be sent**See Also:**[setAnimated(boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#setAnimated(boolean))

### newPixels

public void **newPixels**()

Sends a whole new buffer of pixels to any ImageConsumers that are currently interested in the data for this image and notify them that an animation frame is complete. This method only has effect if the animation flag has been turned on through the setAnimated() method.

**See Also:**[newPixels(int, int, int, int, boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels(int,%20int,%20int,%20int,%20boolean)), [ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html), [setAnimated(boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#setAnimated(boolean))

### newPixels

public void **newPixels**(int x,  
 int y,  
 int w,  
 int h)

Sends a rectangular region of the buffer of pixels to any ImageConsumers that are currently interested in the data for this image and notify them that an animation frame is complete. This method only has effect if the animation flag has been turned on through the setAnimated() method. If the full buffer update flag was turned on with the setFullBufferUpdates() method then the rectangle parameters will be ignored and the entire buffer will always be sent.

**Parameters:**x - the x coordinate of the upper left corner of the rectangle of pixels to be senty - the y coordinate of the upper left corner of the rectangle of pixels to be sentw - the width of the rectangle of pixels to be senth - the height of the rectangle of pixels to be sent**See Also:**[newPixels(int, int, int, int, boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels(int,%20int,%20int,%20int,%20boolean)), [ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html), [setAnimated(boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#setAnimated(boolean)), [setFullBufferUpdates(boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#setFullBufferUpdates(boolean))

### newPixels

public void **newPixels**(int x,  
 int y,  
 int w,  
 int h,  
 boolean framenotify)

Sends a rectangular region of the buffer of pixels to any ImageConsumers that are currently interested in the data for this image. If the framenotify parameter is true then the consumers are also notified that an animation frame is complete. This method only has effect if the animation flag has been turned on through the setAnimated() method. If the full buffer update flag was turned on with the setFullBufferUpdates() method then the rectangle parameters will be ignored and the entire buffer will always be sent.

**Parameters:**x - the x coordinate of the upper left corner of the rectangle of pixels to be senty - the y coordinate of the upper left corner of the rectangle of pixels to be sentw - the width of the rectangle of pixels to be senth - the height of the rectangle of pixels to be sentframenotify - true if the consumers should be sent a [SINGLEFRAMEDONE](http://docs.google.com/java/awt/image/ImageConsumer.html#SINGLEFRAMEDONE) notification**See Also:**[ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html), [setAnimated(boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#setAnimated(boolean)), [setFullBufferUpdates(boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#setFullBufferUpdates(boolean))

### newPixels

public void **newPixels**(byte[] newpix,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) newmodel,  
 int offset,  
 int scansize)

Changes to a new byte array to hold the pixels for this image. If the animation flag has been turned on through the setAnimated() method, then the new pixels will be immediately delivered to any ImageConsumers that are currently interested in the data for this image.

**Parameters:**newpix - the new pixel arraynewmodel - the specified ColorModeloffset - the offset into the arrayscansize - the distance from one row of pixels to the next in the array**See Also:**[newPixels(int, int, int, int, boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels(int,%20int,%20int,%20int,%20boolean)), [setAnimated(boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#setAnimated(boolean))

### newPixels

public void **newPixels**(int[] newpix,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) newmodel,  
 int offset,  
 int scansize)

Changes to a new int array to hold the pixels for this image. If the animation flag has been turned on through the setAnimated() method, then the new pixels will be immediately delivered to any ImageConsumers that are currently interested in the data for this image.

**Parameters:**newpix - the new pixel arraynewmodel - the specified ColorModeloffset - the offset into the arrayscansize - the distance from one row of pixels to the next in the array**See Also:**[newPixels(int, int, int, int, boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#newPixels(int,%20int,%20int,%20int,%20boolean)), [setAnimated(boolean)](http://docs.google.com/java/awt/image/MemoryImageSource.html#setAnimated(boolean))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MemoryImageSource.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/LookupTable.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/MultiPixelPackedSampleModel.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/MemoryImageSource.html)    [**NO FRAMES**](http://docs.google.com/MemoryImageSource.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [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).