
Class java.awt.image.MemoryImageSource

```
java.lang.Object
   |
   +----java.awt.image.MemoryImageSource
```

---

public class **MemoryImageSource**
extends Object
implements ImageProducer

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));
```

See Also:
ImageProducer
Version:
1.13 09/08/95
Author:
Jim Graham

Constructor Index

o **MemoryImageSource**(int, int, ColorModel, byte[], int, int)
Construct an ImageProducer object which uses an array of bytes to produce data for an Image object.

- **MemoryImageSource**(int, int, ColorModel, byte[], int, int, Hashtable)
  Construct an ImageProducer object which uses an array of bytes to produce data for an Image object.
- **MemoryImageSource**(int, int, ColorModel, int[], int, int)
  Construct an ImageProducer object which uses an array of integers to produce data for an Image object.
- **MemoryImageSource**(int, int, ColorModel, int[], int, int, Hashtable)
  Construct an ImageProducer object which uses an array of integers to produce data for an Image object.
- **MemoryImageSource**(int, int, int[], int, int)
  Construct an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.
- **MemoryImageSource**(int, int, int[], int, int, Hashtable)
  Construct an ImageProducer object which uses an array of integers in the default RGB ColorModel to produce data for an Image object.

Method Index

- **addConsumer**(ImageConsumer)
  Add an ImageConsumer to the list of consumers interested in data for this image.
- **isConsumer**(ImageConsumer)
  Determine if an ImageConsumer is on the list of consumers currently interested in data for this image.
- **removeConsumer**(ImageConsumer)
  Remove an ImageConsumer from the list of consumers interested in data for this image.
- **requestTopDownLeftRightResend**(ImageConsumer)
  Request that a given ImageConsumer have the image data delivered one more time in top–down, left–right order.
- **startProduction**(ImageConsumer)
  Add an ImageConsumer to the list of consumers interested in data for this image, and immediately start delivery of the image data through the ImageConsumer interface.

Constructors

- **MemoryImageSource**

```
public MemoryImageSource(int w,
                         int h,
                         ColorModel cm,
                         byte pix[],
                         int off,
                         int scan)
```

Construct an ImageProducer object which uses an array of bytes to produce data for an Image object.
See Also:

createImage

- **MemoryImageSource**

```
public MemoryImageSource(int w,
                         int h,
                         ColorModel cm,
                         byte pix[],
                         int off,
                         int scan,
                         Hashtable props)
```

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

See Also:

createImage

- **MemoryImageSource**

```
public MemoryImageSource(int w,
                         int h,
                         ColorModel cm,
                         int pix[],
                         int off,
                         int scan)
```

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

See Also:

createImage

- **MemoryImageSource**

```
public MemoryImageSource(int w,
                         int h,
                         ColorModel cm,
                         int pix[],
                         int off,
                         int scan,
                         Hashtable props)
```

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

See Also:

createImage

- **MemoryImageSource**

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

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

See Also:

createImage, getRGBdefault

o **MemoryImageSource**

```
public MemoryImageSource(int w,
                         int h,
                         int pix[],
                         int off,
                         int scan,
                         Hashtable props)
```

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

See Also:

createImage, getRGBdefault

Methods

o **addConsumer**

```
public synchronized void addConsumer(ImageConsumer ic)
```

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

See Also:

ImageConsumer

o **isConsumer**

```
public synchronized boolean isConsumer(ImageConsumer ic)
```

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

Returns:

true if the ImageConsumer is on the list; false otherwise

See Also:

ImageConsumer

o **removeConsumer**

```
public synchronized void removeConsumer(ImageConsumer ic)
```

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

See Also:

ImageConsumer

- **startProduction**

```
public void startProduction(ImageConsumer ic)
```

Add an ImageConsumer to the list of consumers interested in data for this image, and immediately start delivery of the image data through the ImageConsumer interface.

See Also:

ImageConsumer

- **requestTopDownLeftRightResend**

```
public void requestTopDownLeftRightResend(ImageConsumer ic)
```

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

See Also:

ImageConsumer
