Class java.awt.image.FilteredImageSource

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

---

public class **FilteredImageSource**
extends Object
implements ImageProducer

This class is an implementation of the ImageProducer interface which takes an existing image and a filter object and uses them to produce image data for a new filtered version of the original image. Here is an example which filters an image by swapping the red and blue compents:

```
Image src = getImage("doc:///demo/images/duke/T1.gif");
ImageFilter colorfilter = new RedBlueSwapFilter();
Image img = createImage(new FilteredImageSource(src.getSource(),
                                                colorfilter));
```

See Also:
ImageProducer
Version:
1.13 10/07/95
Author:
Jim Graham

Constructor Index

o **FilteredImageSource**(ImageProducer, ImageFilter)
Construct an ImageProducer object from an existing ImageProducer and a filter object.

Method Index

o **addConsumer**(ImageConsumer)
Add an ImageConsumer to the list of consumers interested in data for this image.
o **isConsumer**(ImageConsumer)
Determine if an ImageConsumer is on the list of consumers currently interested in data for this image.

o **removeConsumer**(ImageConsumer)
Remove an ImageConsumer from the list of consumers interested in data for this image.
o **requestTopDownLeftRightResend**(ImageConsumer)
Request that a given ImageConsumer have the image data delivered one more time in top−down, left−right order.
o **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

o **FilteredImageSource**

```
public FilteredImageSource(ImageProducer orig,
                           ImageFilter imgf)
```

Construct an ImageProducer object from an existing ImageProducer and a filter object.
See Also:
ImageFilter, createImage

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

o **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

o **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