---

# Class java.awt.image.ImageFilter

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

---

public class **ImageFilter**
extends Object
implements ImageConsumer

This class implements a filter for the set of interface methods that are used to deliver data from an ImageProducer to an ImageConsumer. It is meant to be used in conjunction with a FilteredImageSource object to produce filtered versions of existing images. It is a base class that provides the calls needed to implement a "Null filter" which has no effect on the data being passed through. Filters should subclass this class and override the methods which deal with the data that needs to be filtered and modify it as necessary.

**See Also:**
FilteredImageSource, ImageConsumer

**Version:**
1.10 09/08/95

**Author:**
Jim Graham

---

# Variable Index

o **consumer**
The consumer of the particular image data stream that this instance of the ImageFilter is filtering data for.

# Constructor Index

o **ImageFilter**()

# Method Index


o **getFilterInstance**(ImageConsumer)
Return a unique instance of an ImageFilter object which will actually perform the filtering for the specified ImageConsumer.
o **imageComplete**(int)
Filter the information provided in the imageComplete method of the ImageConsumer interface.
o **setColorModel**(ColorModel)
Filter the information provided in the setColorModel method of the ImageConsumer interface.
o **setDimensions**(int, int)
Filter the information provided in the setDimensions method of the ImageConsumer interface.
o **setHints**(int)
Filter the information provided in the setHints method of the ImageConsumer interface.
o **setPixels**(int, int, int, int, ColorModel, byte[], int, int)
Filter the information provided in the setPixels method of the ImageConsumer interface which takes an array of bytes.
o **setPixels**(int, int, int, int, ColorModel, int[], int, int)
Filter the information provided in the setPixels method of the ImageConsumer interface which takes an array of integers.
o **setProperties**(Hashtable)
Pass the properties from the source object along after adding a property indicating the stream of filters it has been run through.


# Variables

o **consumer**

```
protected ImageConsumer consumer
```

The consumer of the particular image data stream that this instance of the ImageFilter is filtering data for. It is not initialized during the constructor, but rather during the getFilterInstance() method call when the FilteredImageSource is creating a unique instance of this object for a particular image data stream.
**See Also:**
getFilterInstance, ImageConsumer

# Constructors

o **ImageFilter**

```
public ImageFilter()
```

# Methods

o **getFilterInstance**

```
public ImageFilter getFilterInstance(ImageConsumer ic)
```

Return a unique instance of an ImageFilter object which will actually perform the filtering for the specified ImageConsumer. The default implementation just clones this object.

o **setDimensions**

```
public void setDimensions(int width,
                          int height)
```

Filter the information provided in the setDimensions method of the ImageConsumer interface.
**See Also:**
setDimensions

o **setProperties**

```
public void setProperties(Hashtable props)
```

Pass the properties from the source object along after adding a property indicating the stream of filters it has been run through.

o **setColorModel**

```
public void setColorModel(ColorModel model)
```

Filter the information provided in the setColorModel method of the ImageConsumer interface.
**See Also:**
setColorModel

o **setHints**

```
public void setHints(int hints)
```

Filter the information provided in the setHints method of the ImageConsumer interface.
**See Also:**
setHints

o **setPixels**

```
public void setPixels(int x,
                      int y,
                      int w,
                      int h,
                      ColorModel model,
                      byte pixels[],
                      int off,
                      int scansize)
```

Filter the information provided in the setPixels method of the ImageConsumer interface which takes an array of bytes.
**See Also:**
setPixels

o **setPixels**

```
public void setPixels(int x,
                      int y,
                      int w,
                      int h,
                      ColorModel model,
                      int pixels[],
                      int off,
                      int scansize)
```

Filter the information provided in the setPixels method of the ImageConsumer interface which takes an array of integers.

**See Also:**

setPixels

o **imageComplete**

```
public void imageComplete(int status)
```

Filter the information provided in the imageComplete method of the ImageConsumer interface.

**See Also:**

imageComplete

---