| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RGBImageFilter.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/RescaleOp.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/SampleModel.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/RGBImageFilter.html)    [**NO FRAMES**](http://docs.google.com/RGBImageFilter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#35nkun2) | [METHOD](#44sinio) |

## **java.awt.image**

Class RGBImageFilter

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

**All Implemented Interfaces:** [ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [GrayFilter](http://docs.google.com/javax/swing/GrayFilter.html)

public abstract class **RGBImageFilter**extends [ImageFilter](http://docs.google.com/java/awt/image/ImageFilter.html)

This class provides an easy way to create an ImageFilter which modifies the pixels of an image in the default RGB ColorModel. It is meant to be used in conjunction with a FilteredImageSource object to produce filtered versions of existing images. It is an abstract class that provides the calls needed to channel all of the pixel data through a single method which converts pixels one at a time in the default RGB ColorModel regardless of the ColorModel being used by the ImageProducer. The only method which needs to be defined to create a useable image filter is the filterRGB method. Here is an example of a definition of a filter which swaps the red and blue components of an image:

class RedBlueSwapFilter extends RGBImageFilter {  
 public RedBlueSwapFilter() {  
 // The filter's operation does not depend on the  
 // pixel's location, so IndexColorModels can be  
 // filtered directly.  
 canFilterIndexColorModel = true;  
 }  
  
 public int filterRGB(int x, int y, int rgb) {  
 return ((rgb & 0xff00ff00)  
 | ((rgb & 0xff0000) >> 16)  
 | ((rgb & 0xff) << 16));  
 }  
 }

**See Also:**[FilteredImageSource](http://docs.google.com/java/awt/image/FilteredImageSource.html), [ImageFilter](http://docs.google.com/java/awt/image/ImageFilter.html), [ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault())

| **Field Summary** | |
| --- | --- |
| protected  boolean | [**canFilterIndexColorModel**](http://docs.google.com/java/awt/image/RGBImageFilter.html#canFilterIndexColorModel)            This boolean indicates whether or not it is acceptable to apply the color filtering of the filterRGB method to the color table entries of an IndexColorModel object in lieu of pixel by pixel filtering. |
| protected  [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) | [**newmodel**](http://docs.google.com/java/awt/image/RGBImageFilter.html#newmodel)            The ColorModel with which to replace origmodel when the user calls substituteColorModel. |
| protected  [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) | [**origmodel**](http://docs.google.com/java/awt/image/RGBImageFilter.html#origmodel)            The ColorModel to be replaced by newmodel when the user calls [substituteColorModel](http://docs.google.com/java/awt/image/RGBImageFilter.html#substituteColorModel(java.awt.image.ColorModel,%20java.awt.image.ColorModel)). |

| **Fields inherited from class java.awt.image.**[**ImageFilter**](http://docs.google.com/java/awt/image/ImageFilter.html) |
| --- |
| [consumer](http://docs.google.com/java/awt/image/ImageFilter.html#consumer) |

| **Fields inherited from interface java.awt.image.**[**ImageConsumer**](http://docs.google.com/java/awt/image/ImageConsumer.html) |
| --- |
| [COMPLETESCANLINES](http://docs.google.com/java/awt/image/ImageConsumer.html#COMPLETESCANLINES), [IMAGEABORTED](http://docs.google.com/java/awt/image/ImageConsumer.html#IMAGEABORTED), [IMAGEERROR](http://docs.google.com/java/awt/image/ImageConsumer.html#IMAGEERROR), [RANDOMPIXELORDER](http://docs.google.com/java/awt/image/ImageConsumer.html#RANDOMPIXELORDER), [SINGLEFRAME](http://docs.google.com/java/awt/image/ImageConsumer.html#SINGLEFRAME), [SINGLEFRAMEDONE](http://docs.google.com/java/awt/image/ImageConsumer.html#SINGLEFRAMEDONE), [SINGLEPASS](http://docs.google.com/java/awt/image/ImageConsumer.html#SINGLEPASS), [STATICIMAGEDONE](http://docs.google.com/java/awt/image/ImageConsumer.html#STATICIMAGEDONE), [TOPDOWNLEFTRIGHT](http://docs.google.com/java/awt/image/ImageConsumer.html#TOPDOWNLEFTRIGHT) |

| **Constructor Summary** | |
| --- | --- |
| [**RGBImageFilter**](http://docs.google.com/java/awt/image/RGBImageFilter.html#RGBImageFilter())() |

| **Method Summary** | |
| --- | --- |
| [IndexColorModel](http://docs.google.com/java/awt/image/IndexColorModel.html) | [**filterIndexColorModel**](http://docs.google.com/java/awt/image/RGBImageFilter.html#filterIndexColorModel(java.awt.image.IndexColorModel))([IndexColorModel](http://docs.google.com/java/awt/image/IndexColorModel.html) icm)            Filters an IndexColorModel object by running each entry in its color tables through the filterRGB function that RGBImageFilter subclasses must provide. |
| abstract  int | [**filterRGB**](http://docs.google.com/java/awt/image/RGBImageFilter.html#filterRGB(int,%20int,%20int))(int x, int y, int rgb)            Subclasses must specify a method to convert a single input pixel in the default RGB ColorModel to a single output pixel. |
| void | [**filterRGBPixels**](http://docs.google.com/java/awt/image/RGBImageFilter.html#filterRGBPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20int,%20int))(int x, int y, int w, int h, int[] pixels, int off, int scansize)            Filters a buffer of pixels in the default RGB ColorModel by passing them one by one through the filterRGB method. |
| void | [**setColorModel**](http://docs.google.com/java/awt/image/RGBImageFilter.html#setColorModel(java.awt.image.ColorModel))([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) model)            If the ColorModel is an IndexColorModel and the subclass has set the canFilterIndexColorModel flag to true, we substitute a filtered version of the color model here and wherever that original ColorModel object appears in the setPixels methods. |
| void | [**setPixels**](http://docs.google.com/java/awt/image/RGBImageFilter.html#setPixels(int,%20int,%20int,%20int,%20java.awt.image.ColorModel,%20byte%5B%5D,%20int,%20int))(int x, int y, int w, int h, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) model, byte[] pixels, int off, int scansize)            If the ColorModel object is the same one that has already been converted, then simply passes the pixels through with the converted ColorModel. |
| void | [**setPixels**](http://docs.google.com/java/awt/image/RGBImageFilter.html#setPixels(int,%20int,%20int,%20int,%20java.awt.image.ColorModel,%20int%5B%5D,%20int,%20int))(int x, int y, int w, int h, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) model, int[] pixels, int off, int scansize)            If the ColorModel object is the same one that has already been converted, then simply passes the pixels through with the converted ColorModel, otherwise converts the buffer of integer pixels to the default RGB ColorModel and passes the converted buffer to the filterRGBPixels method to be converted one by one. |
| void | [**substituteColorModel**](http://docs.google.com/java/awt/image/RGBImageFilter.html#substituteColorModel(java.awt.image.ColorModel,%20java.awt.image.ColorModel))([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) oldcm, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) newcm)            Registers two ColorModel objects for substitution. |

| **Methods inherited from class java.awt.image.**[**ImageFilter**](http://docs.google.com/java/awt/image/ImageFilter.html) |
| --- |
| [clone](http://docs.google.com/java/awt/image/ImageFilter.html#clone()), [getFilterInstance](http://docs.google.com/java/awt/image/ImageFilter.html#getFilterInstance(java.awt.image.ImageConsumer)), [imageComplete](http://docs.google.com/java/awt/image/ImageFilter.html#imageComplete(int)), [resendTopDownLeftRight](http://docs.google.com/java/awt/image/ImageFilter.html#resendTopDownLeftRight(java.awt.image.ImageProducer)), [setDimensions](http://docs.google.com/java/awt/image/ImageFilter.html#setDimensions(int,%20int)), [setHints](http://docs.google.com/java/awt/image/ImageFilter.html#setHints(int)), [setProperties](http://docs.google.com/java/awt/image/ImageFilter.html#setProperties(java.util.Hashtable)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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** |
| --- |

### origmodel

protected [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **origmodel**

The ColorModel to be replaced by newmodel when the user calls [substituteColorModel](http://docs.google.com/java/awt/image/RGBImageFilter.html#substituteColorModel(java.awt.image.ColorModel,%20java.awt.image.ColorModel)).

### newmodel

protected [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **newmodel**

The ColorModel with which to replace origmodel when the user calls substituteColorModel.

### canFilterIndexColorModel

protected boolean **canFilterIndexColorModel**

This boolean indicates whether or not it is acceptable to apply the color filtering of the filterRGB method to the color table entries of an IndexColorModel object in lieu of pixel by pixel filtering. Subclasses should set this variable to true in their constructor if their filterRGB method does not depend on the coordinate of the pixel being filtered.

**See Also:**[substituteColorModel(java.awt.image.ColorModel, java.awt.image.ColorModel)](http://docs.google.com/java/awt/image/RGBImageFilter.html#substituteColorModel(java.awt.image.ColorModel,%20java.awt.image.ColorModel)), [filterRGB(int, int, int)](http://docs.google.com/java/awt/image/RGBImageFilter.html#filterRGB(int,%20int,%20int)), [IndexColorModel](http://docs.google.com/java/awt/image/IndexColorModel.html)

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

### RGBImageFilter

public **RGBImageFilter**()

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

### setColorModel

public void **setColorModel**([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) model)

If the ColorModel is an IndexColorModel and the subclass has set the canFilterIndexColorModel flag to true, we substitute a filtered version of the color model here and wherever that original ColorModel object appears in the setPixels methods. If the ColorModel is not an IndexColorModel or is null, this method overrides the default ColorModel used by the ImageProducer and specifies the default RGB ColorModel instead.

Note: This method is intended to be called by the ImageProducer of the Image whose pixels are being filtered. Developers using this class to filter pixels from an image should avoid calling this method directly since that operation could interfere with the filtering operation.

**Specified by:**[setColorModel](http://docs.google.com/java/awt/image/ImageConsumer.html#setColorModel(java.awt.image.ColorModel)) in interface [ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html)**Overrides:**[setColorModel](http://docs.google.com/java/awt/image/ImageFilter.html#setColorModel(java.awt.image.ColorModel)) in class [ImageFilter](http://docs.google.com/java/awt/image/ImageFilter.html) **Parameters:**model - the specified ColorModel**See Also:**[ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html), [ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault())

### substituteColorModel

public void **substituteColorModel**([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) oldcm,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) newcm)

Registers two ColorModel objects for substitution. If the oldcm is encountered during any of the setPixels methods, the newcm is substituted and the pixels passed through untouched (but with the new ColorModel object).

**Parameters:**oldcm - the ColorModel object to be replaced on the flynewcm - the ColorModel object to replace oldcm on the fly

### filterIndexColorModel

public [IndexColorModel](http://docs.google.com/java/awt/image/IndexColorModel.html) **filterIndexColorModel**([IndexColorModel](http://docs.google.com/java/awt/image/IndexColorModel.html) icm)

Filters an IndexColorModel object by running each entry in its color tables through the filterRGB function that RGBImageFilter subclasses must provide. Uses coordinates of -1 to indicate that a color table entry is being filtered rather than an actual pixel value.

**Parameters:**icm - the IndexColorModel object to be filtered **Returns:**a new IndexColorModel representing the filtered colors **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if icm is null

### filterRGBPixels

public void **filterRGBPixels**(int x,  
 int y,  
 int w,  
 int h,  
 int[] pixels,  
 int off,  
 int scansize)

Filters a buffer of pixels in the default RGB ColorModel by passing them one by one through the filterRGB method.

**Parameters:**x - the X coordinate of the upper-left corner of the region of pixelsy - the Y coordinate of the upper-left corner of the region of pixelsw - the width of the region of pixelsh - the height of the region of pixelspixels - the array of pixelsoff - the offset into the pixels arrayscansize - the distance from one row of pixels to the next in the array**See Also:**[ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault()), [filterRGB(int, int, int)](http://docs.google.com/java/awt/image/RGBImageFilter.html#filterRGB(int,%20int,%20int))

### setPixels

public void **setPixels**(int x,  
 int y,  
 int w,  
 int h,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) model,  
 byte[] pixels,  
 int off,  
 int scansize)

If the ColorModel object is the same one that has already been converted, then simply passes the pixels through with the converted ColorModel. Otherwise converts the buffer of byte pixels to the default RGB ColorModel and passes the converted buffer to the filterRGBPixels method to be converted one by one.

Note: This method is intended to be called by the ImageProducer of the Image whose pixels are being filtered. Developers using this class to filter pixels from an image should avoid calling this method directly since that operation could interfere with the filtering operation.

**Specified by:**[setPixels](http://docs.google.com/java/awt/image/ImageConsumer.html#setPixels(int,%20int,%20int,%20int,%20java.awt.image.ColorModel,%20byte%5B%5D,%20int,%20int)) in interface [ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html)**Overrides:**[setPixels](http://docs.google.com/java/awt/image/ImageFilter.html#setPixels(int,%20int,%20int,%20int,%20java.awt.image.ColorModel,%20byte%5B%5D,%20int,%20int)) in class [ImageFilter](http://docs.google.com/java/awt/image/ImageFilter.html) **Parameters:**x - the X coordinate of the upper-left corner of the area of pixels to be sety - the Y coordinate of the upper-left corner of the area of pixels to be setw - the width of the area of pixelsh - the height of the area of pixelsmodel - the specified ColorModelpixels - the array of pixelsoff - the offset into the pixels arrayscansize - the distance from one row of pixels to the next in the pixels array**See Also:**[ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault()), [filterRGBPixels(int, int, int, int, int[], int, int)](http://docs.google.com/java/awt/image/RGBImageFilter.html#filterRGBPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20int,%20int))

### setPixels

public void **setPixels**(int x,  
 int y,  
 int w,  
 int h,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) model,  
 int[] pixels,  
 int off,  
 int scansize)

If the ColorModel object is the same one that has already been converted, then simply passes the pixels through with the converted ColorModel, otherwise converts the buffer of integer pixels to the default RGB ColorModel and passes the converted buffer to the filterRGBPixels method to be converted one by one. Converts a buffer of integer pixels to the default RGB ColorModel and passes the converted buffer to the filterRGBPixels method.

Note: This method is intended to be called by the ImageProducer of the Image whose pixels are being filtered. Developers using this class to filter pixels from an image should avoid calling this method directly since that operation could interfere with the filtering operation.

**Specified by:**[setPixels](http://docs.google.com/java/awt/image/ImageConsumer.html#setPixels(int,%20int,%20int,%20int,%20java.awt.image.ColorModel,%20int%5B%5D,%20int,%20int)) in interface [ImageConsumer](http://docs.google.com/java/awt/image/ImageConsumer.html)**Overrides:**[setPixels](http://docs.google.com/java/awt/image/ImageFilter.html#setPixels(int,%20int,%20int,%20int,%20java.awt.image.ColorModel,%20int%5B%5D,%20int,%20int)) in class [ImageFilter](http://docs.google.com/java/awt/image/ImageFilter.html) **Parameters:**x - the X coordinate of the upper-left corner of the area of pixels to be sety - the Y coordinate of the upper-left corner of the area of pixels to be setw - the width of the area of pixelsh - the height of the area of pixelsmodel - the specified ColorModelpixels - the array of pixelsoff - the offset into the pixels arrayscansize - the distance from one row of pixels to the next in the pixels array**See Also:**[ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault()), [filterRGBPixels(int, int, int, int, int[], int, int)](http://docs.google.com/java/awt/image/RGBImageFilter.html#filterRGBPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20int,%20int))

### filterRGB

public abstract int **filterRGB**(int x,  
 int y,  
 int rgb)

Subclasses must specify a method to convert a single input pixel in the default RGB ColorModel to a single output pixel.

**Parameters:**x - the X coordinate of the pixely - the Y coordinate of the pixelrgb - the integer pixel representation in the default RGB color model **Returns:**a filtered pixel in the default RGB color model.**See Also:**[ColorModel.getRGBdefault()](http://docs.google.com/java/awt/image/ColorModel.html#getRGBdefault()), [filterRGBPixels(int, int, int, int, int[], int, int)](http://docs.google.com/java/awt/image/RGBImageFilter.html#filterRGBPixels(int,%20int,%20int,%20int,%20int%5B%5D,%20int,%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/RGBImageFilter.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/RescaleOp.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/SampleModel.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/RGBImageFilter.html)    [**NO FRAMES**](http://docs.google.com/RGBImageFilter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#35nkun2) | [METHOD](#44sinio) |

[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).