| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RescaleOp.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/ReplicateScaleFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/RGBImageFilter.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/RescaleOp.html)    [**NO FRAMES**](http://docs.google.com/RescaleOp.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **java.awt.image**

Class RescaleOp

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

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

public class **RescaleOp**extends [Object](http://docs.google.com/java/lang/Object.html)implements [BufferedImageOp](http://docs.google.com/java/awt/image/BufferedImageOp.html), [RasterOp](http://docs.google.com/java/awt/image/RasterOp.html)

This class performs a pixel-by-pixel rescaling of the data in the source image by multiplying the sample values for each pixel by a scale factor and then adding an offset. The scaled sample values are clipped to the minimum/maximum representable in the destination image.

The pseudo code for the rescaling operation is as follows:

for each pixel from Source object {  
 for each band/component of the pixel {  
 dstElement = (srcElement\*scaleFactor) + offset  
 }  
}

For Rasters, rescaling operates on bands. The number of sets of scaling constants may be one, in which case the same constants are applied to all bands, or it must equal the number of Source Raster bands.

For BufferedImages, rescaling operates on color and alpha components. The number of sets of scaling constants may be one, in which case the same constants are applied to all color (but not alpha) components. Otherwise, the number of sets of scaling constants may equal the number of Source color components, in which case no rescaling of the alpha component (if present) is performed. If neither of these cases apply, the number of sets of scaling constants must equal the number of Source color components plus alpha components, in which case all color and alpha components are rescaled.

BufferedImage sources with premultiplied alpha data are treated in the same manner as non-premultiplied images for purposes of rescaling. That is, the rescaling is done per band on the raw data of the BufferedImage source without regard to whether the data is premultiplied. If a color conversion is required to the destination ColorModel, the premultiplied state of both source and destination will be taken into account for this step.

Images with an IndexColorModel cannot be rescaled.

If a RenderingHints object is specified in the constructor, the color rendering hint and the dithering hint may be used when color conversion is required.

Note that in-place operation is allowed (i.e. the source and destination can be the same object).

**See Also:**[RenderingHints.KEY\_COLOR\_RENDERING](http://docs.google.com/java/awt/RenderingHints.html#KEY_COLOR_RENDERING), [RenderingHints.KEY\_DITHERING](http://docs.google.com/java/awt/RenderingHints.html#KEY_DITHERING)

| **Constructor Summary** | |
| --- | --- |
| [**RescaleOp**](http://docs.google.com/java/awt/image/RescaleOp.html#RescaleOp(float%5B%5D,%20float%5B%5D,%20java.awt.RenderingHints))(float[] scaleFactors, float[] offsets, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Constructs a new RescaleOp with the desired scale factors and offsets. |
| [**RescaleOp**](http://docs.google.com/java/awt/image/RescaleOp.html#RescaleOp(float,%20float,%20java.awt.RenderingHints))(float scaleFactor, float offset, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Constructs a new RescaleOp with the desired scale factor and offset. |

| **Method Summary** | |
| --- | --- |
| [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) | [**createCompatibleDestImage**](http://docs.google.com/java/awt/image/RescaleOp.html#createCompatibleDestImage(java.awt.image.BufferedImage,%20java.awt.image.ColorModel))([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) src, [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) destCM)            Creates a zeroed destination image with the correct size and number of bands. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createCompatibleDestRaster**](http://docs.google.com/java/awt/image/RescaleOp.html#createCompatibleDestRaster(java.awt.image.Raster))([Raster](http://docs.google.com/java/awt/image/Raster.html) src)            Creates a zeroed-destination Raster with the correct size and number of bands, given this source. |
| [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) | [**filter**](http://docs.google.com/java/awt/image/RescaleOp.html#filter(java.awt.image.BufferedImage,%20java.awt.image.BufferedImage))([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) src, [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) dst)            Rescales the source BufferedImage. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**filter**](http://docs.google.com/java/awt/image/RescaleOp.html#filter(java.awt.image.Raster,%20java.awt.image.WritableRaster))([Raster](http://docs.google.com/java/awt/image/Raster.html) src, [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) dst)            Rescales the pixel data in the source Raster. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getBounds2D**](http://docs.google.com/java/awt/image/RescaleOp.html#getBounds2D(java.awt.image.BufferedImage))([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) src)            Returns the bounding box of the rescaled destination image. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getBounds2D**](http://docs.google.com/java/awt/image/RescaleOp.html#getBounds2D(java.awt.image.Raster))([Raster](http://docs.google.com/java/awt/image/Raster.html) src)            Returns the bounding box of the rescaled destination Raster. |
| int | [**getNumFactors**](http://docs.google.com/java/awt/image/RescaleOp.html#getNumFactors())()            Returns the number of scaling factors and offsets used in this RescaleOp. |
| float[] | [**getOffsets**](http://docs.google.com/java/awt/image/RescaleOp.html#getOffsets(float%5B%5D))(float[] offsets)            Returns the offsets in the given array. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getPoint2D**](http://docs.google.com/java/awt/image/RescaleOp.html#getPoint2D(java.awt.geom.Point2D,%20java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) srcPt, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) dstPt)            Returns the location of the destination point given a point in the source. |
| [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) | [**getRenderingHints**](http://docs.google.com/java/awt/image/RescaleOp.html#getRenderingHints())()            Returns the rendering hints for this op. |
| float[] | [**getScaleFactors**](http://docs.google.com/java/awt/image/RescaleOp.html#getScaleFactors(float%5B%5D))(float[] scaleFactors)            Returns the scale factors in the given array. |

| **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** |
| --- |

### RescaleOp

public **RescaleOp**(float[] scaleFactors,  
 float[] offsets,  
 [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)

Constructs a new RescaleOp with the desired scale factors and offsets. The length of the scaleFactor and offset arrays must meet the restrictions stated in the class comments above. The RenderingHints argument may be null.

**Parameters:**scaleFactors - the specified scale factorsoffsets - the specified offsetshints - the specified RenderingHints, or null

### RescaleOp

public **RescaleOp**(float scaleFactor,  
 float offset,  
 [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)

Constructs a new RescaleOp with the desired scale factor and offset. The scaleFactor and offset will be applied to all bands in a source Raster and to all color (but not alpha) components in a BufferedImage. The RenderingHints argument may be null.

**Parameters:**scaleFactor - the specified scale factoroffset - the specified offsethints - the specified RenderingHints, or null

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

### getScaleFactors

public final float[] **getScaleFactors**(float[] scaleFactors)

Returns the scale factors in the given array. The array is also returned for convenience. If scaleFactors is null, a new array will be allocated.

**Parameters:**scaleFactors - the array to contain the scale factors of this RescaleOp **Returns:**the scale factors of this RescaleOp.

### getOffsets

public final float[] **getOffsets**(float[] offsets)

Returns the offsets in the given array. The array is also returned for convenience. If offsets is null, a new array will be allocated.

**Parameters:**offsets - the array to contain the offsets of this RescaleOp **Returns:**the offsets of this RescaleOp.

### getNumFactors

public final int **getNumFactors**()

Returns the number of scaling factors and offsets used in this RescaleOp.

**Returns:**the number of scaling factors and offsets of this RescaleOp.

### filter

public final [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) **filter**([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) src,  
 [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) dst)

Rescales the source BufferedImage. If the color model in the source image is not the same as that in the destination image, the pixels will be converted in the destination. If the destination image is null, a BufferedImage will be created with the source ColorModel. An IllegalArgumentException may be thrown if the number of scaling factors/offsets in this object does not meet the restrictions stated in the class comments above, or if the source image has an IndexColorModel.

**Specified by:**[filter](http://docs.google.com/java/awt/image/BufferedImageOp.html#filter(java.awt.image.BufferedImage,%20java.awt.image.BufferedImage)) in interface [BufferedImageOp](http://docs.google.com/java/awt/image/BufferedImageOp.html) **Parameters:**src - the BufferedImage to be filtereddst - the destination for the filtering operation or null **Returns:**the filtered BufferedImage. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the ColorModel of src is an IndexColorModel, or if the number of scaling factors and offsets in this RescaleOp do not meet the requirements stated in the class comments.

### filter

public final [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **filter**([Raster](http://docs.google.com/java/awt/image/Raster.html) src,  
 [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) dst)

Rescales the pixel data in the source Raster. If the destination Raster is null, a new Raster will be created. The source and destination must have the same number of bands. Otherwise, an IllegalArgumentException is thrown. Note that the number of scaling factors/offsets in this object must meet the restrictions stated in the class comments above. Otherwise, an IllegalArgumentException is thrown.

**Specified by:**[filter](http://docs.google.com/java/awt/image/RasterOp.html#filter(java.awt.image.Raster,%20java.awt.image.WritableRaster)) in interface [RasterOp](http://docs.google.com/java/awt/image/RasterOp.html) **Parameters:**src - the Raster to be filtereddst - the destination for the filtering operation or null **Returns:**the filtered WritableRaster. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if src and dst do not have the same number of bands, or if the number of scaling factors and offsets in this RescaleOp do not meet the requirements stated in the class comments.

### getBounds2D

public final [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **getBounds2D**([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) src)

Returns the bounding box of the rescaled destination image. Since this is not a geometric operation, the bounding box does not change.

**Specified by:**[getBounds2D](http://docs.google.com/java/awt/image/BufferedImageOp.html#getBounds2D(java.awt.image.BufferedImage)) in interface [BufferedImageOp](http://docs.google.com/java/awt/image/BufferedImageOp.html) **Parameters:**src - The BufferedImage to be filtered **Returns:**The Rectangle2D representing the destination image's bounding box.

### getBounds2D

public final [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **getBounds2D**([Raster](http://docs.google.com/java/awt/image/Raster.html) src)

Returns the bounding box of the rescaled destination Raster. Since this is not a geometric operation, the bounding box does not change.

**Specified by:**[getBounds2D](http://docs.google.com/java/awt/image/RasterOp.html#getBounds2D(java.awt.image.Raster)) in interface [RasterOp](http://docs.google.com/java/awt/image/RasterOp.html) **Parameters:**src - the rescaled destination Raster **Returns:**the bounds of the specified Raster.

### createCompatibleDestImage

public [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) **createCompatibleDestImage**([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) src,  
 [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) destCM)

Creates a zeroed destination image with the correct size and number of bands.

**Specified by:**[createCompatibleDestImage](http://docs.google.com/java/awt/image/BufferedImageOp.html#createCompatibleDestImage(java.awt.image.BufferedImage,%20java.awt.image.ColorModel)) in interface [BufferedImageOp](http://docs.google.com/java/awt/image/BufferedImageOp.html) **Parameters:**src - Source image for the filter operation.destCM - ColorModel of the destination. If null, the ColorModel of the source will be used. **Returns:**the zeroed-destination image.

### createCompatibleDestRaster

public [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) **createCompatibleDestRaster**([Raster](http://docs.google.com/java/awt/image/Raster.html) src)

Creates a zeroed-destination Raster with the correct size and number of bands, given this source.

**Specified by:**[createCompatibleDestRaster](http://docs.google.com/java/awt/image/RasterOp.html#createCompatibleDestRaster(java.awt.image.Raster)) in interface [RasterOp](http://docs.google.com/java/awt/image/RasterOp.html) **Parameters:**src - the source Raster **Returns:**the zeroed-destination Raster.

### getPoint2D

public final [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) **getPoint2D**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) srcPt,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) dstPt)

Returns the location of the destination point given a point in the source. If dstPt is non-null, it will be used to hold the return value. Since this is not a geometric operation, the srcPt will equal the dstPt.

**Specified by:**[getPoint2D](http://docs.google.com/java/awt/image/BufferedImageOp.html#getPoint2D(java.awt.geom.Point2D,%20java.awt.geom.Point2D)) in interface [BufferedImageOp](http://docs.google.com/java/awt/image/BufferedImageOp.html)**Specified by:**[getPoint2D](http://docs.google.com/java/awt/image/RasterOp.html#getPoint2D(java.awt.geom.Point2D,%20java.awt.geom.Point2D)) in interface [RasterOp](http://docs.google.com/java/awt/image/RasterOp.html) **Parameters:**srcPt - a point in the source imagedstPt - the destination point or null **Returns:**the location of the destination point.

### getRenderingHints

public final [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) **getRenderingHints**()

Returns the rendering hints for this op.

**Specified by:**[getRenderingHints](http://docs.google.com/java/awt/image/BufferedImageOp.html#getRenderingHints()) in interface [BufferedImageOp](http://docs.google.com/java/awt/image/BufferedImageOp.html)**Specified by:**[getRenderingHints](http://docs.google.com/java/awt/image/RasterOp.html#getRenderingHints()) in interface [RasterOp](http://docs.google.com/java/awt/image/RasterOp.html) **Returns:**the rendering hints of this RescaleOp.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RescaleOp.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/ReplicateScaleFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/RGBImageFilter.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/RescaleOp.html)    [**NO FRAMES**](http://docs.google.com/RescaleOp.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[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).