| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LookupOp.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/Kernel.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/LookupTable.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/LookupOp.html)    [**NO FRAMES**](http://docs.google.com/LookupOp.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.awt.image**

Class LookupOp

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

**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 **LookupOp**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 implements a lookup operation from the source to the destination. The LookupTable object may contain a single array or multiple arrays, subject to the restrictions below.

For Rasters, the lookup operates on bands. The number of lookup arrays may be one, in which case the same array is applied to all bands, or it must equal the number of Source Raster bands.

For BufferedImages, the lookup operates on color and alpha components. The number of lookup arrays may be one, in which case the same array is applied to all color (but not alpha) components. Otherwise, the number of lookup arrays may equal the number of Source color components, in which case no lookup of the alpha component (if present) is performed. If neither of these cases apply, the number of lookup arrays must equal the number of Source color components plus alpha components, in which case lookup is performed for all color and alpha components. This allows non-uniform rescaling of multi-band BufferedImages.

BufferedImage sources with premultiplied alpha data are treated in the same manner as non-premultiplied images for purposes of the lookup. That is, the lookup 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 used.

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.

This class allows the Source to be the same as the Destination.

**See Also:**[LookupTable](http://docs.google.com/java/awt/image/LookupTable.html), [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** | |
| --- | --- |
| [**LookupOp**](http://docs.google.com/java/awt/image/LookupOp.html#LookupOp(java.awt.image.LookupTable,%20java.awt.RenderingHints))([LookupTable](http://docs.google.com/java/awt/image/LookupTable.html) lookup, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Constructs a LookupOp object given the lookup table and a RenderingHints object, which might be null. |

| **Method Summary** | |
| --- | --- |
| [BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) | [**createCompatibleDestImage**](http://docs.google.com/java/awt/image/LookupOp.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/LookupOp.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/LookupOp.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)            Performs a lookup operation on a BufferedImage. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**filter**](http://docs.google.com/java/awt/image/LookupOp.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)            Performs a lookup operation on a Raster. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getBounds2D**](http://docs.google.com/java/awt/image/LookupOp.html#getBounds2D(java.awt.image.BufferedImage))([BufferedImage](http://docs.google.com/java/awt/image/BufferedImage.html) src)            Returns the bounding box of the filtered destination image. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getBounds2D**](http://docs.google.com/java/awt/image/LookupOp.html#getBounds2D(java.awt.image.Raster))([Raster](http://docs.google.com/java/awt/image/Raster.html) src)            Returns the bounding box of the filtered destination Raster. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getPoint2D**](http://docs.google.com/java/awt/image/LookupOp.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/LookupOp.html#getRenderingHints())()            Returns the rendering hints for this op. |
| [LookupTable](http://docs.google.com/java/awt/image/LookupTable.html) | [**getTable**](http://docs.google.com/java/awt/image/LookupOp.html#getTable())()            Returns the LookupTable. |

| **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** |
| --- |

### LookupOp

public **LookupOp**([LookupTable](http://docs.google.com/java/awt/image/LookupTable.html) lookup,  
 [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)

Constructs a LookupOp object given the lookup table and a RenderingHints object, which might be null.

**Parameters:**lookup - the specified LookupTablehints - the specified RenderingHints, or null

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

### getTable

public final [LookupTable](http://docs.google.com/java/awt/image/LookupTable.html) **getTable**()

Returns the LookupTable.

**Returns:**the LookupTable of this LookupOp.

### 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)

Performs a lookup operation on a 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 an appropriate ColorModel. An IllegalArgumentException might be thrown if the number of arrays in the LookupTable does not meet the restrictions stated in the class comment 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 BufferedImage in which to store the results of the filter operation **Returns:**the filtered BufferedImage. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the number of arrays in the LookupTable does not meet the restrictions described in the class comments, or if the source image has an IndexColorModel.

### 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)

Performs a lookup operation on a Raster. If the destination Raster is null, a new Raster will be created. The IllegalArgumentException might be thrown if the source Raster and the destination Raster do not have the same number of bands or if the number of arrays in the LookupTable does not meet the restrictions stated in the class comment above.

**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 source Raster to filterdst - the destination WritableRaster for the filtered src **Returns:**the filtered WritableRaster. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the source and destinations rasters do not have the same number of bands, or the number of arrays in the LookupTable does not meet the restrictions described 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 filtered 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 bounds of the filtered definition image.

### 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 filtered 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 Raster to be filtered **Returns:**the bounds of the filtered definition 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. If destCM is null, an appropriate ColorModel will be used.

**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 - the destination's ColorModel, which can be null. **Returns:**a filtered destination BufferedImage.

### 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 Raster to be transformed **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 not 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 Point2D that represents a point in the source imagedstPt - a Point2Dthat represents the location in the destination **Returns:**the Point2D in the destination that corresponds to the specified point in the source.

### 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 RenderingHints object associated with this op.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LookupOp.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/Kernel.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/LookupTable.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/LookupOp.html)    [**NO FRAMES**](http://docs.google.com/LookupOp.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).