| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BandCombineOp.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/AreaAveragingScaleFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/BandedSampleModel.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/BandCombineOp.html)    [**NO FRAMES**](http://docs.google.com/BandCombineOp.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 BandCombineOp

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

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

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

This class performs an arbitrary linear combination of the bands in a Raster, using a specified matrix.

The width of the matrix must be equal to the number of bands in the source Raster, optionally plus one. If there is one more column in the matrix than the number of bands, there is an implied 1 at the end of the vector of band samples representing a pixel. The height of the matrix must be equal to the number of bands in the destination.

For example, a 3-banded Raster might have the following transformation applied to each pixel in order to invert the second band of the Raster.

[ 1.0 0.0 0.0 0.0 ] [ b1 ]   
 [ 0.0 -1.0 0.0 255.0 ] x [ b2 ]  
 [ 0.0 0.0 1.0 0.0 ] [ b3 ]  
 [ 1 ]

Note that the source and destination can be the same object.

| **Constructor Summary** | |
| --- | --- |
| [**BandCombineOp**](http://docs.google.com/java/awt/image/BandCombineOp.html#BandCombineOp(float%5B%5D%5B%5D,%20java.awt.RenderingHints))(float[][] matrix, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Constructs a BandCombineOp with the specified matrix. |

| **Method Summary** | |
| --- | --- |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**createCompatibleDestRaster**](http://docs.google.com/java/awt/image/BandCombineOp.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. |
| [WritableRaster](http://docs.google.com/java/awt/image/WritableRaster.html) | [**filter**](http://docs.google.com/java/awt/image/BandCombineOp.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)            Transforms the Raster using the matrix specified in the constructor. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getBounds2D**](http://docs.google.com/java/awt/image/BandCombineOp.html#getBounds2D(java.awt.image.Raster))([Raster](http://docs.google.com/java/awt/image/Raster.html) src)            Returns the bounding box of the transformed destination. |
| float[][] | [**getMatrix**](http://docs.google.com/java/awt/image/BandCombineOp.html#getMatrix())()            Returns a copy of the linear combination matrix. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getPoint2D**](http://docs.google.com/java/awt/image/BandCombineOp.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 corresponding destination point given a point in the source Raster. |
| [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) | [**getRenderingHints**](http://docs.google.com/java/awt/image/BandCombineOp.html#getRenderingHints())()            Returns the rendering hints for this operation. |

| **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** |
| --- |

### BandCombineOp

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

Constructs a BandCombineOp with the specified matrix. The width of the matrix must be equal to the number of bands in the source Raster, optionally plus one. If there is one more column in the matrix than the number of bands, there is an implied 1 at the end of the vector of band samples representing a pixel. The height of the matrix must be equal to the number of bands in the destination.

The first subscript is the row index and the second is the column index. This operation uses none of the currently defined rendering hints; the RenderingHints argument can be null.

**Parameters:**matrix - The matrix to use for the band combine operation.hints - The RenderingHints object for this operation. Not currently used so it can be null.

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

### getMatrix

public final float[][] **getMatrix**()

Returns a copy of the linear combination matrix.

**Returns:**The matrix associated with this band combine operation.

### filter

public [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)

Transforms the Raster using the matrix specified in the constructor. An IllegalArgumentException may be thrown if the number of bands in the source or destination is incompatible with the matrix. See the class comments for more details.

If the destination is null, it will be created with a number of bands equalling the number of rows in the matrix. No exception is thrown if the operation causes a data overflow.

**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 filtered.dst - The Raster in which to store the results of the filter operation. **Returns:**The filtered Raster. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the number of bands in the source or destination is incompatible with the matrix.

### 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 transformed destination. Since this is not a geometric operation, the bounding box is the same for the source and destination. An IllegalArgumentException may be thrown if the number of bands in the source is incompatible with the matrix. See the class comments for more details.

**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 Rectangle2D representing the destination image's bounding box. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the number of bands in the source is incompatible with the matrix.

### 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. An IllegalArgumentException may be thrown if the number of bands in the source is incompatible with the matrix. See the class comments for more details.

**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 filtered. **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 corresponding destination point given a point in the source Raster. If dstPt is specified, it is used to hold the return value. Since this is not a geometric operation, the point returned is the same as the specified srcPt.

**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 - The Point2D that represents the point in the source RasterdstPt - The Point2D in which to store the result. **Returns:**The Point2D in the destination image that corresponds to the specified point in the source image.

### getRenderingHints

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

Returns the rendering hints for this operation.

**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 operation. Returns null if no hints have been set.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BandCombineOp.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/AreaAveragingScaleFilter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/BandedSampleModel.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/BandCombineOp.html)    [**NO FRAMES**](http://docs.google.com/BandCombineOp.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).