| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RenderableImageOp.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/renderable/RenderableImage.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/renderable/RenderableImageProducer.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/renderable/RenderableImageOp.html)    [**NO FRAMES**](http://docs.google.com/RenderableImageOp.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

## **java.awt.image.renderable**

Class RenderableImageOp

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

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

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

This class handles the renderable aspects of an operation with help from its associated instance of a ContextualRenderedImageFactory.

| **Field Summary** | |
| --- | --- |

| **Fields inherited from interface java.awt.image.renderable.**[**RenderableImage**](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) |
| --- |
| [HINTS\_OBSERVED](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#HINTS_OBSERVED) |

| **Constructor Summary** | |
| --- | --- |
| [**RenderableImageOp**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#RenderableImageOp(java.awt.image.renderable.ContextualRenderedImageFactory,%20java.awt.image.renderable.ParameterBlock))([ContextualRenderedImageFactory](http://docs.google.com/java/awt/image/renderable/ContextualRenderedImageFactory.html) CRIF, [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) paramBlock)            Constructs a RenderedImageOp given a ContextualRenderedImageFactory object, and a ParameterBlock containing RenderableImage sources and other parameters. |

| **Method Summary** | |
| --- | --- |
| [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) | [**createDefaultRendering**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#createDefaultRendering())()            Gets a RenderedImage instance of this image with a default width and height in pixels. |
| [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) | [**createRendering**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#createRendering(java.awt.image.renderable.RenderContext))([RenderContext](http://docs.google.com/java/awt/image/renderable/RenderContext.html) renderContext)            Creates a RenderedImage which represents this RenderableImageOp (including its Renderable sources) rendered according to the given RenderContext. |
| [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) | [**createScaledRendering**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#createScaledRendering(int,%20int,%20java.awt.RenderingHints))(int w, int h, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Creates a RenderedImage instance of this image with width w, and height h in pixels. |
| float | [**getHeight**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getHeight())()            Gets the height in user coordinate space. |
| float | [**getMinX**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getMinX())()            Gets the minimum X coordinate of the rendering-independent image data. |
| float | [**getMinY**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getMinY())()            Gets the minimum Y coordinate of the rendering-independent image data. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**getParameterBlock**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getParameterBlock())()            Returns a reference to the current parameter block. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getProperty(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Gets a property from the property set of this image. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getPropertyNames**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getPropertyNames())()            Return a list of names recognized by getProperty. |
| [Vector](http://docs.google.com/java/util/Vector.html)<[RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html)> | [**getSources**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getSources())()            Returns a vector of RenderableImages that are the sources of image data for this RenderableImage. |
| float | [**getWidth**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getWidth())()            Gets the width in user coordinate space. |
| boolean | [**isDynamic**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#isDynamic())()            Returns true if successive renderings (that is, calls to createRendering() or createScaledRendering()) with the same arguments may produce different results. |
| [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) | [**setParameterBlock**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#setParameterBlock(java.awt.image.renderable.ParameterBlock))([ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) paramBlock)            Change the current ParameterBlock of the operation, allowing editing of image rendering chains. |

| **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** |
| --- |

### RenderableImageOp

public **RenderableImageOp**([ContextualRenderedImageFactory](http://docs.google.com/java/awt/image/renderable/ContextualRenderedImageFactory.html) CRIF,  
 [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) paramBlock)

Constructs a RenderedImageOp given a ContextualRenderedImageFactory object, and a ParameterBlock containing RenderableImage sources and other parameters. Any RenderedImage sources referenced by the ParameterBlock will be ignored.

**Parameters:**CRIF - a ContextualRenderedImageFactory objectparamBlock - a ParameterBlock containing this operation's source images and other parameters necessary for the operation to run.

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

### getSources

public [Vector](http://docs.google.com/java/util/Vector.html)<[RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html)> **getSources**()

Returns a vector of RenderableImages that are the sources of image data for this RenderableImage. Note that this method may return an empty vector, to indicate that the image has no sources, or null, to indicate that no information is available.

**Specified by:**[getSources](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getSources()) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Returns:**a (possibly empty) Vector of RenderableImages, or null.

### getProperty

public [Object](http://docs.google.com/java/lang/Object.html) **getProperty**([String](http://docs.google.com/java/lang/String.html) name)

Gets a property from the property set of this image. If the property name is not recognized, java.awt.Image.UndefinedProperty will be returned.

**Specified by:**[getProperty](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getProperty(java.lang.String)) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Parameters:**name - the name of the property to get, as a String. **Returns:**a reference to the property Object, or the value java.awt.Image.UndefinedProperty.

### getPropertyNames

public [String](http://docs.google.com/java/lang/String.html)[] **getPropertyNames**()

Return a list of names recognized by getProperty.

**Specified by:**[getPropertyNames](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getPropertyNames()) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Returns:**a list of property names.

### isDynamic

public boolean **isDynamic**()

Returns true if successive renderings (that is, calls to createRendering() or createScaledRendering()) with the same arguments may produce different results. This method may be used to determine whether an existing rendering may be cached and reused. The CRIF's isDynamic method will be called.

**Specified by:**[isDynamic](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#isDynamic()) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Returns:**true if successive renderings with the same arguments might produce different results; false otherwise.

### getWidth

public float **getWidth**()

Gets the width in user coordinate space. By convention, the usual width of a RenderableImage is equal to the image's aspect ratio (width divided by height).

**Specified by:**[getWidth](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getWidth()) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Returns:**the width of the image in user coordinates.

### getHeight

public float **getHeight**()

Gets the height in user coordinate space. By convention, the usual height of a RenderedImage is equal to 1.0F.

**Specified by:**[getHeight](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getHeight()) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Returns:**the height of the image in user coordinates.

### getMinX

public float **getMinX**()

Gets the minimum X coordinate of the rendering-independent image data.

**Specified by:**[getMinX](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getMinX()) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Returns:**the minimum X coordinate of the rendering-independent image data.

### getMinY

public float **getMinY**()

Gets the minimum Y coordinate of the rendering-independent image data.

**Specified by:**[getMinY](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getMinY()) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Returns:**the minimum Y coordinate of the rendering-independent image data.

### setParameterBlock

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **setParameterBlock**([ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) paramBlock)

Change the current ParameterBlock of the operation, allowing editing of image rendering chains. The effects of such a change will be visible when a new rendering is created from this RenderableImageOp or any dependent RenderableImageOp.

**Parameters:**paramBlock - the new ParameterBlock. **Returns:**the old ParameterBlock.**See Also:**[getParameterBlock()](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#getParameterBlock())

### getParameterBlock

public [ParameterBlock](http://docs.google.com/java/awt/image/renderable/ParameterBlock.html) **getParameterBlock**()

Returns a reference to the current parameter block.

**Returns:**the ParameterBlock of this RenderableImageOp.**See Also:**[setParameterBlock(ParameterBlock)](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html#setParameterBlock(java.awt.image.renderable.ParameterBlock))

### createScaledRendering

public [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) **createScaledRendering**(int w,  
 int h,  
 [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)

Creates a RenderedImage instance of this image with width w, and height h in pixels. The RenderContext is built automatically with an appropriate usr2dev transform and an area of interest of the full image. All the rendering hints come from hints passed in.

If w == 0, it will be taken to equal Math.round(h\*(getWidth()/getHeight())). Similarly, if h == 0, it will be taken to equal Math.round(w\*(getHeight()/getWidth())). One of w or h must be non-zero or else an IllegalArgumentException will be thrown.

The created RenderedImage may have a property identified by the String HINTS\_OBSERVED to indicate which RenderingHints were used to create the image. In addition any RenderedImages that are obtained via the getSources() method on the created RenderedImage may have such a property.

**Specified by:**[createScaledRendering](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#createScaledRendering(int,%20int,%20java.awt.RenderingHints)) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Parameters:**w - the width of rendered image in pixels, or 0.h - the height of rendered image in pixels, or 0.hints - a RenderingHints object containg hints. **Returns:**a RenderedImage containing the rendered data.

### createDefaultRendering

public [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) **createDefaultRendering**()

Gets a RenderedImage instance of this image with a default width and height in pixels. The RenderContext is built automatically with an appropriate usr2dev transform and an area of interest of the full image. All the rendering hints come from hints passed in. Implementors of this interface must be sure that there is a defined default width and height.

**Specified by:**[createDefaultRendering](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#createDefaultRendering()) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Returns:**a RenderedImage containing the rendered data.

### createRendering

public [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) **createRendering**([RenderContext](http://docs.google.com/java/awt/image/renderable/RenderContext.html) renderContext)

Creates a RenderedImage which represents this RenderableImageOp (including its Renderable sources) rendered according to the given RenderContext.

This method supports chaining of either Renderable or RenderedImage operations. If sources in the ParameterBlock used to construct the RenderableImageOp are RenderableImages, then a three step process is followed:

1. mapRenderContext() is called on the associated CRIF for each RenderableImage source;
2. createRendering() is called on each of the RenderableImage sources using the backwards-mapped RenderContexts obtained in step 1, resulting in a rendering of each source;
3. ContextualRenderedImageFactory.create() is called with a new ParameterBlock containing the parameters of the RenderableImageOp and the RenderedImages that were created by the createRendering() calls.

If the elements of the source Vector of the ParameterBlock used to construct the RenderableImageOp are instances of RenderedImage, then the CRIF.create() method is called immediately using the original ParameterBlock. This provides a basis case for the recursion.

The created RenderedImage may have a property identified by the String HINTS\_OBSERVED to indicate which RenderingHints (from the RenderContext) were used to create the image. In addition any RenderedImages that are obtained via the getSources() method on the created RenderedImage may have such a property.

**Specified by:**[createRendering](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#createRendering(java.awt.image.renderable.RenderContext)) in interface [RenderableImage](http://docs.google.com/java/awt/image/renderable/RenderableImage.html) **Parameters:**renderContext - The RenderContext to use to perform the rendering. **Returns:**a RenderedImage containing the desired output image.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RenderableImageOp.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/renderable/RenderableImage.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/renderable/RenderableImageProducer.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/renderable/RenderableImageOp.html)    [**NO FRAMES**](http://docs.google.com/RenderableImageOp.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

[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).