| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RenderableImage.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/ParameterBlock.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/renderable/RenderableImage.html)    [**NO FRAMES**](http://docs.google.com/RenderableImage.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#tyjcwt) | CONSTR | [METHOD](#1t3h5sf) |

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

Interface RenderableImage

**All Known Implementing Classes:** [RenderableImageOp](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html)

public interface **RenderableImage**

A RenderableImage is a common interface for rendering-independent images (a notion which subsumes resolution independence). That is, images which are described and have operations applied to them independent of any specific rendering of the image. For example, a RenderableImage can be rotated and cropped in resolution-independent terms. Then, it can be rendered for various specific contexts, such as a draft preview, a high-quality screen display, or a printer, each in an optimal fashion.

A RenderedImage is returned from a RenderableImage via the createRendering() method, which takes a RenderContext. The RenderContext specifies how the RenderedImage should be constructed. Note that it is not possible to extract pixels directly from a RenderableImage.

The createDefaultRendering() and createScaledRendering() methods are convenience methods that construct an appropriate RenderContext internally. All of the rendering methods may return a reference to a previously produced rendering.

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**HINTS\_OBSERVED**](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#HINTS_OBSERVED)            String constant that can be used to identify a property on a RenderedImage obtained via the createRendering or createScaledRendering methods. |

| **Method Summary** | |
| --- | --- |
| [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) | [**createDefaultRendering**](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#createDefaultRendering())()            Returnd 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/RenderableImage.html#createRendering(java.awt.image.renderable.RenderContext))([RenderContext](http://docs.google.com/java/awt/image/renderable/RenderContext.html) renderContext)            Creates a RenderedImage that represented a rendering of this image using a given RenderContext. |
| [RenderedImage](http://docs.google.com/java/awt/image/RenderedImage.html) | [**createScaledRendering**](http://docs.google.com/java/awt/image/renderable/RenderableImage.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/RenderableImage.html#getHeight())()            Gets the height in user coordinate space. |
| float | [**getMinX**](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getMinX())()            Gets the minimum X coordinate of the rendering-independent image data. |
| float | [**getMinY**](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#getMinY())()            Gets the minimum Y coordinate of the rendering-independent image data. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/java/awt/image/renderable/RenderableImage.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/RenderableImage.html#getPropertyNames())()            Returns 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/RenderableImage.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/RenderableImage.html#getWidth())()            Gets the width in user coordinate space. |
| boolean | [**isDynamic**](http://docs.google.com/java/awt/image/renderable/RenderableImage.html#isDynamic())()            Returns true if successive renderings (that is, calls to createRendering() or createScaledRendering()) with the same arguments may produce different results. |

| **Field Detail** |
| --- |

### HINTS\_OBSERVED

static final [String](http://docs.google.com/java/lang/String.html) **HINTS\_OBSERVED**

String constant that can be used to identify a property on a RenderedImage obtained via the createRendering or createScaledRendering methods. If such a property exists, the value of the propoery will be a RenderingHints object specifying which hints were observed in creating the rendering.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.image.renderable.RenderableImage.HINTS_OBSERVED)

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

### getSources

[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.

**Returns:**a (possibly empty) Vector of RenderableImages, or null.

### getProperty

[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.

**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

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

Returns a list of names recognized by getProperty.

**Returns:**a list of property names.

### isDynamic

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. It is always safe to return true.

**Returns:**true if successive renderings with the same arguments might produce different results; false otherwise.

### getWidth

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).

**Returns:**the width of the image in user coordinates.

### getHeight

float **getHeight**()

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

**Returns:**the height of the image in user coordinates.

### getMinX

float **getMinX**()

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

**Returns:**the minimum X coordinate of the rendering-independent image data.

### getMinY

float **getMinY**()

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

**Returns:**the minimum Y coordinate of the rendering-independent image data.

### createScaledRendering

[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.

**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

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

Returnd 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. The rendering hints are empty. createDefaultRendering may make use of a stored rendering for speed.

**Returns:**a RenderedImage containing the rendered data.

### createRendering

[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 that represented a rendering of this image using a given RenderContext. This is the most general way to obtain a rendering of a RenderableImage.

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.

**Parameters:**renderContext - the RenderContext to use to produce the rendering. **Returns:**a RenderedImage containing the rendered data.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RenderableImage.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/ParameterBlock.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/image/renderable/RenderableImageOp.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/image/renderable/RenderableImage.html)    [**NO FRAMES**](http://docs.google.com/RenderableImage.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#tyjcwt) | CONSTR | [METHOD](#1t3h5sf) |

[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).