

[Overview](#) [Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)[Summary: Nested | Field | Constr | Method](#) [Detail: Field | Constr | Method](#)

javafx.scene.image

Class WritableImage

java.lang.Object

javafx.scene.image.Image

javafx.scene.image.WritableImage

```
public class WritableImage
extends Image
```

The WritableImage class represents a custom graphical image that is constructed from pixels supplied by the application, and possibly from PixelReader objects from any number of sources, including images read from a file or URL.

Since:

2.2

Property Summary

Properties inherited from class javafx.scene.image.Image

error, height, progress, width

Constructor Summary

Constructors

Constructor and Description

WritableImage(int width, int height)

Construct an empty image of the specified dimensions.

WritableImage(PixelReader reader, int width, int height)

Construct an image of the specified dimensions, initialized from the indicated PixelReader.

WritableImage(PixelReader reader, int x, int y, int width, int height)

Construct an image of the specified dimensions, initialized from the indicated region of the PixelReader.

Method Summary

Methods

Modifier and Type	Method and Description
PixelWriter	getPixelWriter () This method returns a PixelWriter that provides access to write the pixels of the image.

Methods inherited from class javafx.scene.image.Image

```
cancel, errorProperty, getHeight, getPixelReader, getProgress, getRequestedHeight,
getRequestedWidth, getWidth, heightProperty, isBackgroundLoading, isError, isPreserveRatio,
isSmooth, progressProperty, widthProperty
```

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

WritableImage

```
public WritableImage(int width,
                    int height)
```

Construct an empty image of the specified dimensions. The image will initially be filled with transparent pixels. Images constructed this way will always be readable and writable so the corresponding `getPixelReader()` and `getPixelWriter()` will always return valid objects. The dimensions must both be positive numbers (> 0).

Parameters:

width - the desired width of the writable image
height - the desired height of the desired image

Throws:

`java.lang.IllegalArgumentException` - if either dimension is negative or zero.

WritableImage

```
public WritableImage(PixelReader reader,
                    int width,
                    int height)
```

Construct an image of the specified dimensions, initialized from the indicated `PixelReader`. The image will initially be filled with data returned from the `PixelReader`. If the `PixelReader` accesses a surface that does not contain the necessary number of pixel rows and columns then an `ArrayIndexOutOfBoundsException` will be thrown. Images constructed this way will always be readable and writable so the corresponding `getPixelReader()` and `getPixelWriter()` will always return valid objects. The dimensions must both be positive numbers (> 0).

Parameters:

width - the desired width of the writable image and the width of the region to be read from the reader
height - the desired height of the desired image and the width of the region to be read from the reader

Throws:

`java.lang.ArrayIndexOutOfBoundsException` - if the reader does not access a surface of at least the requested dimensions
`java.lang.IllegalArgumentException` - if either dimension is negative or zero.

WritableImage

```
public WritableImage(PixelReader reader,
                    int x,
                    int y,
```

```
int width,  
int height)
```

Construct an image of the specified dimensions, initialized from the indicated region of the [PixelReader](#). The image will initially be filled with data returned from the [PixelReader](#) for the specified region. If the [PixelReader](#) accesses a surface that does not contain the necessary number of pixel rows and columns then an [ArrayIndexOutOfBoundsException](#) will be thrown. Images constructed this way will always be readable and writable so the corresponding [getPixelReader\(\)](#) and [getPixelWriter\(\)](#) will always return valid objects. The dimensions must both be positive numbers (> 0).

Parameters:

x - the X coordinate of the upper left corner of the region to read from the reader

y - the Y coordinate of the upper left corner of the region to read from the reader

width - the desired width of the writable image and the width of the region to be read from the reader

height - the desired height of the desired image and the width of the region to be read from the reader

Throws:

[java.lang.ArrayIndexOutOfBoundsException](#) - if the reader does not access a surface containing at least the indicated region

[java.lang.IllegalArgumentException](#) - if either dimension is negative or zero.

Method Detail

getPixelWriter

```
public final PixelWriter getPixelWriter()
```

This method returns a [PixelWriter](#) that provides access to write the pixels of the image.

Returns:

the [PixelWriter](#) for writing pixels to the image

[Overview](#) [Package](#) **[Class](#)** [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

JavaFX 2.2

[Prev Class](#) **[Next Class](#)** [Frames](#) [No Frames](#) [All Classes](#)

[Summary: Nested](#) | [Field](#) | [Constr](#) | **[Method](#)** [Detail: Field](#) | [Constr](#) | [Method](#)

Copyright (c) 2008, 2014, Oracle and/or its affiliates. All rights reserved.