

# Package ‘RBioFormats’

February 11, 2025

**Version** 1.6.0

**BioFormats** 7.3.0

**Title** R interface to Bio-Formats

**Description** An R package which interfaces the OME Bio-Formats Java library to allow reading of proprietary microscopy image data and metadata.

**Encoding** UTF-8

**biocViews** DataImport

**URL** <https://github.com/aoles/RBioFormats>

**BugReports** <https://github.com/aoles/RBioFormats/issues>

**Imports** EBImage, methods, rJava (>= 0.9-6), S4Vectors, stats

**Suggests** BiocStyle, knitr, testthat, xml2

**SystemRequirements** Java (>= 1.7)

**License** GPL-3

**VignetteBuilder** knitr

**Collate** 'ImageMetadata.R' 'AnnotatedImage.R' 'RBioFormats.R'  
'metadataAccessors.R' 'mockFile.R' 'read.image.R'  
'read.metadata.R' 'read.omexml.R' 'utils.R' 'write.image.R'  
'zzz.R'

**RoxygenNote** 7.2.3

**git\_url** <https://git.bioconductor.org/packages/RBioFormats>

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** dcd6c87

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2025-02-10

**Author** Andrzej Oleś [aut, cre] (<<https://orcid.org/0000-0003-0285-2787>>),  
John Lee [ctb] (<<https://orcid.org/0000-0001-5884-4247>>)

**Maintainer** Andrzej Oleś <[andrzej.oles@gmail.com](mailto:andrzej.oles@gmail.com)>

## Contents

AnnotatedImage-class . . . . .	2
AnnotatedImageList-class . . . . .	3
checkJavaMemory . . . . .	4
coreMetadata . . . . .	4
dimorder . . . . .	6
FormatTools . . . . .	7
ImageMetadata-class . . . . .	7
mockFile . . . . .	8
RBioFormats . . . . .	9
read.image . . . . .	10
read.metadata . . . . .	11
read.omexml . . . . .	12
seriesCount . . . . .	13
write.image . . . . .	14
<b>Index</b>	<b>15</b>

---

AnnotatedImage-class    *AnnotatedImage Class*

---

### Description

Extends the [Image](#) class from the **EBImage** package.

### Usage

```
AnnotatedImage(..., metadata = ImageMetadata())

## S3 method for class 'AnnotatedImage'
print(x, short = FALSE, ...)

## S4 method for signature 'AnnotatedImage'
show(object)

## S3 method for class 'AnnotatedImage'
as.Image(x)
```

### Arguments

...	arguments passed to the <a href="#">Image</a> constructor.
metadata	an <a href="#">ImageMetadata</a> object containing image metadata
x	an <a href="#">AnnotatedImage</a> object.
short	logical, turns off image data preview.
object	an <a href="#">AnnotatedImage</a> object

### Value

AnnotatedImage returns a new [AnnotatedImage](#) object.  
as.Image returns an [Image](#) object.

### Slots

metadata an [ImageMetadata](#) object containing image metadata

### Author(s)

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

### See Also

[AnnotatedImageList](#)

### Examples

```
f = mockFile(sizeX = 256, sizeY = 256)
img = read.image(f)
img
```

---

AnnotatedImageList-class

*AnnotatedImageList Class*

---

### Description

A list of [AnnotatedImage](#) objects.

### Usage

```
AnnotatedImageList(...)
```

### Arguments

... a list of [AnnotatedImage](#) objects to include in the new object.

### Value

AnnotatedImageList returns a new [AnnotatedImageList](#) object.

### Examples

```
f = mockFile(sizeX = 256, sizeY = 256, series = 2)
img = read.image(f)
img
```

checkJavaMemory      *Java Memory Settings*

---

**Description**

Information about the Java heap space usage.

**Usage**

```
checkJavaMemory(units = "m")
```

**Arguments**

units                  Units to return the size in: "k", "m" or "g"

**Value**

The maximum amount of memory that the JVM will attempt to use, measured in units.

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**Examples**

```
## Not run:
## assign 4 gigabytes of heap space to the Java environment.
options( java.parameters = "-Xmx4g" )
library( "RBioFormats" )

## End(Not run)

checkJavaMemory()
```

---

coreMetadata              *Metadata Accessors*

---

**Description**

Get and set image metadata.

**Usage**

```
coreMetadata(x, series, ...)

globalMetadata(x, series, ...)

seriesMetadata(x, series, ...)

## S4 method for signature 'ANY'
metadata(x)
```

```

## S4 method for signature 'AnnotatedImage'
metadata(x)

## S4 method for signature 'AnnotatedImageList'
metadata(x)

## S4 method for signature 'ImageMetadata'
metadata(x)

## S4 method for signature 'ImageMetadataList'
metadata(x)

## S4 replacement method for signature 'AnnotatedImage'
metadata(x) <- value

coreMetadata(y) <- value

## S4 replacement method for signature 'AnnotatedImage'
coreMetadata(y) <- value

## S4 replacement method for signature 'ImageMetadata'
coreMetadata(y) <- value

globalMetadata(y) <- value

## S4 replacement method for signature 'AnnotatedImage'
globalMetadata(y) <- value

## S4 replacement method for signature 'ImageMetadata'
globalMetadata(y) <- value

seriesMetadata(y) <- value

## S4 replacement method for signature 'AnnotatedImage'
seriesMetadata(y) <- value

## S4 replacement method for signature 'ImageMetadata'
seriesMetadata(y) <- value

```

### Arguments

x	an <a href="#">AnnotatedImage</a> , <a href="#">AnnotatedImageList</a> , <a href="#">ImageMetadata</a> , or <a href="#">ImageMetadataList</a> object
series	series ID
...	arguments passed to <a href="#">grep</a>
value	depending on the context, an <a href="#">ImageMetadata</a> object or a list
y	an <a href="#">AnnotatedImage</a> or <a href="#">ImageMetadata</a> object

### Details

The ... arguments are passed to [grep](#) called on metadata names allowing for convenient subsetting.

**Value**

Named list consisting of key value pairs.

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**Examples**

```
img = read.image(system.file("images", "nuclei.tif", package="EBImage"))
coreMetadata(img)
# subset for specific names
globalMetadata(img, pattern="Image")
```

---

dimorder

*Image Frames Order*

---

**Description**

Get the ordering of image frames.

**Usage**

```
dimorder(x)
```

**Arguments**

x                    An [Image](#) object or an array

**Value**

A character vector giving the dimension names.

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**Examples**

```
# sample timelapse image
f = mockFile(sizeC=2, sizeT=10)
img = read.image(f)

dimorder(img)
```

---

FormatTools	<i>Bio-Formats FormatTools Class</i>
-------------	--------------------------------------

---

**Description**

A utility class for format reader and writer implementations.

**Usage**

```
FormatTools
```

**Format**

An object of class `jclassName` of length 1.

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**Examples**

```
# List available pixel types
sapply(0:7, FormatTools$getPixelTypeString)
```

---

ImageMetadata-class	<i>ImageMetadata and ImageMetadataList Class</i>
---------------------	--

---

**Description**

Formal representation of image metadata.

**Usage**

```
ImageMetadata(
  coreMetadata = NULL,
  globalMetadata = NULL,
  seriesMetadata = NULL
)

ImageMetadataList(...)

## S3 method for class 'ImageMetadata'
print(x, list.len = 5L, ...)

## S4 method for signature 'ImageMetadata'
show(object)

## S4 method for signature 'ImageMetadataList'
show(object)

## S3 method for class 'ImageMetadataList'
print(x, ...)
```

**Arguments**

coreMetadata	a list of core metadata entries
globalMetadata	a list of global metadata entries
seriesMetadata	a list of series metadata entries
...	further arguments to be passed to other methods
x	An ImageMetadata object
list.len	numeric; maximum number of metadata entries to display
object	an ImageMetadata object

**Value**

ImageMetadata returns a new [ImageMetadata](#) object.

ImageMetadataList returns a new [ImageMetadataList](#) object.

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**Examples**

```
f = mockFile(sizeX = 256, sizeY = 256)
img = read.image(f)
metadata(img)
```

---

mockFile

*Generate Test Images*


---

**Description**

Create a mock image of specific parameters for testing.

**Usage**

```
mockFile(name = "mockfile", ...)
```

**Arguments**

name	File name.
...	File parameters; available parameters are listed below.

**Details**

Generates mock files of specific size or pixel type containing gradient images. The desired parameters can be provided as key value pairs to the mockFile function. For a list of available parameters see below.

**Value**

A character string.



**Parameters**

<b>Name</b>	<b>Default</b>	<b>Description</b>
sizeX	512	horizontal size in pixels
sizeY	512	vertical size in pixels
sizeZ	1	number of Z sections
sizeC	1	number of channels
sizeT	1	number of time points
pixelType	uint8	string specifying pixel type: int8, uint8, int16, uint16, int32, uint32, float, double
bitsPerPixel	0	number of valid bits (<= number of bits implied by pixel type)
rgb	1	number of channels that are merged together
dimOrder	XYZCT	string describing dimension order
orderCertain	true	
little	true	whether or not the pixel data should be little-endian
interleaved	true	whether or not merged channels are interleaved
indexed	false	whether or not a color lookup table is present
falseColor	false	whether or not the color lookup table is just for making the image look pretty
series	1	number of series (Images)
lutLength	3	number of entries in the color lookup table

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**Examples**

```
f = mockFile(sizeX = 256, sizeY = 256)
img = read.image(f)
img
```

---

 RBioFormats

*R Interface to Bio-Formats*


---

**Description**

Interfaces the **Bio-Formats** Java library.

**Usage**

```
BioFormats.version()
```

**Value**

A character string containing the Bio-Formats library version.

**Functions**

- `BioFormats.version()`: Provides the version of the Bio-Formats library embedded in the package.

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**References**

[Bio-Formats website](#)

**Examples**

```
BioFormats.version()
```

---

read.image

*Read Images*

---

**Description**

Read image files using the Bio-Formats library. A list of supported formats can be found on the [Bio-Formats website](#).

**Usage**

```
read.image(
  file,
  filter.metadata = FALSE,
  proprietary.metadata = TRUE,
  normalize = TRUE,
  series,
  resolution,
  subset,
  read.metadata = TRUE
)
```

**Arguments**

file	character, file name
filter.metadata	logical, specifies whether ugly metadata (entries with unprintable characters, and extremely large entries) should be discarded from the metadata table
proprietary.metadata	logical, should proprietary metadata be populated to OME-XML
normalize	logical, should the original image data be mapped to the [0,1] range
series	integer vector specifying series to read; if missing all series included in the file are read
resolution	integer vector specifying resolution levels to read; if missing all levels read
subset	named list specifying image subsetting
read.metadata	logical, should image metadata be read

**Value**

An [AnnotatedImage](#) object or an [AnnotatedImageList](#) object in case of multi-series data.

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**See Also**

[read.metadata](#) for reading image metadata, [read.omexml](#) for reading image metadata as OME-XML

**Examples**

```
require(EBImage)
f = system.file("images", "sample-color.png", package="EBImage")

img = read.image(f)
img
```

---

read.metadata	<i>Read Image Metadata</i>
---------------	----------------------------

---

**Description**

Read image metadata using the Bio-Formats library. The list of supported file formats can be found on the [Bio-Formats website](#).

**Usage**

```
read.metadata(file, filter.metadata = FALSE, proprietary.metadata = TRUE)
```

**Arguments**

file	character, file name
filter.metadata	logical, specifies whether ugly metadata (entries with unprintable characters, and extremely large entries) should be discarded from the metadata table
proprietary.metadata	logical, should proprietary metadata be populated to OME-XML

**Value**

An [ImageMetadata](#) or [ImageMetadataList](#) object.

**Author(s)**

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

**See Also**

[read.omexml](#) for reading image metadata as OME-XML, [read.image](#) for reading image data

## Examples

```
require(EBImage)
f = system.file("images", "nuclei.tif", package="EBImage")

metadata = read.metadata(f)
str(metadata)
```

---

read.omexml

*Read OME-XML Metadata*

---

## Description

Read the OME-XML DOM tree.

## Usage

```
read.omexml(file, filter.metadata = FALSE, proprietary.metadata = TRUE)
```

## Arguments

file	character, file name
filter.metadata	logical, specifies whether ugly metadata (entries with unprintable characters, and extremely large entries) should be discarded from the metadata table
proprietary.metadata	logical, should proprietary metadata be populated to OME-XML

## Value

A string containing a dumped OME-XML DOM tree.

## Author(s)

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

## See Also

[read.metadata](#) for reading image metadata, [read.image](#) for reading image data

## Examples

```
require(EBImage)
f = system.file("images", "nuclei.tif", package="EBImage")

omexml = read.omexml(f)
omexml
```

---

seriesCount	<i>Number of Image Series</i>
-------------	-------------------------------

---

### Description

Get the number of image series contained in an object.

### Usage

```
seriesCount(x)

## S4 method for signature 'ANY'
seriesCount(x)

## S4 method for signature 'matrix'
seriesCount(x)

## S4 method for signature 'array'
seriesCount(x)

## S4 method for signature 'ImageMetadata'
seriesCount(x)

## S4 method for signature 'AnnotatedImageList'
seriesCount(x)

## S4 method for signature 'ImageMetadataList'
seriesCount(x)
```

### Arguments

x                    an images object.

### Details

Image series are encoded by [AnnotatedImageList](#) or [ImageMetadataList](#) objects. Therefore, only these objects can possibly yield image series count higher than 1 while for all the rest of image objects this number is expected to be 1.

### Value

The number of image series the object contains, see Details.

### Author(s)

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

## Examples

```
f = mockFile(sizeX = 256, sizeY = 256, series = 2)
img = read.image(f)
seriesCount(img)

meta <- metadata(img)
seriesCount(meta)
```

---

write.image

*Write Images*

---

## Description

Save image files using the Bio-Formats library. A list of supported formats can be found on the [Bio-Formats website](#).

## Usage

```
write.image(x, file, force = FALSE, pixelType, littleEndian)
```

## Arguments

x	an Image or <a href="#">AnnotatedImage</a> object
file	character, file name
force	logical(1), if TRUE overwrite existing file
pixelType	character(1), data type to store pixel values
littleEndian	boolean(1), pixel data endianness

## Value

File path to file is returned invisibly.

## Author(s)

Andrzej Oleś <<andrzej.oles@gmail.com>>, 2014-2023

## See Also

[read.image](#) for reading images.

## Examples

```
f = mockFile(sizeX = 256, sizeY = 256)
img = read.image(f)

tempfile = tempfile("", , ".png")
write.image(img, tempfile)
```

# Index

- \* **datasets**
  - FormatTools, 7
  - .AnnotatedImage (AnnotatedImage-class), 2
  - .AnnotatedImageList (AnnotatedImageList-class), 3
  - .ImageMetadata (ImageMetadata-class), 7
  - .ImageMetadataList (ImageMetadata-class), 7
- AnnotatedImage, 2, 3, 5, 10, 14
- AnnotatedImage (AnnotatedImage-class), 2
- AnnotatedImage-class, 2
- AnnotatedImageList, 3, 5, 10, 13
- AnnotatedImageList (AnnotatedImageList-class), 3
- AnnotatedImageList-class, 3
- as.Image.AnnotatedImage (AnnotatedImage-class), 2
- BioFormats.version (RBioFormats), 9
- checkJavaMemory, 4
- coreMetadata, 4
- coreMetadata<- (coreMetadata), 4
- coreMetadata<- , AnnotatedImage-method (coreMetadata), 4
- coreMetadata<- , ImageMetadata-method (coreMetadata), 4
- dimorder, 6
- FormatTools, 7
- globalMetadata (coreMetadata), 4
- globalMetadata<- (coreMetadata), 4
- globalMetadata<- , AnnotatedImage-method (coreMetadata), 4
- globalMetadata<- , ImageMetadata-method (coreMetadata), 4
- grep, 5
- Image, 2, 6
- ImageMetadata, 2, 3, 5, 8, 11
- ImageMetadata (ImageMetadata-class), 7
- ImageMetadata-class, 7
- ImageMetadataList, 5, 8, 11, 13
- ImageMetadataList (ImageMetadata-class), 7
- ImageMetadataList-class (ImageMetadata-class), 7
- metadata, AnnotatedImage-method (coreMetadata), 4
- metadata, AnnotatedImageList-method (coreMetadata), 4
- metadata, ANY-method (coreMetadata), 4
- metadata, ImageMetadata-method (coreMetadata), 4
- metadata, ImageMetadataList-method (coreMetadata), 4
- metadata<- , AnnotatedImage-method (coreMetadata), 4
- mockFile, 8
- print.AnnotatedImage (AnnotatedImage-class), 2
- print.ImageMetadata (ImageMetadata-class), 7
- print.ImageMetadataList (ImageMetadata-class), 7
- RBioFormats, 9
- read.image, 10, 11, 12, 14
- read.metadata, 11, 11, 12
- read.omexml, 11, 12
- seriesCount, 13
- seriesCount, AnnotatedImageList-method (seriesCount), 13
- seriesCount, ANY-method (seriesCount), 13
- seriesCount, array-method (seriesCount), 13
- seriesCount, ImageMetadata-method (seriesCount), 13
- seriesCount, ImageMetadataList-method (seriesCount), 13
- seriesCount, matrix-method (seriesCount), 13

`seriesMetadata` (`coreMetadata`), 4  
`seriesMetadata<-` (`coreMetadata`), 4  
`seriesMetadata<-`, `AnnotatedImage`-method  
    (`coreMetadata`), 4  
`seriesMetadata<-`, `ImageMetadata`-method  
    (`coreMetadata`), 4  
`show`, `AnnotatedImage`-method  
    (`AnnotatedImage`-class), 2  
`show`, `ImageMetadata`-method  
    (`ImageMetadata`-class), 7  
`show`, `ImageMetadataList`-method  
    (`ImageMetadata`-class), 7  
`write.image`, 14