| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/crypto/spec/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/imageio/event/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.imageio

The main package of the Java Image I/O API.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**IIOParamController**](http://docs.google.com/javax/imageio/IIOParamController.html) | An interface to be implemented by objects that can determine the settings of an IIOParam object, either by putting up a GUI to obtain values from a user, or by other means. |
| [**ImageTranscoder**](http://docs.google.com/javax/imageio/ImageTranscoder.html) | An interface providing metadata transcoding capability. |

| **Class Summary** | |
| --- | --- |
| [**IIOImage**](http://docs.google.com/javax/imageio/IIOImage.html) | A simple container class to aggregate an image, a set of thumbnail (preview) images, and an object representing metadata associated with the image. |
| [**IIOParam**](http://docs.google.com/javax/imageio/IIOParam.html) | A superclass of all classes describing how streams should be decoded or encoded. |
| [**ImageIO**](http://docs.google.com/javax/imageio/ImageIO.html) | A class containing static convenience methods for locating ImageReaders and ImageWriters, and performing simple encoding and decoding. |
| [**ImageReader**](http://docs.google.com/javax/imageio/ImageReader.html) | An abstract superclass for parsing and decoding of images. |
| [**ImageReadParam**](http://docs.google.com/javax/imageio/ImageReadParam.html) | A class describing how a stream is to be decoded. |
| [**ImageTypeSpecifier**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | A class that allows the format of an image (in particular, its SampleModel and ColorModel) to be specified in a convenient manner. |
| [**ImageWriteParam**](http://docs.google.com/javax/imageio/ImageWriteParam.html) | A class describing how a stream is to be encoded. |
| [**ImageWriter**](http://docs.google.com/javax/imageio/ImageWriter.html) | An abstract superclass for encoding and writing images. |

| **Exception Summary** | |
| --- | --- |
| [**IIOException**](http://docs.google.com/javax/imageio/IIOException.html) | An exception class used for signaling run-time failure of reading and writing operations. |

## Package javax.imageio Description

The main package of the Java Image I/O API.

Many common image I/O operations may be performed using the static methods of the ImageIO class.

This package contains the basic classes and interfaces for describing the contents of image files, including metadata and thumbnails (IIOImage); for controlling the image reading process (ImageReader, ImageReadParam, and ImageTypeSpecifier) and image writing process (ImageWriter and ImageWriteParam); for performing transcoding between formats (ImageTranscoder), and for reporting errors (IIOException).

All implementations of javax.imageio provide the following standard image format plug-ins:

|  | Reading | Writing | Notes | Metadata |
| --- | --- | --- | --- | --- |
| [JPEG](http://www.jpeg.org) | yes | yes | none | [JPEG metadata format](http://docs.google.com/metadata/doc-files/jpeg_metadata.html) |
| [PNG](http://www.libpng.org/pub/png/spec/) | yes | yes | none | [PNG metadata format](http://docs.google.com/metadata/doc-files/png_metadata.html) |
| BMP | yes | yes | none | [BMP metadata format](http://docs.google.com/metadata/doc-files/bmp_metadata.html) |
| [WBMP](http://www.wapforum.org/what/technical/SPEC-WAESpec-19990524.pdf) | yes | yes | none | [WBMP metadata format](http://docs.google.com/metadata/doc-files/wbmp_metadata.html) |
| [GIF](http://www.w3.org/Graphics/GIF/spec-gif89a.txt) | yes | yes | [GIF plug-in notes](#_2et92p0) | [GIF metadata format](http://docs.google.com/metadata/doc-files/gif_metadata.html) |

## Standard Plug-in Notes

### Standard plug-in for GIF image format

ImageIO provides ImageReader and ImageWriter plug-ins for the  [Graphics Interchange Format (GIF)](http://www.w3.org/Graphics/GIF/spec-gif89a.txt) image format. These are the "standard" GIF plug-ins, meaning those that are included in the JRE, as distinct from those included in standard extensions, or 3rd party plug-ins. The following notes and metadata specification apply to the standard plug-ins.

### Writing GIF images

The GIF image writer plug-in guarantees lossless writing for images which meet the following requirements:

* the number of bands is 1;
* the number of bits per sample is not greater than 8;
* the size of a color component is not greater than 8;

By default the GIF writer plug-in creates version "89a" images. This can be changed to "87a" by explicitly setting the version in the stream metadata (see  [GIF Stream Metadata Format Specification](http://docs.google.com/metadata/doc-files/gif_metadata.html#gif_stream_metadata_format)).

The GIF writer plug-in supports the creation of animated GIF images through the standard sequence writing methods defined in the ImageWriter class.

A global color table is written to the output stream if one of the following conditions is met:

* stream metadata containing a GlobalColorTable element is supplied;
* a sequence is being written and image metadata containing a LocalColorTable element is supplied for the first image in the sequence;
* image metadata is not supplied or does not contain a LocalColorTable element.

In the first case the global color table in the stream metadata is used, in the second the local color table in the image metadata is used, and in the third a global color table is created from the ColorModel or SampleModel of the (first) image.

A local color table is written to the output stream only if image metadata containing a LocalColorTable element is supplied to the writer, or no image metadata is supplied to the writer and the local color table which would be generated from the image itself is not equal to the global color table.

A Graphic Control Extension block is written to the output stream only if image metadata containing a GraphicControlExtension element is supplied to the writer, or no image metadata is supplied and the local color table generated from the image requires a transparent index. Application, Plain Text, and Comment Extension blocks are written only if they are supplied to the writer via image metadata.

The writing of interlaced images can be controlled by the progressive mode of the provided ImageWriteParam instance. If progressive mode is MODE\_DISABLED then a non-interlaced image will be written. If progressive mode is MODE\_DEFAULT then an interlaced image will be written. If progressive mode is MODE\_COPY\_FROM\_METADATA, then the metadata setting is used (if it is provided, otherwise an interlaced image will be written).

The GIF image writer plug-in supports setting output stream metadata from metadata supplied to the writer in either the native GIF stream metadata format  [javax\_imageio\_gif\_stream\_1.0](http://docs.google.com/metadata/doc-files/gif_metadata.html#gif_stream_metadata_format)  or the standard metadata format  [javax\_imageio\_1.0](http://docs.google.com/metadata/doc-files/standard_metadata.html), and setting output image metadata from metadata supplied to the writer in either the native GIF image metadata format  [javax\_imageio\_gif\_image\_1.0](http://docs.google.com/metadata/doc-files/gif_metadata.html#gif_image_metadata_format)  or the standard metadata format [javax\_imageio\_1.0](http://docs.google.com/metadata/doc-files/standard_metadata.html). The mapping of standard metadata format to the GIF native stream and image metadata formats is given in the tables  [here](http://docs.google.com/metadata/doc-files/gif_metadata.html#mapping) .

**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/crypto/spec/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/imageio/event/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).