| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ImageTranscoder.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/javax/imageio/ImageReadParam.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/ImageTranscoder.html)    [**NO FRAMES**](http://docs.google.com/ImageTranscoder.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **javax.imageio**

Interface ImageTranscoder

**All Known Implementing Classes:** [ImageWriter](http://docs.google.com/javax/imageio/ImageWriter.html)

public interface **ImageTranscoder**

An interface providing metadata transcoding capability.

Any image may be transcoded (written to a different format than the one it was originally stored in) simply by performing a read operation followed by a write operation. However, loss of data may occur in this process due to format differences.

In general, the best results will be achieved when format-specific metadata objects can be created to encapsulate as much information about the image and its associated metadata as possible, in terms that are understood by the specific ImageWriter used to perform the encoding.

An ImageTranscoder may be used to convert the IIOMetadata objects supplied by the ImageReader (representing per-stream and per-image metadata) into corresponding objects suitable for encoding by a particular ImageWriter. In the case where the methods of this interface are being called directly on an ImageWriter, the output will be suitable for that writer.

The internal details of converting an IIOMetadata object into a writer-specific format will vary according to the context of the operation. Typically, an ImageWriter will inspect the incoming object to see if it implements additional interfaces with which the writer is familiar. This might be the case, for example, if the object was obtained by means of a read operation on a reader plug-in written by the same vendor as the writer. In this case, the writer may access the incoming object by means of its plug-in specific interfaces. In this case, the re-encoding may be close to lossless if the image file format is kept constant. If the format is changing, the writer may still attempt to preserve as much information as possible.

If the incoming object does not implement any additional interfaces known to the writer, the writer has no choice but to access it via the standard IIOMetadata interfaces such as the tree view provided by IIOMetadata.getAsTree. In this case, there is likely to be significant loss of information.

An independent ImageTranscoder essentially takes on the same role as the writer plug-in in the above examples. It must be familiar with the private interfaces used by both the reader and writer plug-ins, and manually instantiate an object that will be usable by the writer. The resulting metadata objects may be used by the writer directly.

No independent implementations of ImageTranscoder are provided as part of the standard API. Instead, the intention of this interface is to provide a way for implementations to be created and discovered by applications as the need arises.

| **Method Summary** | |
| --- | --- |
| [IIOMetadata](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html) | [**convertImageMetadata**](http://docs.google.com/javax/imageio/ImageTranscoder.html#convertImageMetadata(javax.imageio.metadata.IIOMetadata,%20javax.imageio.ImageTypeSpecifier,%20javax.imageio.ImageWriteParam))([IIOMetadata](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html) inData, [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) imageType, [ImageWriteParam](http://docs.google.com/javax/imageio/ImageWriteParam.html) param)            Returns an IIOMetadata object that may be used for encoding and optionally modified using its document interfaces or other interfaces specific to the writer plug-in that will be used for encoding. |
| [IIOMetadata](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html) | [**convertStreamMetadata**](http://docs.google.com/javax/imageio/ImageTranscoder.html#convertStreamMetadata(javax.imageio.metadata.IIOMetadata,%20javax.imageio.ImageWriteParam))([IIOMetadata](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html) inData, [ImageWriteParam](http://docs.google.com/javax/imageio/ImageWriteParam.html) param)            Returns an IIOMetadata object that may be used for encoding and optionally modified using its document interfaces or other interfaces specific to the writer plug-in that will be used for encoding. |

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

### convertStreamMetadata

[IIOMetadata](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html) **convertStreamMetadata**([IIOMetadata](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html) inData,  
 [ImageWriteParam](http://docs.google.com/javax/imageio/ImageWriteParam.html) param)

Returns an IIOMetadata object that may be used for encoding and optionally modified using its document interfaces or other interfaces specific to the writer plug-in that will be used for encoding.

An optional ImageWriteParam may be supplied for cases where it may affect the structure of the stream metadata.

If the supplied ImageWriteParam contains optional setting values not understood by this writer or transcoder, they will be ignored.

**Parameters:**inData - an IIOMetadata object representing stream metadata, used to initialize the state of the returned object.param - an ImageWriteParam that will be used to encode the image, or null. **Returns:**an IIOMetadata object, or null if the plug-in does not provide metadata encoding capabilities. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if inData is null.

### convertImageMetadata

[IIOMetadata](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html) **convertImageMetadata**([IIOMetadata](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html) inData,  
 [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) imageType,  
 [ImageWriteParam](http://docs.google.com/javax/imageio/ImageWriteParam.html) param)

Returns an IIOMetadata object that may be used for encoding and optionally modified using its document interfaces or other interfaces specific to the writer plug-in that will be used for encoding.

An optional ImageWriteParam may be supplied for cases where it may affect the structure of the image metadata.

If the supplied ImageWriteParam contains optional setting values not understood by this writer or transcoder, they will be ignored.

**Parameters:**inData - an IIOMetadata object representing image metadata, used to initialize the state of the returned object.imageType - an ImageTypeSpecifier indicating the layout and color information of the image with which the metadata will be associated.param - an ImageWriteParam that will be used to encode the image, or null. **Returns:**an IIOMetadata object, or null if the plug-in does not provide metadata encoding capabilities. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if either of inData or imageType is null.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ImageTranscoder.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/javax/imageio/ImageReadParam.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/ImageTranscoder.html)    [**NO FRAMES**](http://docs.google.com/ImageTranscoder.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).