| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/IIOMetadata.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/metadata/IIOInvalidTreeException.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/metadata/IIOMetadata.html)    [**NO FRAMES**](http://docs.google.com/IIOMetadata.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#1ksv4uv) | [METHOD](#z337ya) |

## **javax.imageio.metadata**

Class IIOMetadata

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.imageio.metadata.IIOMetadata**

public abstract class **IIOMetadata**extends [Object](http://docs.google.com/java/lang/Object.html)

An abstract class to be extended by objects that represent metadata (non-image data) associated with images and streams. Plug-ins represent metadata using opaque, plug-in specific objects. These objects, however, provide the ability to access their internal information as a tree of IIOMetadataNode objects that support the XML DOM interfaces as well as additional interfaces for storing non-textual data and retrieving information about legal data values. The format of such trees is plug-in dependent, but plug-ins may choose to support a plug-in neutral format described below. A single plug-in may support multiple metadata formats, whose names maybe determined by calling getMetadataFormatNames. The plug-in may also support a single special format, referred to as the "native" format, which is designed to encode its metadata losslessly. This format will typically be designed specifically to work with a specific file format, so that images may be loaded and saved in the same format with no loss of metadata, but may be less useful for transfering metadata between an ImageReader and an ImageWriter for different image formats. To convert between two native formats as losslessly as the image file formats will allow, an ImageTranscoder object must be used.

**See Also:**[ImageReader.getImageMetadata(int)](http://docs.google.com/javax/imageio/ImageReader.html#getImageMetadata(int)), [ImageReader.getStreamMetadata()](http://docs.google.com/javax/imageio/ImageReader.html#getStreamMetadata()), [ImageReader.readAll(int, javax.imageio.ImageReadParam)](http://docs.google.com/javax/imageio/ImageReader.html#readAll(int,%20javax.imageio.ImageReadParam)), [ImageWriter.getDefaultStreamMetadata(javax.imageio.ImageWriteParam)](http://docs.google.com/javax/imageio/ImageWriter.html#getDefaultStreamMetadata(javax.imageio.ImageWriteParam)), [ImageWriter.getDefaultImageMetadata(javax.imageio.ImageTypeSpecifier, javax.imageio.ImageWriteParam)](http://docs.google.com/javax/imageio/ImageWriter.html#getDefaultImageMetadata(javax.imageio.ImageTypeSpecifier,%20javax.imageio.ImageWriteParam)), [ImageWriter.write(javax.imageio.metadata.IIOMetadata, javax.imageio.IIOImage, javax.imageio.ImageWriteParam)](http://docs.google.com/javax/imageio/ImageWriter.html#write(javax.imageio.metadata.IIOMetadata,%20javax.imageio.IIOImage,%20javax.imageio.ImageWriteParam)), [ImageWriter.convertImageMetadata(javax.imageio.metadata.IIOMetadata, javax.imageio.ImageTypeSpecifier, javax.imageio.ImageWriteParam)](http://docs.google.com/javax/imageio/ImageWriter.html#convertImageMetadata(javax.imageio.metadata.IIOMetadata,%20javax.imageio.ImageTypeSpecifier,%20javax.imageio.ImageWriteParam)), [ImageWriter.convertStreamMetadata(javax.imageio.metadata.IIOMetadata, javax.imageio.ImageWriteParam)](http://docs.google.com/javax/imageio/ImageWriter.html#convertStreamMetadata(javax.imageio.metadata.IIOMetadata,%20javax.imageio.ImageWriteParam)), [IIOImage](http://docs.google.com/javax/imageio/IIOImage.html), [ImageTranscoder](http://docs.google.com/javax/imageio/ImageTranscoder.html)

| **Field Summary** | |
| --- | --- |
| protected  [IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) | [**controller**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#controller)            The IIOMetadataController that will be used to provide settings for this IIOMetadata object when the activateController method is called. |
| protected  [IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) | [**defaultController**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#defaultController)            An IIOMetadataController that is suggested for use as the controller for this IIOMetadata object. |
| protected  [String](http://docs.google.com/java/lang/String.html)[] | [**extraMetadataFormatClassNames**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#extraMetadataFormatClassNames)            An array of names of classes implementing IIOMetadataFormat and representing the metadata formats, other than the standard and native formats, that are supported by this plug-in, initialized to null and set via the constructor. |
| protected  [String](http://docs.google.com/java/lang/String.html)[] | [**extraMetadataFormatNames**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#extraMetadataFormatNames)            An array of names of formats, other than the standard and native formats, that are supported by this plug-in, initialized to null and set via the constructor. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**nativeMetadataFormatClassName**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#nativeMetadataFormatClassName)            The name of the class implementing IIOMetadataFormat and representing the native metadata format, initialized to null and set via the constructor. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**nativeMetadataFormatName**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#nativeMetadataFormatName)            The name of the native metadata format for this object, initialized to null and set via the constructor. |
| protected  boolean | [**standardFormatSupported**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#standardFormatSupported)            A boolean indicating whether the concrete subclass supports the standard metadata format, set via the constructor. |

| **Constructor Summary** | |
| --- | --- |
| protected | [**IIOMetadata**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#IIOMetadata())()            Constructs an empty IIOMetadata object. |
| protected | [**IIOMetadata**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#IIOMetadata(boolean,%20java.lang.String,%20java.lang.String,%20java.lang.String%5B%5D,%20java.lang.String%5B%5D))(boolean standardMetadataFormatSupported, [String](http://docs.google.com/java/lang/String.html) nativeMetadataFormatName, [String](http://docs.google.com/java/lang/String.html) nativeMetadataFormatClassName, [String](http://docs.google.com/java/lang/String.html)[] extraMetadataFormatNames, [String](http://docs.google.com/java/lang/String.html)[] extraMetadataFormatClassNames)            Constructs an IIOMetadata object with the given format names and format class names, as well as a boolean indicating whether the standard format is supported. |

| **Method Summary** | |
| --- | --- |
| boolean | [**activateController**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#activateController())()            Activates the installed IIOMetadataController for this IIOMetadata object and returns the resulting value. |
| abstract  [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**getAsTree**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getAsTree(java.lang.String))([String](http://docs.google.com/java/lang/String.html) formatName)            Returns an XML DOM Node object that represents the root of a tree of metadata contained within this object according to the conventions defined by a given metadata format. |
| [IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) | [**getController**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getController())()            Returns whatever IIOMetadataController is currently installed. |
| [IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) | [**getDefaultController**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getDefaultController())()            Returns the default IIOMetadataController, if there is one, regardless of the currently installed controller. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getExtraMetadataFormatNames**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getExtraMetadataFormatNames())()            Returns an array of Strings containing the names of additional metadata formats, other than the native and standard formats, recognized by this plug-in's getAsTree, setFromTree, and mergeTree methods. |
| [IIOMetadataFormat](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html) | [**getMetadataFormat**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getMetadataFormat(java.lang.String))([String](http://docs.google.com/java/lang/String.html) formatName)            Returns an IIOMetadataFormat object describing the given metadata format, or null if no description is available. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getMetadataFormatNames**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getMetadataFormatNames())()            Returns an array of Strings containing the names of all metadata formats, including the native and standard formats, recognized by this plug-in's getAsTree, setFromTree, and mergeTree methods. |
| [String](http://docs.google.com/java/lang/String.html) | [**getNativeMetadataFormatName**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getNativeMetadataFormatName())()            Returns the name of the "native" metadata format for this plug-in, which typically allows for lossless encoding and transmission of the metadata stored in the format handled by this plug-in. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardChromaNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardChromaNode())()            Returns an IIOMetadataNode representing the chroma information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardCompressionNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardCompressionNode())()            Returns an IIOMetadataNode representing the compression information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardDataNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardDataNode())()            Returns an IIOMetadataNode representing the data format information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardDimensionNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardDimensionNode())()            Returns an IIOMetadataNode representing the dimension information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardDocumentNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardDocumentNode())()            Returns an IIOMetadataNode representing the document information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardTextNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTextNode())()            Returns an IIOMetadataNode representing the textual information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardTileNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTileNode())()            Returns an IIOMetadataNode representing the tiling information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardTransparencyNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTransparencyNode())()            Returns an IIOMetadataNode representing the transparency information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. |
| protected  [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) | [**getStandardTree**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTree())()            A utility method to return a tree of IIOMetadataNodes representing the metadata contained within this object according to the conventions of the standard javax\_imageio\_1.0 metadata format. |
| boolean | [**hasController**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#hasController())()            Returns true if there is a controller installed for this IIOMetadata object. |
| abstract  boolean | [**isReadOnly**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#isReadOnly())()            Returns true if this object does not support the mergeTree, setFromTree, and reset methods. |
| boolean | [**isStandardMetadataFormatSupported**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#isStandardMetadataFormatSupported())()            Returns true if the standard metadata format is supported by getMetadataFormat, getAsTree, setFromTree, and mergeTree. |
| abstract  void | [**mergeTree**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#mergeTree(java.lang.String,%20org.w3c.dom.Node))([String](http://docs.google.com/java/lang/String.html) formatName, [Node](http://docs.google.com/org/w3c/dom/Node.html) root)            Alters the internal state of this IIOMetadata object from a tree of XML DOM Nodes whose syntax is defined by the given metadata format. |
| abstract  void | [**reset**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#reset())()            Resets all the data stored in this object to default values, usually to the state this object was in immediately after construction, though the precise semantics are plug-in specific. |
| void | [**setController**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setController(javax.imageio.metadata.IIOMetadataController))([IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) controller)            Sets the IIOMetadataController to be used to provide settings for this IIOMetadata object when the activateController method is called, overriding any default controller. |
| void | [**setFromTree**](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setFromTree(java.lang.String,%20org.w3c.dom.Node))([String](http://docs.google.com/java/lang/String.html) formatName, [Node](http://docs.google.com/org/w3c/dom/Node.html) root)            Sets the internal state of this IIOMetadata object from a tree of XML DOM Nodes whose syntax is defined by the given metadata format. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

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

### standardFormatSupported

protected boolean **standardFormatSupported**

A boolean indicating whether the concrete subclass supports the standard metadata format, set via the constructor.

### nativeMetadataFormatName

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

The name of the native metadata format for this object, initialized to null and set via the constructor.

### nativeMetadataFormatClassName

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

The name of the class implementing IIOMetadataFormat and representing the native metadata format, initialized to null and set via the constructor.

### extraMetadataFormatNames

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

An array of names of formats, other than the standard and native formats, that are supported by this plug-in, initialized to null and set via the constructor.

### extraMetadataFormatClassNames

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

An array of names of classes implementing IIOMetadataFormat and representing the metadata formats, other than the standard and native formats, that are supported by this plug-in, initialized to null and set via the constructor.

### defaultController

protected [IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) **defaultController**

An IIOMetadataController that is suggested for use as the controller for this IIOMetadata object. It may be retrieved via getDefaultController. To install the default controller, call setController(getDefaultController()). This instance variable should be set by subclasses that choose to provide their own default controller, usually a GUI, for setting parameters.

**See Also:**[IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html), [getDefaultController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getDefaultController())

### controller

protected [IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) **controller**

The IIOMetadataController that will be used to provide settings for this IIOMetadata object when the activateController method is called. This value overrides any default controller, even when null.

**See Also:**[IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html), [setController(IIOMetadataController)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setController(javax.imageio.metadata.IIOMetadataController)), [hasController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#hasController()), [activateController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#activateController())

| **Constructor Detail** |
| --- |

### IIOMetadata

protected **IIOMetadata**()

Constructs an empty IIOMetadata object. The subclass is responsible for suppying values for all protected instance variables that will allow any non-overridden default implemtations of methods to satisfy their contracts. For example, extraMetadataFormatNames should not have length 0.

### IIOMetadata

protected **IIOMetadata**(boolean standardMetadataFormatSupported,  
 [String](http://docs.google.com/java/lang/String.html) nativeMetadataFormatName,  
 [String](http://docs.google.com/java/lang/String.html) nativeMetadataFormatClassName,  
 [String](http://docs.google.com/java/lang/String.html)[] extraMetadataFormatNames,  
 [String](http://docs.google.com/java/lang/String.html)[] extraMetadataFormatClassNames)

Constructs an IIOMetadata object with the given format names and format class names, as well as a boolean indicating whether the standard format is supported.

This constructor does not attempt to check the class names for validity. Invalid class names may cause exceptions in subsequent calls to getMetadataFormat.

**Parameters:**standardMetadataFormatSupported - true if this object can return or accept a DOM tree using the standard metadata format.nativeMetadataFormatName - the name of the native metadata format, as a String, or null if there is no native format.nativeMetadataFormatClassName - the name of the class of the native metadata format, or null if there is no native format.extraMetadataFormatNames - an array of Strings indicating additional formats supported by this object, or null if there are none.extraMetadataFormatClassNames - an array of Strings indicating the class names of any additional formats supported by this object, or null if there are none. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if extraMetadataFormatNames has length 0. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if extraMetadataFormatNames and extraMetadataFormatClassNames are neither both null, nor of the same length.

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

### isStandardMetadataFormatSupported

public boolean **isStandardMetadataFormatSupported**()

Returns true if the standard metadata format is supported by getMetadataFormat, getAsTree, setFromTree, and mergeTree.

The default implementation returns the value of the standardFormatSupported instance variable.

**Returns:**true if the standard metadata format is supported.**See Also:**[getAsTree(java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getAsTree(java.lang.String)), [setFromTree(java.lang.String, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setFromTree(java.lang.String,%20org.w3c.dom.Node)), [mergeTree(java.lang.String, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#mergeTree(java.lang.String,%20org.w3c.dom.Node)), [getMetadataFormat(java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getMetadataFormat(java.lang.String))

### isReadOnly

public abstract boolean **isReadOnly**()

Returns true if this object does not support the mergeTree, setFromTree, and reset methods.

**Returns:**true if this IIOMetadata object cannot be modified.

### getNativeMetadataFormatName

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

Returns the name of the "native" metadata format for this plug-in, which typically allows for lossless encoding and transmission of the metadata stored in the format handled by this plug-in. If no such format is supported, nullwill be returned.

The structure and contents of the "native" metadata format are defined by the plug-in that created this IIOMetadata object. Plug-ins for simple formats will usually create a dummy node for the root, and then a series of child nodes representing individual tags, chunks, or keyword/value pairs. A plug-in may choose whether or not to document its native format.

The default implementation returns the value of the nativeMetadataFormatName instance variable.

**Returns:**the name of the native format, or null.**See Also:**[getExtraMetadataFormatNames()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getExtraMetadataFormatNames()), [getMetadataFormatNames()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getMetadataFormatNames())

### getExtraMetadataFormatNames

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

Returns an array of Strings containing the names of additional metadata formats, other than the native and standard formats, recognized by this plug-in's getAsTree, setFromTree, and mergeTree methods. If there are no such additional formats, null is returned.

The default implementation returns a clone of the extraMetadataFormatNames instance variable.

**Returns:**an array of Strings with length at least 1, or null.**See Also:**[getAsTree(java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getAsTree(java.lang.String)), [setFromTree(java.lang.String, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setFromTree(java.lang.String,%20org.w3c.dom.Node)), [mergeTree(java.lang.String, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#mergeTree(java.lang.String,%20org.w3c.dom.Node)), [getNativeMetadataFormatName()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getNativeMetadataFormatName()), [getMetadataFormatNames()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getMetadataFormatNames())

### getMetadataFormatNames

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

Returns an array of Strings containing the names of all metadata formats, including the native and standard formats, recognized by this plug-in's getAsTree, setFromTree, and mergeTree methods. If there are no such formats, null is returned.

The default implementation calls getNativeMetadataFormatName, isStandardMetadataFormatSupported, and getExtraMetadataFormatNames and returns the combined results.

**Returns:**an array of Strings.**See Also:**[getNativeMetadataFormatName()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getNativeMetadataFormatName()), [isStandardMetadataFormatSupported()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#isStandardMetadataFormatSupported()), [getExtraMetadataFormatNames()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getExtraMetadataFormatNames())

### getMetadataFormat

public [IIOMetadataFormat](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html) **getMetadataFormat**([String](http://docs.google.com/java/lang/String.html) formatName)

Returns an IIOMetadataFormat object describing the given metadata format, or null if no description is available. The supplied name must be one of those returned by getMetadataFormatNames (*i.e.*, either the native format name, the standard format name, or one of those returned by getExtraMetadataFormatNames).

The default implementation checks the name against the global standard metadata format name, and returns that format if it is supported. Otherwise, it checks against the native format names followed by any additional format names. If a match is found, it retrieves the name of the IIOMetadataFormat class from nativeMetadataFormatClassName or extraMetadataFormatClassNames as appropriate, and constructs an instance of that class using its getInstance method.

**Parameters:**formatName - the desired metadata format. **Returns:**an IIOMetadataFormat object. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if formatName is null or is not one of the names recognized by the plug-in. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the class corresponding to the format name cannot be loaded.

### getAsTree

public abstract [Node](http://docs.google.com/org/w3c/dom/Node.html) **getAsTree**([String](http://docs.google.com/java/lang/String.html) formatName)

Returns an XML DOM Node object that represents the root of a tree of metadata contained within this object according to the conventions defined by a given metadata format.

The names of the available metadata formats may be queried using the getMetadataFormatNames method.

**Parameters:**formatName - the desired metadata format. **Returns:**an XML DOM Node object forming the root of a tree. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if formatName is null or is not one of the names returned by getMetadataFormatNames.**See Also:**[getMetadataFormatNames()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getMetadataFormatNames()), [setFromTree(java.lang.String, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setFromTree(java.lang.String,%20org.w3c.dom.Node)), [mergeTree(java.lang.String, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#mergeTree(java.lang.String,%20org.w3c.dom.Node))

### mergeTree

public abstract void **mergeTree**([String](http://docs.google.com/java/lang/String.html) formatName,  
 [Node](http://docs.google.com/org/w3c/dom/Node.html) root)  
 throws [IIOInvalidTreeException](http://docs.google.com/javax/imageio/metadata/IIOInvalidTreeException.html)

Alters the internal state of this IIOMetadata object from a tree of XML DOM Nodes whose syntax is defined by the given metadata format. The previous state is altered only as necessary to accomodate the nodes that are present in the given tree. If the tree structure or contents are invalid, an IIOInvalidTreeException will be thrown.

As the semantics of how a tree or subtree may be merged with another tree are completely format-specific, plug-in authors may implement this method in whatever manner is most appropriate for the format, including simply replacing all existing state with the contents of the given tree.

**Parameters:**formatName - the desired metadata format.root - an XML DOM Node object forming the root of a tree. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this object is read-only. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if formatName is null or is not one of the names returned by getMetadataFormatNames. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if root is null. [IIOInvalidTreeException](http://docs.google.com/javax/imageio/metadata/IIOInvalidTreeException.html) - if the tree cannot be parsed successfully using the rules of the given format.**See Also:**[getMetadataFormatNames()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getMetadataFormatNames()), [getAsTree(java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getAsTree(java.lang.String)), [setFromTree(java.lang.String, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setFromTree(java.lang.String,%20org.w3c.dom.Node))

### getStandardChromaNode

protected [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardChromaNode**()

Returns an IIOMetadataNode representing the chroma information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. This method is intended to be called by the utility routine getStandardTree.

The default implementation returns null.

Subclasses should override this method to produce an appropriate subtree if they wish to support the standard metadata format.

**Returns:**an IIOMetadataNode, or null.**See Also:**[getStandardTree()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTree())

### getStandardCompressionNode

protected [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardCompressionNode**()

Returns an IIOMetadataNode representing the compression information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. This method is intended to be called by the utility routine getStandardTree.

The default implementation returns null.

Subclasses should override this method to produce an appropriate subtree if they wish to support the standard metadata format.

**Returns:**an IIOMetadataNode, or null.**See Also:**[getStandardTree()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTree())

### getStandardDataNode

protected [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardDataNode**()

Returns an IIOMetadataNode representing the data format information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. This method is intended to be called by the utility routine getStandardTree.

The default implementation returns null.

Subclasses should override this method to produce an appropriate subtree if they wish to support the standard metadata format.

**Returns:**an IIOMetadataNode, or null.**See Also:**[getStandardTree()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTree())

### getStandardDimensionNode

protected [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardDimensionNode**()

Returns an IIOMetadataNode representing the dimension information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. This method is intended to be called by the utility routine getStandardTree.

The default implementation returns null.

Subclasses should override this method to produce an appropriate subtree if they wish to support the standard metadata format.

**Returns:**an IIOMetadataNode, or null.**See Also:**[getStandardTree()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTree())

### getStandardDocumentNode

protected [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardDocumentNode**()

Returns an IIOMetadataNode representing the document information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. This method is intended to be called by the utility routine getStandardTree.

The default implementation returns null.

Subclasses should override this method to produce an appropriate subtree if they wish to support the standard metadata format.

**Returns:**an IIOMetadataNode, or null.**See Also:**[getStandardTree()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTree())

### getStandardTextNode

protected [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardTextNode**()

Returns an IIOMetadataNode representing the textual information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. This method is intended to be called by the utility routine getStandardTree.

The default implementation returns null.

Subclasses should override this method to produce an appropriate subtree if they wish to support the standard metadata format.

**Returns:**an IIOMetadataNode, or null.**See Also:**[getStandardTree()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTree())

### getStandardTileNode

protected [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardTileNode**()

Returns an IIOMetadataNode representing the tiling information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. This method is intended to be called by the utility routine getStandardTree.

The default implementation returns null.

Subclasses should override this method to produce an appropriate subtree if they wish to support the standard metadata format.

**Returns:**an IIOMetadataNode, or null.**See Also:**[getStandardTree()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTree())

### getStandardTransparencyNode

protected [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardTransparencyNode**()

Returns an IIOMetadataNode representing the transparency information of the standard javax\_imageio\_1.0 metadata format, or null if no such information is available. This method is intended to be called by the utility routine getStandardTree.

The default implementation returns null.

Subclasses should override this method to produce an appropriate subtree if they wish to support the standard metadata format.

**Returns:**an IIOMetadataNode, or null.

### getStandardTree

protected final [IIOMetadataNode](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html) **getStandardTree**()

A utility method to return a tree of IIOMetadataNodes representing the metadata contained within this object according to the conventions of the standard javax\_imageio\_1.0 metadata format.

This method calls the various getStandard\*Node methods to supply each of the subtrees rooted at the children of the root node. If any of those methods returns null, the corresponding subtree will be omitted. If all of them return null, a tree consisting of a single root node will be returned.

**Returns:**an IIOMetadataNode representing the root of a metadata tree in the javax\_imageio\_1.0 format.**See Also:**[getStandardChromaNode()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardChromaNode()), [getStandardCompressionNode()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardCompressionNode()), [getStandardDataNode()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardDataNode()), [getStandardDimensionNode()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardDimensionNode()), [getStandardDocumentNode()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardDocumentNode()), [getStandardTextNode()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTextNode()), [getStandardTileNode()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTileNode()), [getStandardTransparencyNode()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getStandardTransparencyNode())

### setFromTree

public void **setFromTree**([String](http://docs.google.com/java/lang/String.html) formatName,  
 [Node](http://docs.google.com/org/w3c/dom/Node.html) root)  
 throws [IIOInvalidTreeException](http://docs.google.com/javax/imageio/metadata/IIOInvalidTreeException.html)

Sets the internal state of this IIOMetadata object from a tree of XML DOM Nodes whose syntax is defined by the given metadata format. The previous state is discarded. If the tree's structure or contents are invalid, an IIOInvalidTreeException will be thrown.

The default implementation calls reset followed by mergeTree(formatName, root).

**Parameters:**formatName - the desired metadata format.root - an XML DOM Node object forming the root of a tree. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this object is read-only. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if formatName is null or is not one of the names returned by getMetadataFormatNames. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if root is null. [IIOInvalidTreeException](http://docs.google.com/javax/imageio/metadata/IIOInvalidTreeException.html) - if the tree cannot be parsed successfully using the rules of the given format.**See Also:**[getMetadataFormatNames()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getMetadataFormatNames()), [getAsTree(java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getAsTree(java.lang.String)), [mergeTree(java.lang.String, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#mergeTree(java.lang.String,%20org.w3c.dom.Node))

### reset

public abstract void **reset**()

Resets all the data stored in this object to default values, usually to the state this object was in immediately after construction, though the precise semantics are plug-in specific. Note that there are many possible default values, depending on how the object was created.

**Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this object is read-only.**See Also:**[ImageReader.getStreamMetadata()](http://docs.google.com/javax/imageio/ImageReader.html#getStreamMetadata()), [ImageReader.getImageMetadata(int)](http://docs.google.com/javax/imageio/ImageReader.html#getImageMetadata(int)), [ImageWriter.getDefaultStreamMetadata(javax.imageio.ImageWriteParam)](http://docs.google.com/javax/imageio/ImageWriter.html#getDefaultStreamMetadata(javax.imageio.ImageWriteParam)), [ImageWriter.getDefaultImageMetadata(javax.imageio.ImageTypeSpecifier, javax.imageio.ImageWriteParam)](http://docs.google.com/javax/imageio/ImageWriter.html#getDefaultImageMetadata(javax.imageio.ImageTypeSpecifier,%20javax.imageio.ImageWriteParam))

### setController

public void **setController**([IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) controller)

Sets the IIOMetadataController to be used to provide settings for this IIOMetadata object when the activateController method is called, overriding any default controller. If the argument is null, no controller will be used, including any default. To restore the default, use setController(getDefaultController()).

The default implementation sets the controller instance variable to the supplied value.

**Parameters:**controller - An appropriate IIOMetadataController, or null.**See Also:**[IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html), [getController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getController()), [getDefaultController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getDefaultController()), [hasController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#hasController()), [activateController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#activateController())

### getController

public [IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) **getController**()

Returns whatever IIOMetadataController is currently installed. This could be the default if there is one, null, or the argument of the most recent call to setController.

The default implementation returns the value of the controller instance variable.

**Returns:**the currently installed IIOMetadataController, or null.**See Also:**[IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html), [setController(javax.imageio.metadata.IIOMetadataController)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setController(javax.imageio.metadata.IIOMetadataController)), [getDefaultController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getDefaultController()), [hasController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#hasController()), [activateController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#activateController())

### getDefaultController

public [IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) **getDefaultController**()

Returns the default IIOMetadataController, if there is one, regardless of the currently installed controller. If there is no default controller, returns null.

The default implementation returns the value of the defaultController instance variable.

**Returns:**the default IIOMetadataController, or null.**See Also:**[IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html), [setController(IIOMetadataController)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setController(javax.imageio.metadata.IIOMetadataController)), [getController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getController()), [hasController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#hasController()), [activateController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#activateController())

### hasController

public boolean **hasController**()

Returns true if there is a controller installed for this IIOMetadata object.

The default implementation returns true if the getController method returns a non-null value.

**Returns:**true if a controller is installed.**See Also:**[IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html), [setController(IIOMetadataController)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setController(javax.imageio.metadata.IIOMetadataController)), [getController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getController()), [getDefaultController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getDefaultController()), [activateController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#activateController())

### activateController

public boolean **activateController**()

Activates the installed IIOMetadataController for this IIOMetadata object and returns the resulting value. When this method returns true, all values for this IIOMetadata object will be ready for the next write operation. If false is returned, no settings in this object will have been disturbed (*i.e.*, the user canceled the operation).

Ordinarily, the controller will be a GUI providing a user interface for a subclass of IIOMetadata for a particular plug-in. Controllers need not be GUIs, however.

The default implementation calls getController and the calls activate on the returned object if hasController returns true.

**Returns:**true if the controller completed normally. **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if there is no controller currently installed.**See Also:**[IIOMetadataController](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html), [setController(IIOMetadataController)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#setController(javax.imageio.metadata.IIOMetadataController)), [getController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getController()), [getDefaultController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getDefaultController()), [hasController()](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#hasController())

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/IIOMetadata.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/metadata/IIOInvalidTreeException.html)   [**NEXT CLASS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataController.html) | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/metadata/IIOMetadata.html)    [**NO FRAMES**](http://docs.google.com/IIOMetadata.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#1ksv4uv) | [METHOD](#z337ya) |

[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).