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

## **javax.imageio.metadata**

Interface IIOMetadataFormat

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

public interface **IIOMetadataFormat**

An object describing the structure of metadata documents returned from IIOMetadata.getAsTree and passed to IIOMetadata.setFromTree and mergeTree. Document structures are described by a set of constraints on the type and number of child elements that may belong to a given parent element type, the names, types, and values of attributes that may belong to an element, and the type and values of Object reference that may be stored at a node.

N.B: classes that implement this interface should contain a method declared as public static getInstance() which returns an instance of the class. Commonly, an implentation will construct only a single instance and cache it for future invocations of getInstance.

The structures that may be described by this class are a subset of those expressible using XML document type definitions (DTDs), with the addition of some basic information on the datatypes of attributes and the ability to store an Object reference within a node. In the future, XML Schemas could be used to represent these structures, and many others.

The differences between IIOMetadataFormat-described structures and DTDs are as follows:

* Elements may not contain text or mix text with embedded tags.
* The children of an element must conform to one of a few simple patterns, described in the documentation for the CHILD\_\* constants;
* The in-memory representation of an elements may contain a reference to an Object. There is no provision for representing such objects textually.

| **Field Summary** | |
| --- | --- |
| static int | [**CHILD\_POLICY\_ALL**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#CHILD_POLICY_ALL)            A constant returned by getChildPolicy to indicate that an element must have a single instance of each of its legal child elements, in order. |
| static int | [**CHILD\_POLICY\_CHOICE**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#CHILD_POLICY_CHOICE)            A constant returned by getChildPolicy to indicate that an element must have zero or one children, selected from among its legal child elements. |
| static int | [**CHILD\_POLICY\_EMPTY**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#CHILD_POLICY_EMPTY)            A constant returned by getChildPolicy to indicate that an element may not have any children. |
| static int | [**CHILD\_POLICY\_MAX**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#CHILD_POLICY_MAX)            The largest valid CHILD\_POLICY\_\* constant, to be used for range checks. |
| static int | [**CHILD\_POLICY\_REPEAT**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#CHILD_POLICY_REPEAT)            A constant returned by getChildPolicy to indicate that an element must have zero or more instances of its unique legal child element. |
| static int | [**CHILD\_POLICY\_SEQUENCE**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#CHILD_POLICY_SEQUENCE)            A constant returned by getChildPolicy to indicate that an element must have a sequence of instances of any of its legal child elements. |
| static int | [**CHILD\_POLICY\_SOME**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#CHILD_POLICY_SOME)            A constant returned by getChildPolicy to indicate that an element must have zero or one instance of each of its legal child elements, in order. |
| static int | [**DATATYPE\_BOOLEAN**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#DATATYPE_BOOLEAN)            A constant returned by getAttributeDataType indicating that the value of an attribute is one of 'true' or 'false'. |
| static int | [**DATATYPE\_DOUBLE**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#DATATYPE_DOUBLE)            A constant returned by getAttributeDataType indicating that the value of an attribute is a string representation of a double-precision decimal floating-point number. |
| static int | [**DATATYPE\_FLOAT**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#DATATYPE_FLOAT)            A constant returned by getAttributeDataType indicating that the value of an attribute is a string representation of a decimal floating-point number. |
| static int | [**DATATYPE\_INTEGER**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#DATATYPE_INTEGER)            A constant returned by getAttributeDataType indicating that the value of an attribute is a string representation of an integer. |
| static int | [**DATATYPE\_STRING**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#DATATYPE_STRING)            A constant returned by getAttributeDataType indicating that the value of an attribute is a general Unicode string. |
| static int | [**VALUE\_ARBITRARY**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_ARBITRARY)            A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set a single, arbitrary value. |
| static int | [**VALUE\_ENUMERATION**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_ENUMERATION)            A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set one of a number of enumerated values. |
| static int | [**VALUE\_LIST**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_LIST)            A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set to a list or array of values. |
| static int | [**VALUE\_NONE**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_NONE)            A constant returned by getObjectValueType to indicate the absence of a user object. |
| static int | [**VALUE\_RANGE**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_RANGE)            A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set a range of values. |
| static int | [**VALUE\_RANGE\_MAX\_INCLUSIVE**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_RANGE_MAX_INCLUSIVE)            A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set to a range of values. |
| static int | [**VALUE\_RANGE\_MAX\_INCLUSIVE\_MASK**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_RANGE_MAX_INCLUSIVE_MASK)            A value that may be or'ed with VALUE\_RANGE to obtain VALUE\_RANGE\_MAX\_INCLUSIVE, and with VALUE\_RANGE\_MIN\_INCLUSIVE to obtain VALUE\_RANGE\_MIN\_MAX\_INCLUSIVE. |
| static int | [**VALUE\_RANGE\_MIN\_INCLUSIVE**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_RANGE_MIN_INCLUSIVE)            A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set to a range of values. |
| static int | [**VALUE\_RANGE\_MIN\_INCLUSIVE\_MASK**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_RANGE_MIN_INCLUSIVE_MASK)            A value that may be or'ed with VALUE\_RANGE to obtain VALUE\_RANGE\_MIN\_INCLUSIVE, and with VALUE\_RANGE\_MAX\_INCLUSIVE to obtain VALUE\_RANGE\_MIN\_MAX\_INCLUSIVE. |
| static int | [**VALUE\_RANGE\_MIN\_MAX\_INCLUSIVE**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_RANGE_MIN_MAX_INCLUSIVE)            A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set a range of values. |

| **Method Summary** | |
| --- | --- |
| boolean | [**canNodeAppear**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#canNodeAppear(java.lang.String,%20javax.imageio.ImageTypeSpecifier))([String](http://docs.google.com/java/lang/String.html) elementName, [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) imageType)            Returns true if the element (and the subtree below it) is allowed to appear in a metadata document for an image of the given type, defined by an ImageTypeSpecifier. |
| int | [**getAttributeDataType**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeDataType(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns one of the constants starting with DATATYPE\_, indicating the format and interpretation of the value of the given attribute within th enamed element. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAttributeDefaultValue**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeDefaultValue(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns the default value of the named attribute, if it is not explictly present within the named element, as a String, or null if no default value is available. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAttributeDescription**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeDescription(java.lang.String,%20java.lang.String,%20java.util.Locale))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName, [Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a String containing a description of the named attribute, or null. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getAttributeEnumerations**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeEnumerations(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns an array of Strings containing the legal enumerated values for the given attribute within the named element. |
| int | [**getAttributeListMaxLength**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeListMaxLength(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns the maximum number of list items that may be used to define this attribute. |
| int | [**getAttributeListMinLength**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeListMinLength(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns the minimum number of list items that may be used to define this attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAttributeMaxValue**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeMaxValue(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns the maximum legal value for the attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAttributeMinValue**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeMinValue(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns the minimum legal value for the attribute. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getAttributeNames**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeNames(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns an array of Strings listing the names of the attributes that may be associated with the named element. |
| int | [**getAttributeValueType**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getAttributeValueType(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns one of the constants starting with VALUE\_, indicating whether the values of the given attribute within the named element are arbitrary, constrained to lie within a specified range, constrained to be one of a set of enumerated values, or are a whitespace-separated list of arbitrary values. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getChildNames**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getChildNames(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns an array of Strings indicating the names of the element which are allowed to be children of the named element, in the order in which they should appear. |
| int | [**getChildPolicy**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getChildPolicy(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns one of the constants starting with CHILD\_POLICY\_, indicating the legal pattern of children for the named element. |
| [String](http://docs.google.com/java/lang/String.html) | [**getElementDescription**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getElementDescription(java.lang.String,%20java.util.Locale))([String](http://docs.google.com/java/lang/String.html) elementName, [Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a String containing a description of the named element, or null. |
| int | [**getElementMaxChildren**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getElementMaxChildren(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns the maximum number of children of the named element with child policy CHILD\_POLICY\_REPEAT. |
| int | [**getElementMinChildren**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getElementMinChildren(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns the minimum number of children of the named element with child policy CHILD\_POLICY\_REPEAT. |
| int | [**getObjectArrayMaxLength**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getObjectArrayMaxLength(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns the maximum number of array elements that may be used to define the Object reference within the named element. |
| int | [**getObjectArrayMinLength**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getObjectArrayMinLength(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns the minimum number of array elements that may be used to define the Object reference within the named element. |
| [Class](http://docs.google.com/java/lang/Class.html)<?> | [**getObjectClass**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getObjectClass(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns the Class type of the Object reference stored within the element. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getObjectDefaultValue**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getObjectDefaultValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns an Objects containing the default value for the Object reference within the named element. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getObjectEnumerations**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getObjectEnumerations(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns an array of Objects containing the legal enumerated values for the Object reference within the named element. |
| [Comparable](http://docs.google.com/java/lang/Comparable.html)<?> | [**getObjectMaxValue**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getObjectMaxValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns the maximum legal value for the Object reference within the named element. |
| [Comparable](http://docs.google.com/java/lang/Comparable.html)<?> | [**getObjectMinValue**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getObjectMinValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns the minimum legal value for the Object reference within the named element. |
| int | [**getObjectValueType**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getObjectValueType(java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName)            Returns one of the enumerated values starting with VALUE\_, indicating the type of values (enumeration, range, or array) that are allowed for the Object reference. |
| [String](http://docs.google.com/java/lang/String.html) | [**getRootName**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#getRootName())()            Returns the name of the root element of the format. |
| boolean | [**isAttributeRequired**](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#isAttributeRequired(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) elementName, [String](http://docs.google.com/java/lang/String.html) attrName)            Returns true if the named attribute must be present within the named element. |

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

### CHILD\_POLICY\_EMPTY

static final int **CHILD\_POLICY\_EMPTY**

A constant returned by getChildPolicy to indicate that an element may not have any children. In other words, it is required to be a leaf node.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.CHILD_POLICY_EMPTY)

### CHILD\_POLICY\_ALL

static final int **CHILD\_POLICY\_ALL**

A constant returned by getChildPolicy to indicate that an element must have a single instance of each of its legal child elements, in order. In DTD terms, the contents of the element are defined by a sequence a,b,c,d,....

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.CHILD_POLICY_ALL)

### CHILD\_POLICY\_SOME

static final int **CHILD\_POLICY\_SOME**

A constant returned by getChildPolicy to indicate that an element must have zero or one instance of each of its legal child elements, in order. In DTD terms, the contents of the element are defined by a sequence a?,b?,c?,d?,....

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.CHILD_POLICY_SOME)

### CHILD\_POLICY\_CHOICE

static final int **CHILD\_POLICY\_CHOICE**

A constant returned by getChildPolicy to indicate that an element must have zero or one children, selected from among its legal child elements. In DTD terms, the contents of the element are defined by a selection a|b|c|d|....

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.CHILD_POLICY_CHOICE)

### CHILD\_POLICY\_SEQUENCE

static final int **CHILD\_POLICY\_SEQUENCE**

A constant returned by getChildPolicy to indicate that an element must have a sequence of instances of any of its legal child elements. In DTD terms, the contents of the element are defined by a sequence (a|b|c|d|...)\*.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.CHILD_POLICY_SEQUENCE)

### CHILD\_POLICY\_REPEAT

static final int **CHILD\_POLICY\_REPEAT**

A constant returned by getChildPolicy to indicate that an element must have zero or more instances of its unique legal child element. In DTD terms, the contents of the element are defined by a starred expression a\*.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.CHILD_POLICY_REPEAT)

### CHILD\_POLICY\_MAX

static final int **CHILD\_POLICY\_MAX**

The largest valid CHILD\_POLICY\_\* constant, to be used for range checks.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.CHILD_POLICY_MAX)

### VALUE\_NONE

static final int **VALUE\_NONE**

A constant returned by getObjectValueType to indicate the absence of a user object.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_NONE)

### VALUE\_ARBITRARY

static final int **VALUE\_ARBITRARY**

A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set a single, arbitrary value.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_ARBITRARY)

### VALUE\_RANGE

static final int **VALUE\_RANGE**

A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set a range of values. Both the minimum and maximum values of the range are exclusive. It is recommended that ranges of integers be inclusive on both ends, and that exclusive ranges be used only for floating-point data.

**See Also:**[VALUE\_RANGE\_MIN\_MAX\_INCLUSIVE](http://docs.google.com/javax/imageio/metadata/IIOMetadataFormat.html#VALUE_RANGE_MIN_MAX_INCLUSIVE), [Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_RANGE)

### VALUE\_RANGE\_MIN\_INCLUSIVE\_MASK

static final int **VALUE\_RANGE\_MIN\_INCLUSIVE\_MASK**

A value that may be or'ed with VALUE\_RANGE to obtain VALUE\_RANGE\_MIN\_INCLUSIVE, and with VALUE\_RANGE\_MAX\_INCLUSIVE to obtain VALUE\_RANGE\_MIN\_MAX\_INCLUSIVE.

Similarly, the value may be and'ed with the value of getAttributeValueTypeor getObjectValueType to determine if the minimum value of the range is inclusive.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_RANGE_MIN_INCLUSIVE_MASK)

### VALUE\_RANGE\_MAX\_INCLUSIVE\_MASK

static final int **VALUE\_RANGE\_MAX\_INCLUSIVE\_MASK**

A value that may be or'ed with VALUE\_RANGE to obtain VALUE\_RANGE\_MAX\_INCLUSIVE, and with VALUE\_RANGE\_MIN\_INCLUSIVE to obtain VALUE\_RANGE\_MIN\_MAX\_INCLUSIVE.

Similarly, the value may be and'ed with the value of getAttributeValueTypeor getObjectValueType to determine if the maximum value of the range is inclusive.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_RANGE_MAX_INCLUSIVE_MASK)

### VALUE\_RANGE\_MIN\_INCLUSIVE

static final int **VALUE\_RANGE\_MIN\_INCLUSIVE**

A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set to a range of values. The minimum (but not the maximum) value of the range is inclusive.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_RANGE_MIN_INCLUSIVE)

### VALUE\_RANGE\_MAX\_INCLUSIVE

static final int **VALUE\_RANGE\_MAX\_INCLUSIVE**

A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set to a range of values. The maximum (but not the minimum) value of the range is inclusive.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_RANGE_MAX_INCLUSIVE)

### VALUE\_RANGE\_MIN\_MAX\_INCLUSIVE

static final int **VALUE\_RANGE\_MIN\_MAX\_INCLUSIVE**

A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set a range of values. Both the minimum and maximum values of the range are inclusive. It is recommended that ranges of integers be inclusive on both ends, and that exclusive ranges be used only for floating-point data.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_RANGE_MIN_MAX_INCLUSIVE)

### VALUE\_ENUMERATION

static final int **VALUE\_ENUMERATION**

A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set one of a number of enumerated values. In the case of attributes, these values are Strings; for objects, they are Objects implementing a given class or interface.

Attribute values of type DATATYPE\_BOOLEAN should be marked as enumerations.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_ENUMERATION)

### VALUE\_LIST

static final int **VALUE\_LIST**

A constant returned by getAttributeValueType and getObjectValueType to indicate that the attribute or user object may be set to a list or array of values. In the case of attributes, the list will consist of whitespace-separated values within a String; for objects, an array will be used.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.VALUE_LIST)

### DATATYPE\_STRING

static final int **DATATYPE\_STRING**

A constant returned by getAttributeDataType indicating that the value of an attribute is a general Unicode string.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.DATATYPE_STRING)

### DATATYPE\_BOOLEAN

static final int **DATATYPE\_BOOLEAN**

A constant returned by getAttributeDataType indicating that the value of an attribute is one of 'true' or 'false'.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.DATATYPE_BOOLEAN)

### DATATYPE\_INTEGER

static final int **DATATYPE\_INTEGER**

A constant returned by getAttributeDataType indicating that the value of an attribute is a string representation of an integer.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.DATATYPE_INTEGER)

### DATATYPE\_FLOAT

static final int **DATATYPE\_FLOAT**

A constant returned by getAttributeDataType indicating that the value of an attribute is a string representation of a decimal floating-point number.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.DATATYPE_FLOAT)

### DATATYPE\_DOUBLE

static final int **DATATYPE\_DOUBLE**

A constant returned by getAttributeDataType indicating that the value of an attribute is a string representation of a double-precision decimal floating-point number.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.imageio.metadata.IIOMetadataFormat.DATATYPE_DOUBLE)

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

### getRootName

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

Returns the name of the root element of the format.

**Returns:**a String.

### canNodeAppear

boolean **canNodeAppear**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [ImageTypeSpecifier](http://docs.google.com/javax/imageio/ImageTypeSpecifier.html) imageType)

Returns true if the element (and the subtree below it) is allowed to appear in a metadata document for an image of the given type, defined by an ImageTypeSpecifier. For example, a metadata document format might contain an element that describes the primary colors of the image, which would not be allowed when writing a grayscale image.

**Parameters:**elementName - the name of the element being queried.imageType - an ImageTypeSpecifier indicating the type of the image that will be associated with the metadata. **Returns:**true if the node is meaningful for images of the given type.

### getElementMinChildren

int **getElementMinChildren**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns the minimum number of children of the named element with child policy CHILD\_POLICY\_REPEAT. For example, an element representing color primary information might be required to have at least 3 children, one for each primay.

**Parameters:**elementName - the name of the element being queried. **Returns:**an int. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element does not have a child policy of CHILD\_POLICY\_REPEAT.

### getElementMaxChildren

int **getElementMaxChildren**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns the maximum number of children of the named element with child policy CHILD\_POLICY\_REPEAT. For example, an element representing an entry in an 8-bit color palette might be allowed to repeat up to 256 times. A value of Integer.MAX\_VALUE may be used to specify that there is no upper bound.

**Parameters:**elementName - the name of the element being queried. **Returns:**an int. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element does not have a child policy of CHILD\_POLICY\_REPEAT.

### getElementDescription

[String](http://docs.google.com/java/lang/String.html) **getElementDescription**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a String containing a description of the named element, or null. The desciption will be localized for the supplied Locale if possible.

If locale is null, the current default Locale returned by Locale.getLocale will be used.

**Parameters:**elementName - the name of the element.locale - the Locale for which localization will be attempted. **Returns:**the element description. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null, or is not a legal element name for this format.

### getChildPolicy

int **getChildPolicy**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns one of the constants starting with CHILD\_POLICY\_, indicating the legal pattern of children for the named element.

**Parameters:**elementName - the name of the element being queried. **Returns:**one of the CHILD\_POLICY\_\* constants. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format.

### getChildNames

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

Returns an array of Strings indicating the names of the element which are allowed to be children of the named element, in the order in which they should appear. If the element cannot have children, null is returned.

**Parameters:**elementName - the name of the element being queried. **Returns:**an array of Strings, or null. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format.

### getAttributeNames

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

Returns an array of Strings listing the names of the attributes that may be associated with the named element.

**Parameters:**elementName - the name of the element being queried. **Returns:**an array of Strings. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format.

### getAttributeValueType

int **getAttributeValueType**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns one of the constants starting with VALUE\_, indicating whether the values of the given attribute within the named element are arbitrary, constrained to lie within a specified range, constrained to be one of a set of enumerated values, or are a whitespace-separated list of arbitrary values.

**Parameters:**elementName - the name of the element being queried.attrName - the name of the attribute being queried. **Returns:**one of the VALUE\_\* constants. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element.

### getAttributeDataType

int **getAttributeDataType**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns one of the constants starting with DATATYPE\_, indicating the format and interpretation of the value of the given attribute within th enamed element. If getAttributeValueType returns VALUE\_LIST, then the legal value is a whitespace-spearated list of values of the returned datatype.

**Parameters:**elementName - the name of the element being queried.attrName - the name of the attribute being queried. **Returns:**one of the DATATYPE\_\* constants. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element.

### isAttributeRequired

boolean **isAttributeRequired**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns true if the named attribute must be present within the named element.

**Parameters:**elementName - the name of the element being queried.attrName - the name of the attribute being queried. **Returns:**true if the attribut must be present. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element.

### getAttributeDefaultValue

[String](http://docs.google.com/java/lang/String.html) **getAttributeDefaultValue**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns the default value of the named attribute, if it is not explictly present within the named element, as a String, or null if no default value is available.

**Parameters:**elementName - the name of the element being queried.attrName - the name of the attribute being queried. **Returns:**a String containing the default value, or null. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element.

### getAttributeEnumerations

[String](http://docs.google.com/java/lang/String.html)[] **getAttributeEnumerations**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns an array of Strings containing the legal enumerated values for the given attribute within the named element. This method should only be called if getAttributeValueType returns VALUE\_ENUMERATION.

**Parameters:**elementName - the name of the element being queried.attrName - the name of the attribute being queried. **Returns:**an array of Strings. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the given attribute is not defined as an enumeration.

### getAttributeMinValue

[String](http://docs.google.com/java/lang/String.html) **getAttributeMinValue**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns the minimum legal value for the attribute. Whether this value is inclusive or exclusive may be determined by the value of getAttributeValueType. The value is returned as a String; its interpretation is dependent on the value of getAttributeDataType. This method should only be called if getAttributeValueType returns VALUE\_RANGE\_\*.

**Parameters:**elementName - the name of the element being queried.attrName - the name of the attribute being queried. **Returns:**a String containing the smallest legal value for the attribute. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the given attribute is not defined as a range.

### getAttributeMaxValue

[String](http://docs.google.com/java/lang/String.html) **getAttributeMaxValue**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns the maximum legal value for the attribute. Whether this value is inclusive or exclusive may be determined by the value of getAttributeValueType. The value is returned as a String; its interpretation is dependent on the value of getAttributeDataType. This method should only be called if getAttributeValueType returns VALUE\_RANGE\_\*.

**Parameters:**elementName - the name of the element being queried, as a String.attrName - the name of the attribute being queried. **Returns:**a String containing the largest legal value for the attribute. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the given attribute is not defined as a range.

### getAttributeListMinLength

int **getAttributeListMinLength**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns the minimum number of list items that may be used to define this attribute. The attribute itself is defined as a String containing multiple whitespace-separated items. This method should only be called if getAttributeValueType returns VALUE\_LIST.

**Parameters:**elementName - the name of the element being queried.attrName - the name of the attribute being queried. **Returns:**the smallest legal number of list items for the attribute. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the given attribute is not defined as a list.

### getAttributeListMaxLength

int **getAttributeListMaxLength**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName)

Returns the maximum number of list items that may be used to define this attribute. A value of Integer.MAX\_VALUE may be used to specify that there is no upper bound. The attribute itself is defined as a String containing multiple whitespace-separated items. This method should only be called if getAttributeValueType returns VALUE\_LIST.

**Parameters:**elementName - the name of the element being queried.attrName - the name of the attribute being queried. **Returns:**the largest legal number of list items for the attribute. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the given attribute is not defined as a list.

### getAttributeDescription

[String](http://docs.google.com/java/lang/String.html) **getAttributeDescription**([String](http://docs.google.com/java/lang/String.html) elementName,  
 [String](http://docs.google.com/java/lang/String.html) attrName,  
 [Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a String containing a description of the named attribute, or null. The desciption will be localized for the supplied Locale if possible.

If locale is null, the current default Locale returned by Locale.getLocale will be used.

**Parameters:**elementName - the name of the element.attrName - the name of the attribute.locale - the Locale for which localization will be attempted. **Returns:**the attribute description. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null, or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if attrName is null or is not a legal attribute name for this element.

### getObjectValueType

int **getObjectValueType**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns one of the enumerated values starting with VALUE\_, indicating the type of values (enumeration, range, or array) that are allowed for the Object reference. If no object value can be stored within the given element, the result of this method will be VALUE\_NONE.

Object references whose legal values are defined as a range must implement the Comparable interface.

**Parameters:**elementName - the name of the element being queried. **Returns:**one of the VALUE\_\* constants. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format.**See Also:**[Comparable](http://docs.google.com/java/lang/Comparable.html)

### getObjectClass

[Class](http://docs.google.com/java/lang/Class.html)<?> **getObjectClass**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns the Class type of the Object reference stored within the element. If this element may not contain an Object reference, an IllegalArgumentException will be thrown. If the class type is an array, this field indicates the underlying class type (*e.g*, for an array of ints, this method would return int.class).

Object references whose legal values are defined as a range must implement the Comparable interface.

**Parameters:**elementName - the name of the element being queried. **Returns:**a Class object. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element cannot contain an object value (*i.e.*, if getObjectValueType(elementName) == VALUE\_NONE).

### getObjectDefaultValue

[Object](http://docs.google.com/java/lang/Object.html) **getObjectDefaultValue**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns an Objects containing the default value for the Object reference within the named element.

**Parameters:**elementName - the name of the element being queried. **Returns:**an Object. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element cannot contain an object value (*i.e.*, if getObjectValueType(elementName) == VALUE\_NONE).

### getObjectEnumerations

[Object](http://docs.google.com/java/lang/Object.html)[] **getObjectEnumerations**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns an array of Objects containing the legal enumerated values for the Object reference within the named element. This method should only be called if getObjectValueType returns VALUE\_ENUMERATION.

The Object associated with a node that accepts emuerated values must be equal to one of the values returned by this method, as defined by the == operator (as opposed to the Object.equals method).

**Parameters:**elementName - the name of the element being queried. **Returns:**an array of Objects. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element cannot contain an object value (*i.e.*, if getObjectValueType(elementName) == VALUE\_NONE). [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the Object is not defined as an enumeration.

### getObjectMinValue

[Comparable](http://docs.google.com/java/lang/Comparable.html)<?> **getObjectMinValue**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns the minimum legal value for the Object reference within the named element. Whether this value is inclusive or exclusive may be determined by the value of getObjectValueType. This method should only be called if getObjectValueType returns one of the constants starting with VALUE\_RANGE.

**Parameters:**elementName - the name of the element being queried. **Returns:**the smallest legal value for the attribute. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element cannot contain an object value (*i.e.*, if getObjectValueType(elementName) == VALUE\_NONE). [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the Object is not defined as a range.

### getObjectMaxValue

[Comparable](http://docs.google.com/java/lang/Comparable.html)<?> **getObjectMaxValue**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns the maximum legal value for the Object reference within the named element. Whether this value is inclusive or exclusive may be determined by the value of getObjectValueType. This method should only be called if getObjectValueType returns one of the constants starting with VALUE\_RANGE.

**Parameters:**elementName - the name of the element being queried. **Returns:**the smallest legal value for the attribute. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element cannot contain an object value (*i.e.*, if getObjectValueType(elementName) == VALUE\_NONE). [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the Object is not defined as a range.

### getObjectArrayMinLength

int **getObjectArrayMinLength**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns the minimum number of array elements that may be used to define the Object reference within the named element. This method should only be called if getObjectValueType returns VALUE\_LIST.

**Parameters:**elementName - the name of the element being queried. **Returns:**the smallest valid array length for the Object reference. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element cannot contain an object value (*i.e.*, if getObjectValueType(elementName) == VALUE\_NONE). [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the Object is not an array.

### getObjectArrayMaxLength

int **getObjectArrayMaxLength**([String](http://docs.google.com/java/lang/String.html) elementName)

Returns the maximum number of array elements that may be used to define the Object reference within the named element. A value of Integer.MAX\_VALUE may be used to specify that there is no upper bound. This method should only be called if getObjectValueType returns VALUE\_LIST.

**Parameters:**elementName - the name of the element being queried. **Returns:**the largest valid array length for the Object reference. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if elementName is null or is not a legal element name for this format. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the named element cannot contain an object value (*i.e.*, if getObjectValueType(elementName) == VALUE\_NONE). [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the Object is not an array.

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

[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).