| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/IIOMetadataNode.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/IIOMetadataFormatImpl.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/metadata/IIOMetadataNode.html)    [**NO FRAMES**](http://docs.google.com/IIOMetadataNode.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#3rdcrjn) |

## **javax.imageio.metadata**

Class IIOMetadataNode

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

**All Implemented Interfaces:** [Element](http://docs.google.com/org/w3c/dom/Element.html), [Node](http://docs.google.com/org/w3c/dom/Node.html), [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html)

public class **IIOMetadataNode**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Element](http://docs.google.com/org/w3c/dom/Element.html), [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html)

A class representing a node in a meta-data tree, which implements the  [org.w3c.dom.Element](http://docs.google.com/api/org/w3c/dom/Element.html) interface and additionally allows for the storage of non-textual objects via the getUserObject and setUserObject methods.

This class is not intended to be used for general XML processing. In particular, Element nodes created within the Image I/O API are not compatible with those created by Sun's standard implementation of the org.w3.dom API. In particular, the implementation is tuned for simple uses and may not perform well for intensive processing.

Namespaces are ignored in this implementation. The terms "tag name" and "node name" are always considered to be synonymous.

**See Also:**[IIOMetadata.getAsTree(java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadata.html#getAsTree(java.lang.String)), [IIOMetadata.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)), [IIOMetadata.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))

| **Field Summary** | |
| --- | --- |

| **Fields inherited from interface org.w3c.dom.**[**Node**](http://docs.google.com/org/w3c/dom/Node.html) |
| --- |
| [ATTRIBUTE\_NODE](http://docs.google.com/org/w3c/dom/Node.html#ATTRIBUTE_NODE), [CDATA\_SECTION\_NODE](http://docs.google.com/org/w3c/dom/Node.html#CDATA_SECTION_NODE), [COMMENT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#COMMENT_NODE), [DOCUMENT\_FRAGMENT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_FRAGMENT_NODE), [DOCUMENT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_NODE), [DOCUMENT\_POSITION\_CONTAINED\_BY](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_CONTAINED_BY), [DOCUMENT\_POSITION\_CONTAINS](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_CONTAINS), [DOCUMENT\_POSITION\_DISCONNECTED](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_DISCONNECTED), [DOCUMENT\_POSITION\_FOLLOWING](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_FOLLOWING), [DOCUMENT\_POSITION\_IMPLEMENTATION\_SPECIFIC](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC), [DOCUMENT\_POSITION\_PRECEDING](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_POSITION_PRECEDING), [DOCUMENT\_TYPE\_NODE](http://docs.google.com/org/w3c/dom/Node.html#DOCUMENT_TYPE_NODE), [ELEMENT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#ELEMENT_NODE), [ENTITY\_NODE](http://docs.google.com/org/w3c/dom/Node.html#ENTITY_NODE), [ENTITY\_REFERENCE\_NODE](http://docs.google.com/org/w3c/dom/Node.html#ENTITY_REFERENCE_NODE), [NOTATION\_NODE](http://docs.google.com/org/w3c/dom/Node.html#NOTATION_NODE), [PROCESSING\_INSTRUCTION\_NODE](http://docs.google.com/org/w3c/dom/Node.html#PROCESSING_INSTRUCTION_NODE), [TEXT\_NODE](http://docs.google.com/org/w3c/dom/Node.html#TEXT_NODE) |

| **Constructor Summary** | |
| --- | --- |
| [**IIOMetadataNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#IIOMetadataNode())()            Constructs an empty IIOMetadataNode. |
| [**IIOMetadataNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#IIOMetadataNode(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nodeName)            Constructs an IIOMetadataNode with a given node name. |

| **Method Summary** | |
| --- | --- |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**appendChild**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#appendChild(org.w3c.dom.Node))([Node](http://docs.google.com/org/w3c/dom/Node.html) newChild)            Adds the node newChild to the end of the list of children of this node. |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**cloneNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#cloneNode(boolean))(boolean deep)            Returns a duplicate of this node. |
| short | [**compareDocumentPosition**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#compareDocumentPosition(org.w3c.dom.Node))([Node](http://docs.google.com/org/w3c/dom/Node.html) other)            Compares the reference node, i.e. |
| [String](http://docs.google.com/java/lang/String.html) | [**getAttribute**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getAttribute(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Retrieves an attribute value by name. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**getAttributeNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getAttributeNode(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Retrieves an attribute node by name. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**getAttributeNodeNS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getAttributeNodeNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Equivalent to getAttributeNode(localName). |
| [String](http://docs.google.com/java/lang/String.html) | [**getAttributeNS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getAttributeNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Equivalent to getAttribute(localName). |
| [NamedNodeMap](http://docs.google.com/org/w3c/dom/NamedNodeMap.html) | [**getAttributes**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getAttributes())()            A NamedNodeMap containing the attributes of this node (if it is an Element) or null otherwise. |
| [String](http://docs.google.com/java/lang/String.html) | [**getBaseURI**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getBaseURI())()            The absolute base URI of this node or null if the implementation wasn't able to obtain an absolute URI. |
| [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) | [**getChildNodes**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getChildNodes())()            A NodeList that contains all children of this node. |
| [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) | [**getElementsByTagName**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getElementsByTagName(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns a NodeList of all descendant Elements with a given tag name, in document order. |
| [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) | [**getElementsByTagNameNS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getElementsByTagNameNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Equivalent to getElementsByTagName(localName). |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getFeature**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getFeature(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) feature, [String](http://docs.google.com/java/lang/String.html) version)            This method returns a specialized object which implements the specialized APIs of the specified feature and version, as specified in . |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**getFirstChild**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getFirstChild())()            Returns the first child of this node, or null if the node has no children. |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**getLastChild**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getLastChild())()            Returns the last child of this node, or null if the node has no children. |
| int | [**getLength**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getLength())()            The number of nodes in the list. |
| [String](http://docs.google.com/java/lang/String.html) | [**getLocalName**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getLocalName())()            Equivalent to getNodeName. |
| [String](http://docs.google.com/java/lang/String.html) | [**getNamespaceURI**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getNamespaceURI())()            Returns null, since namespaces are not supported. |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**getNextSibling**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getNextSibling())()            Returns the next sibling of this node, or null if the node has no next sibling. |
| [String](http://docs.google.com/java/lang/String.html) | [**getNodeName**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getNodeName())()            Returns the node name associated with this node. |
| short | [**getNodeType**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getNodeType())()            Returns the node type, which is always ELEMENT\_NODE. |
| [String](http://docs.google.com/java/lang/String.html) | [**getNodeValue**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getNodeValue())()            The value of this node, depending on its type; see the table above. |
| [Document](http://docs.google.com/org/w3c/dom/Document.html) | [**getOwnerDocument**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getOwnerDocument())()            Returns null, since IIOMetadataNodes do not belong to any Document. |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**getParentNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getParentNode())()            Returns the parent of this node. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPrefix**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getPrefix())()            Returns null, since namespaces are not supported. |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**getPreviousSibling**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getPreviousSibling())()            Returns the previous sibling of this node, or null if this node has no previous sibling. |
| [TypeInfo](http://docs.google.com/org/w3c/dom/TypeInfo.html) | [**getSchemaTypeInfo**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getSchemaTypeInfo())()            The type information associated with this element. |
| [String](http://docs.google.com/java/lang/String.html) | [**getTagName**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getTagName())()            The name of the element. |
| [String](http://docs.google.com/java/lang/String.html) | [**getTextContent**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getTextContent())()            This attribute returns the text content of this node and its descendants. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getUserData**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getUserData(java.lang.String))([String](http://docs.google.com/java/lang/String.html) key)            Retrieves the object associated to a key on a this node. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getUserObject**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getUserObject())()            Returns the Object value associated with this node. |
| boolean | [**hasAttribute**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#hasAttribute(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns true when an attribute with a given name is specified on this element or has a default value, false otherwise. |
| boolean | [**hasAttributeNS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#hasAttributeNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Equivalent to hasAttribute(localName). |
| boolean | [**hasAttributes**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#hasAttributes())()            Returns whether this node (if it is an element) has any attributes. |
| boolean | [**hasChildNodes**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#hasChildNodes())()            Returns true if this node has child nodes. |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**insertBefore**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#insertBefore(org.w3c.dom.Node,%20org.w3c.dom.Node))([Node](http://docs.google.com/org/w3c/dom/Node.html) newChild, [Node](http://docs.google.com/org/w3c/dom/Node.html) refChild)            Inserts the node newChild before the existing child node refChild. |
| boolean | [**isDefaultNamespace**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#isDefaultNamespace(java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI)            This method checks if the specified namespaceURI is the default namespace or not. |
| boolean | [**isEqualNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#isEqualNode(org.w3c.dom.Node))([Node](http://docs.google.com/org/w3c/dom/Node.html) node)            Tests whether two nodes are equal. |
| boolean | [**isSameNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#isSameNode(org.w3c.dom.Node))([Node](http://docs.google.com/org/w3c/dom/Node.html) node)            Returns whether this node is the same node as the given one. |
| boolean | [**isSupported**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#isSupported(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) feature, [String](http://docs.google.com/java/lang/String.html) version)            Returns false since DOM features are not supported. |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**item**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#item(int))(int index)            Returns the indexth item in the collection. |
| [String](http://docs.google.com/java/lang/String.html) | [**lookupNamespaceURI**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#lookupNamespaceURI(java.lang.String))([String](http://docs.google.com/java/lang/String.html) prefix)            Look up the namespace URI associated to the given prefix, starting from this node. |
| [String](http://docs.google.com/java/lang/String.html) | [**lookupPrefix**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#lookupPrefix(java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI)            Look up the prefix associated to the given namespace URI, starting from this node. |
| void | [**normalize**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#normalize())()            Does nothing, since IIOMetadataNodes do not contain Text children. |
| void | [**removeAttribute**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#removeAttribute(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Removes an attribute by name. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**removeAttributeNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#removeAttributeNode(org.w3c.dom.Attr))([Attr](http://docs.google.com/org/w3c/dom/Attr.html) oldAttr)            Removes the specified attribute node. |
| void | [**removeAttributeNS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#removeAttributeNS(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName)            Equivalent to removeAttribute(localName). |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**removeChild**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#removeChild(org.w3c.dom.Node))([Node](http://docs.google.com/org/w3c/dom/Node.html) oldChild)            Removes the child node indicated by oldChild from the list of children, and returns it. |
| [Node](http://docs.google.com/org/w3c/dom/Node.html) | [**replaceChild**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#replaceChild(org.w3c.dom.Node,%20org.w3c.dom.Node))([Node](http://docs.google.com/org/w3c/dom/Node.html) newChild, [Node](http://docs.google.com/org/w3c/dom/Node.html) oldChild)            Replaces the child node oldChild with newChild in the list of children, and returns the oldChild node. |
| void | [**setAttribute**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setAttribute(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) name, [String](http://docs.google.com/java/lang/String.html) value)            Adds a new attribute. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**setAttributeNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setAttributeNode(org.w3c.dom.Attr))([Attr](http://docs.google.com/org/w3c/dom/Attr.html) newAttr)            Adds a new attribute node. |
| [Attr](http://docs.google.com/org/w3c/dom/Attr.html) | [**setAttributeNodeNS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setAttributeNodeNS(org.w3c.dom.Attr))([Attr](http://docs.google.com/org/w3c/dom/Attr.html) newAttr)            Equivalent to setAttributeNode(newAttr). |
| void | [**setAttributeNS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setAttributeNS(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) qualifiedName, [String](http://docs.google.com/java/lang/String.html) value)            Equivalent to setAttribute(qualifiedName, value). |
| void | [**setIdAttribute**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setIdAttribute(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) name, boolean isId)            If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . |
| void | [**setIdAttributeNode**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setIdAttributeNode(org.w3c.dom.Attr,%20boolean))([Attr](http://docs.google.com/org/w3c/dom/Attr.html) idAttr, boolean isId)            If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . |
| void | [**setIdAttributeNS**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setIdAttributeNS(java.lang.String,%20java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) namespaceURI, [String](http://docs.google.com/java/lang/String.html) localName, boolean isId)            If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . |
| void | [**setNodeValue**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setNodeValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) nodeValue)            The value of this node, depending on its type; see the table above. |
| void | [**setPrefix**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setPrefix(java.lang.String))([String](http://docs.google.com/java/lang/String.html) prefix)            Does nothing, since namespaces are not supported. |
| void | [**setTextContent**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setTextContent(java.lang.String))([String](http://docs.google.com/java/lang/String.html) textContent)            This attribute returns the text content of this node and its descendants. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**setUserData**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setUserData(java.lang.String,%20java.lang.Object,%20org.w3c.dom.UserDataHandler))([String](http://docs.google.com/java/lang/String.html) key, [Object](http://docs.google.com/java/lang/Object.html) data, [UserDataHandler](http://docs.google.com/org/w3c/dom/UserDataHandler.html) handler)            Associate an object to a key on this node. |
| void | [**setUserObject**](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setUserObject(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) userObject)            Sets the value associated with this node. |

| **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)) |

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

### IIOMetadataNode

public **IIOMetadataNode**()

Constructs an empty IIOMetadataNode.

### IIOMetadataNode

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

Constructs an IIOMetadataNode with a given node name.

**Parameters:**nodeName - the name of the node, as a String.

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

### getNodeName

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

Returns the node name associated with this node.

**Specified by:**[getNodeName](http://docs.google.com/org/w3c/dom/Node.html#getNodeName()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**the node name, as a String.

### getNodeValue

public [String](http://docs.google.com/java/lang/String.html) **getNodeValue**()  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#getNodeValue()) The value of this node, depending on its type; see the table above. When it is defined to be null, setting it has no effect, including if the node is read-only.

**Specified by:**[getNodeValue](http://docs.google.com/org/w3c/dom/Node.html#getNodeValue()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - DOMSTRING\_SIZE\_ERR: Raised when it would return more characters than fit in a DOMString variable on the implementation platform.

### setNodeValue

public void **setNodeValue**([String](http://docs.google.com/java/lang/String.html) nodeValue)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#setNodeValue(java.lang.String)) The value of this node, depending on its type; see the table above. When it is defined to be null, setting it has no effect, including if the node is read-only.

**Specified by:**[setNodeValue](http://docs.google.com/org/w3c/dom/Node.html#setNodeValue(java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised when the node is readonly and if it is not defined to be null.

### getNodeType

public short **getNodeType**()

Returns the node type, which is always ELEMENT\_NODE.

**Specified by:**[getNodeType](http://docs.google.com/org/w3c/dom/Node.html#getNodeType()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**the short value ELEMENT\_NODE.

### getParentNode

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **getParentNode**()

Returns the parent of this node. A null value indicates that the node is the root of its own tree. To add a node to an existing tree, use one of the insertBefore, replaceChild, or appendChild methods.

**Specified by:**[getParentNode](http://docs.google.com/org/w3c/dom/Node.html#getParentNode()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**the parent, as a Node.**See Also:**[insertBefore(org.w3c.dom.Node, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#insertBefore(org.w3c.dom.Node,%20org.w3c.dom.Node)), [replaceChild(org.w3c.dom.Node, org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#replaceChild(org.w3c.dom.Node,%20org.w3c.dom.Node)), [appendChild(org.w3c.dom.Node)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#appendChild(org.w3c.dom.Node))

### getChildNodes

public [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) **getChildNodes**()

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#getChildNodes()) A NodeList that contains all children of this node. If there are no children, this is a NodeList containing no nodes.

**Specified by:**[getChildNodes](http://docs.google.com/org/w3c/dom/Node.html#getChildNodes()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html)

### getFirstChild

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **getFirstChild**()

Returns the first child of this node, or null if the node has no children.

**Specified by:**[getFirstChild](http://docs.google.com/org/w3c/dom/Node.html#getFirstChild()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**the first child, as a Node, or null

### getLastChild

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **getLastChild**()

Returns the last child of this node, or null if the node has no children.

**Specified by:**[getLastChild](http://docs.google.com/org/w3c/dom/Node.html#getLastChild()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**the last child, as a Node, or null.

### getPreviousSibling

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **getPreviousSibling**()

Returns the previous sibling of this node, or null if this node has no previous sibling.

**Specified by:**[getPreviousSibling](http://docs.google.com/org/w3c/dom/Node.html#getPreviousSibling()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**the previous sibling, as a Node, or null.

### getNextSibling

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **getNextSibling**()

Returns the next sibling of this node, or null if the node has no next sibling.

**Specified by:**[getNextSibling](http://docs.google.com/org/w3c/dom/Node.html#getNextSibling()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**the next sibling, as a Node, or null.

### getAttributes

public [NamedNodeMap](http://docs.google.com/org/w3c/dom/NamedNodeMap.html) **getAttributes**()

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#getAttributes()) A NamedNodeMap containing the attributes of this node (if it is an Element) or null otherwise.

**Specified by:**[getAttributes](http://docs.google.com/org/w3c/dom/Node.html#getAttributes()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html)

### getOwnerDocument

public [Document](http://docs.google.com/org/w3c/dom/Document.html) **getOwnerDocument**()

Returns null, since IIOMetadataNodes do not belong to any Document.

**Specified by:**[getOwnerDocument](http://docs.google.com/org/w3c/dom/Node.html#getOwnerDocument()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**null.

### insertBefore

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **insertBefore**([Node](http://docs.google.com/org/w3c/dom/Node.html) newChild,  
 [Node](http://docs.google.com/org/w3c/dom/Node.html) refChild)

Inserts the node newChild before the existing child node refChild. If refChild is null, insert newChild at the end of the list of children.

**Specified by:**[insertBefore](http://docs.google.com/org/w3c/dom/Node.html#insertBefore(org.w3c.dom.Node,%20org.w3c.dom.Node)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**newChild - the Node to insert.refChild - the reference Node. **Returns:**the node being inserted. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if newChild is null.

### replaceChild

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **replaceChild**([Node](http://docs.google.com/org/w3c/dom/Node.html) newChild,  
 [Node](http://docs.google.com/org/w3c/dom/Node.html) oldChild)

Replaces the child node oldChild with newChild in the list of children, and returns the oldChild node.

**Specified by:**[replaceChild](http://docs.google.com/org/w3c/dom/Node.html#replaceChild(org.w3c.dom.Node,%20org.w3c.dom.Node)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**newChild - the Node to insert.oldChild - the Node to be replaced. **Returns:**the node replaced. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if newChild is null.

### removeChild

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **removeChild**([Node](http://docs.google.com/org/w3c/dom/Node.html) oldChild)

Removes the child node indicated by oldChild from the list of children, and returns it.

**Specified by:**[removeChild](http://docs.google.com/org/w3c/dom/Node.html#removeChild(org.w3c.dom.Node)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**oldChild - the Node to be removed. **Returns:**the node removed. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if oldChild is null.

### appendChild

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **appendChild**([Node](http://docs.google.com/org/w3c/dom/Node.html) newChild)

Adds the node newChild to the end of the list of children of this node.

**Specified by:**[appendChild](http://docs.google.com/org/w3c/dom/Node.html#appendChild(org.w3c.dom.Node)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**newChild - the Node to insert. **Returns:**the node added. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if newChild is null.

### hasChildNodes

public boolean **hasChildNodes**()

Returns true if this node has child nodes.

**Specified by:**[hasChildNodes](http://docs.google.com/org/w3c/dom/Node.html#hasChildNodes()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**true if this node has children.

### cloneNode

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **cloneNode**(boolean deep)

Returns a duplicate of this node. The duplicate node has no parent (getParentNode returns null). If a shallow clone is being performed (deep is false), the new node will not have any children or siblings. If a deep clone is being performed, the new node will form the root of a complete cloned subtree.

**Specified by:**[cloneNode](http://docs.google.com/org/w3c/dom/Node.html#cloneNode(boolean)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**deep - if true, recursively clone the subtree under the specified node; if false, clone only the node itself. **Returns:**the duplicate node.

### normalize

public void **normalize**()

Does nothing, since IIOMetadataNodes do not contain Text children.

**Specified by:**[normalize](http://docs.google.com/org/w3c/dom/Node.html#normalize()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html)

### isSupported

public boolean **isSupported**([String](http://docs.google.com/java/lang/String.html) feature,  
 [String](http://docs.google.com/java/lang/String.html) version)

Returns false since DOM features are not supported.

**Specified by:**[isSupported](http://docs.google.com/org/w3c/dom/Node.html#isSupported(java.lang.String,%20java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**feature - a String, which is ignored.version - a String, which is ignored. **Returns:**false.

### getNamespaceURI

public [String](http://docs.google.com/java/lang/String.html) **getNamespaceURI**()  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

Returns null, since namespaces are not supported.

**Specified by:**[getNamespaceURI](http://docs.google.com/org/w3c/dom/Node.html#getNamespaceURI()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

### getPrefix

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

Returns null, since namespaces are not supported.

**Specified by:**[getPrefix](http://docs.google.com/org/w3c/dom/Node.html#getPrefix()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**null.**See Also:**[setPrefix(java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setPrefix(java.lang.String))

### setPrefix

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

Does nothing, since namespaces are not supported.

**Specified by:**[setPrefix](http://docs.google.com/org/w3c/dom/Node.html#setPrefix(java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**prefix - a String, which is ignored.**See Also:**[getPrefix()](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getPrefix())

### getLocalName

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

Equivalent to getNodeName.

**Specified by:**[getLocalName](http://docs.google.com/org/w3c/dom/Node.html#getLocalName()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**the node name, as a String.

### getTagName

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

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#getTagName()) The name of the element. If Node.localName is different from null, this attribute is a qualified name. For example, in:

<elementExample id="demo"> ...   
 </elementExample> ,

tagName has the value "elementExample". Note that this is case-preserving in XML, as are all of the operations of the DOM. The HTML DOM returns the tagName of an HTML element in the canonical uppercase form, regardless of the case in the source HTML document.

**Specified by:**[getTagName](http://docs.google.com/org/w3c/dom/Element.html#getTagName()) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html)

### getAttribute

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

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#getAttribute(java.lang.String)) Retrieves an attribute value by name.

**Specified by:**[getAttribute](http://docs.google.com/org/w3c/dom/Element.html#getAttribute(java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**name - The name of the attribute to retrieve. **Returns:**The Attr value as a string, or the empty string if that attribute does not have a specified or default value.

### getAttributeNS

public [String](http://docs.google.com/java/lang/String.html) **getAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)

Equivalent to getAttribute(localName).

**Specified by:**[getAttributeNS](http://docs.google.com/org/w3c/dom/Element.html#getAttributeNS(java.lang.String,%20java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**namespaceURI - The namespace URI of the attribute to retrieve.localName - The local name of the attribute to retrieve. **Returns:**The Attr value as a string, or the empty string if that attribute does not have a specified or default value.**See Also:**[setAttributeNS(java.lang.String, java.lang.String, java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setAttributeNS(java.lang.String,%20java.lang.String,%20java.lang.String))

### setAttribute

public void **setAttribute**([String](http://docs.google.com/java/lang/String.html) name,  
 [String](http://docs.google.com/java/lang/String.html) value)

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#setAttribute(java.lang.String,%20java.lang.String)) Adds a new attribute. If an attribute with that name is already present in the element, its value is changed to be that of the value parameter. This value is a simple string; it is not parsed as it is being set. So any markup (such as syntax to be recognized as an entity reference) is treated as literal text, and needs to be appropriately escaped by the implementation when it is written out. In order to assign an attribute value that contains entity references, the user must create an Attr node plus any Text and EntityReference nodes, build the appropriate subtree, and use setAttributeNode to assign it as the value of an attribute.

To set an attribute with a qualified name and namespace URI, use the setAttributeNS method.

**Specified by:**[setAttribute](http://docs.google.com/org/w3c/dom/Element.html#setAttribute(java.lang.String,%20java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**name - The name of the attribute to create or alter.value - Value to set in string form.

### setAttributeNS

public void **setAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) qualifiedName,  
 [String](http://docs.google.com/java/lang/String.html) value)

Equivalent to setAttribute(qualifiedName, value).

**Specified by:**[setAttributeNS](http://docs.google.com/org/w3c/dom/Element.html#setAttributeNS(java.lang.String,%20java.lang.String,%20java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**namespaceURI - The namespace URI of the attribute to create or alter.qualifiedName - The qualified name of the attribute to create or alter.value - The value to set in string form.**See Also:**[getAttributeNS(java.lang.String, java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getAttributeNS(java.lang.String,%20java.lang.String))

### removeAttribute

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

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#removeAttribute(java.lang.String)) Removes an attribute by name. If a default value for the removed attribute is defined in the DTD, a new attribute immediately appears with the default value as well as the corresponding namespace URI, local name, and prefix when applicable. The implementation may handle default values from other schemas similarly but applications should use Document.normalizeDocument() to guarantee this information is up-to-date.

If no attribute with this name is found, this method has no effect.

To remove an attribute by local name and namespace URI, use the removeAttributeNS method.

**Specified by:**[removeAttribute](http://docs.google.com/org/w3c/dom/Element.html#removeAttribute(java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**name - The name of the attribute to remove.

### removeAttributeNS

public void **removeAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)

Equivalent to removeAttribute(localName).

**Specified by:**[removeAttributeNS](http://docs.google.com/org/w3c/dom/Element.html#removeAttributeNS(java.lang.String,%20java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**namespaceURI - The namespace URI of the attribute to remove.localName - The local name of the attribute to remove.

### getAttributeNode

public [Attr](http://docs.google.com/org/w3c/dom/Attr.html) **getAttributeNode**([String](http://docs.google.com/java/lang/String.html) name)

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#getAttributeNode(java.lang.String)) Retrieves an attribute node by name.

To retrieve an attribute node by qualified name and namespace URI, use the getAttributeNodeNS method.

**Specified by:**[getAttributeNode](http://docs.google.com/org/w3c/dom/Element.html#getAttributeNode(java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**name - The name (nodeName) of the attribute to retrieve. **Returns:**The Attr node with the specified name ( nodeName) or null if there is no such attribute.

### getAttributeNodeNS

public [Attr](http://docs.google.com/org/w3c/dom/Attr.html) **getAttributeNodeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)

Equivalent to getAttributeNode(localName).

**Specified by:**[getAttributeNodeNS](http://docs.google.com/org/w3c/dom/Element.html#getAttributeNodeNS(java.lang.String,%20java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**namespaceURI - The namespace URI of the attribute to retrieve.localName - The local name of the attribute to retrieve. **Returns:**The Attr node with the specified attribute local name and namespace URI or null if there is no such attribute.**See Also:**[setAttributeNodeNS(org.w3c.dom.Attr)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setAttributeNodeNS(org.w3c.dom.Attr))

### setAttributeNode

public [Attr](http://docs.google.com/org/w3c/dom/Attr.html) **setAttributeNode**([Attr](http://docs.google.com/org/w3c/dom/Attr.html) newAttr)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#setAttributeNode(org.w3c.dom.Attr)) Adds a new attribute node. If an attribute with that name ( nodeName) is already present in the element, it is replaced by the new one. Replacing an attribute node by itself has no effect.

To add a new attribute node with a qualified name and namespace URI, use the setAttributeNodeNS method.

**Specified by:**[setAttributeNode](http://docs.google.com/org/w3c/dom/Element.html#setAttributeNode(org.w3c.dom.Attr)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**newAttr - The Attr node to add to the attribute list. **Returns:**If the newAttr attribute replaces an existing attribute, the replaced Attr node is returned, otherwise null is returned. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - WRONG\_DOCUMENT\_ERR: Raised if newAttr was created from a different document than the one that created the element.

NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

INUSE\_ATTRIBUTE\_ERR: Raised if newAttr is already an attribute of another Element object. The DOM user must explicitly clone Attr nodes to re-use them in other elements.

### setAttributeNodeNS

public [Attr](http://docs.google.com/org/w3c/dom/Attr.html) **setAttributeNodeNS**([Attr](http://docs.google.com/org/w3c/dom/Attr.html) newAttr)

Equivalent to setAttributeNode(newAttr).

**Specified by:**[setAttributeNodeNS](http://docs.google.com/org/w3c/dom/Element.html#setAttributeNodeNS(org.w3c.dom.Attr)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**newAttr - The Attr node to add to the attribute list. **Returns:**If the newAttr attribute replaces an existing attribute with the same local name and namespace URI, the replaced Attr node is returned, otherwise null is returned.**See Also:**[getAttributeNodeNS(java.lang.String, java.lang.String)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getAttributeNodeNS(java.lang.String,%20java.lang.String))

### removeAttributeNode

public [Attr](http://docs.google.com/org/w3c/dom/Attr.html) **removeAttributeNode**([Attr](http://docs.google.com/org/w3c/dom/Attr.html) oldAttr)

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#removeAttributeNode(org.w3c.dom.Attr)) Removes the specified attribute node. If a default value for the removed Attr node is defined in the DTD, a new node immediately appears with the default value as well as the corresponding namespace URI, local name, and prefix when applicable. The implementation may handle default values from other schemas similarly but applications should use Document.normalizeDocument() to guarantee this information is up-to-date.

**Specified by:**[removeAttributeNode](http://docs.google.com/org/w3c/dom/Element.html#removeAttributeNode(org.w3c.dom.Attr)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**oldAttr - The Attr node to remove from the attribute list. **Returns:**The Attr node that was removed.

### getElementsByTagName

public [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) **getElementsByTagName**([String](http://docs.google.com/java/lang/String.html) name)

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#getElementsByTagName(java.lang.String)) Returns a NodeList of all descendant Elements with a given tag name, in document order.

**Specified by:**[getElementsByTagName](http://docs.google.com/org/w3c/dom/Element.html#getElementsByTagName(java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**name - The name of the tag to match on. The special value "\*" matches all tags. **Returns:**A list of matching Element nodes.

### getElementsByTagNameNS

public [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) **getElementsByTagNameNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)

Equivalent to getElementsByTagName(localName).

**Specified by:**[getElementsByTagNameNS](http://docs.google.com/org/w3c/dom/Element.html#getElementsByTagNameNS(java.lang.String,%20java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**namespaceURI - The namespace URI of the elements to match on. The special value "\*" matches all namespaces.localName - The local name of the elements to match on. The special value "\*" matches all local names. **Returns:**A new NodeList object containing all the matched Elements.

### hasAttributes

public boolean **hasAttributes**()

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#hasAttributes()) Returns whether this node (if it is an element) has any attributes.

**Specified by:**[hasAttributes](http://docs.google.com/org/w3c/dom/Node.html#hasAttributes()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Returns:**Returns true if this node has any attributes, false otherwise.

### hasAttribute

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

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#hasAttribute(java.lang.String)) Returns true when an attribute with a given name is specified on this element or has a default value, false otherwise.

**Specified by:**[hasAttribute](http://docs.google.com/org/w3c/dom/Element.html#hasAttribute(java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**name - The name of the attribute to look for. **Returns:**true if an attribute with the given name is specified on this element or has a default value, false otherwise.

### hasAttributeNS

public boolean **hasAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName)

Equivalent to hasAttribute(localName).

**Specified by:**[hasAttributeNS](http://docs.google.com/org/w3c/dom/Element.html#hasAttributeNS(java.lang.String,%20java.lang.String)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**namespaceURI - The namespace URI of the attribute to look for.localName - The local name of the attribute to look for. **Returns:**true if an attribute with the given local name and namespace URI is specified or has a default value on this element, false otherwise.

### getLength

public int **getLength**()

**Description copied from interface:** [**NodeList**](http://docs.google.com/org/w3c/dom/NodeList.html#getLength()) The number of nodes in the list. The range of valid child node indices is 0 to length-1 inclusive.

**Specified by:**[getLength](http://docs.google.com/org/w3c/dom/NodeList.html#getLength()) in interface [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html)

### item

public [Node](http://docs.google.com/org/w3c/dom/Node.html) **item**(int index)

**Description copied from interface:** [**NodeList**](http://docs.google.com/org/w3c/dom/NodeList.html#item(int)) Returns the indexth item in the collection. If index is greater than or equal to the number of nodes in the list, this returns null.

**Specified by:**[item](http://docs.google.com/org/w3c/dom/NodeList.html#item(int)) in interface [NodeList](http://docs.google.com/org/w3c/dom/NodeList.html) **Parameters:**index - Index into the collection. **Returns:**The node at the indexth position in the NodeList, or null if that is not a valid index.

### getUserObject

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

Returns the Object value associated with this node.

**Returns:**the user Object.**See Also:**[setUserObject(java.lang.Object)](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#setUserObject(java.lang.Object))

### setUserObject

public void **setUserObject**([Object](http://docs.google.com/java/lang/Object.html) userObject)

Sets the value associated with this node.

**Parameters:**userObject - the user Object.**See Also:**[getUserObject()](http://docs.google.com/javax/imageio/metadata/IIOMetadataNode.html#getUserObject())

### setIdAttribute

public void **setIdAttribute**([String](http://docs.google.com/java/lang/String.html) name,  
 boolean isId)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#setIdAttribute(java.lang.String,%20boolean)) If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . This affects the value of Attr.isId and the behavior of Document.getElementById, but does not change any schema that may be in use, in particular this does not affect the Attr.schemaTypeInfo of the specified Attr node. Use the value false for the parameter isId to undeclare an attribute for being a user-determined ID attribute.

To specify an attribute by local name and namespace URI, use the setIdAttributeNS method.

**Specified by:**[setIdAttribute](http://docs.google.com/org/w3c/dom/Element.html#setIdAttribute(java.lang.String,%20boolean)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**name - The name of the attribute.isId - Whether the attribute is a of type ID. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

NOT\_FOUND\_ERR: Raised if the specified node is not an attribute of this element.

### setIdAttributeNS

public void **setIdAttributeNS**([String](http://docs.google.com/java/lang/String.html) namespaceURI,  
 [String](http://docs.google.com/java/lang/String.html) localName,  
 boolean isId)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#setIdAttributeNS(java.lang.String,%20java.lang.String,%20boolean)) If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . This affects the value of Attr.isId and the behavior of Document.getElementById, but does not change any schema that may be in use, in particular this does not affect the Attr.schemaTypeInfo of the specified Attr node. Use the value false for the parameter isId to undeclare an attribute for being a user-determined ID attribute.

**Specified by:**[setIdAttributeNS](http://docs.google.com/org/w3c/dom/Element.html#setIdAttributeNS(java.lang.String,%20java.lang.String,%20boolean)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**namespaceURI - The namespace URI of the attribute.localName - The local name of the attribute.isId - Whether the attribute is a of type ID. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

NOT\_FOUND\_ERR: Raised if the specified node is not an attribute of this element.

### setIdAttributeNode

public void **setIdAttributeNode**([Attr](http://docs.google.com/org/w3c/dom/Attr.html) idAttr,  
 boolean isId)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#setIdAttributeNode(org.w3c.dom.Attr,%20boolean)) If the parameter isId is true, this method declares the specified attribute to be a user-determined ID attribute . This affects the value of Attr.isId and the behavior of Document.getElementById, but does not change any schema that may be in use, in particular this does not affect the Attr.schemaTypeInfo of the specified Attr node. Use the value false for the parameter isId to undeclare an attribute for being a user-determined ID attribute.

**Specified by:**[setIdAttributeNode](http://docs.google.com/org/w3c/dom/Element.html#setIdAttributeNode(org.w3c.dom.Attr,%20boolean)) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html) **Parameters:**idAttr - The attribute node.isId - Whether the attribute is a of type ID. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised if this node is readonly.

NOT\_FOUND\_ERR: Raised if the specified node is not an attribute of this element.

### getSchemaTypeInfo

public [TypeInfo](http://docs.google.com/org/w3c/dom/TypeInfo.html) **getSchemaTypeInfo**()

**Description copied from interface:** [**Element**](http://docs.google.com/org/w3c/dom/Element.html#getSchemaTypeInfo()) The type information associated with this element.

**Specified by:**[getSchemaTypeInfo](http://docs.google.com/org/w3c/dom/Element.html#getSchemaTypeInfo()) in interface [Element](http://docs.google.com/org/w3c/dom/Element.html)

### setUserData

public [Object](http://docs.google.com/java/lang/Object.html) **setUserData**([String](http://docs.google.com/java/lang/String.html) key,  
 [Object](http://docs.google.com/java/lang/Object.html) data,  
 [UserDataHandler](http://docs.google.com/org/w3c/dom/UserDataHandler.html) handler)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#setUserData(java.lang.String,%20java.lang.Object,%20org.w3c.dom.UserDataHandler)) Associate an object to a key on this node. The object can later be retrieved from this node by calling getUserData with the same key.

**Specified by:**[setUserData](http://docs.google.com/org/w3c/dom/Node.html#setUserData(java.lang.String,%20java.lang.Object,%20org.w3c.dom.UserDataHandler)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**key - The key to associate the object to.data - The object to associate to the given key, or null to remove any existing association to that key.handler - The handler to associate to that key, or null. **Returns:**Returns the DOMUserData previously associated to the given key on this node, or null if there was none.

### getUserData

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

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#getUserData(java.lang.String)) Retrieves the object associated to a key on a this node. The object must first have been set to this node by calling setUserData with the same key.

**Specified by:**[getUserData](http://docs.google.com/org/w3c/dom/Node.html#getUserData(java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**key - The key the object is associated to. **Returns:**Returns the DOMUserData associated to the given key on this node, or null if there was none.

### getFeature

public [Object](http://docs.google.com/java/lang/Object.html) **getFeature**([String](http://docs.google.com/java/lang/String.html) feature,  
 [String](http://docs.google.com/java/lang/String.html) version)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#getFeature(java.lang.String,%20java.lang.String)) This method returns a specialized object which implements the specialized APIs of the specified feature and version, as specified in . The specialized object may also be obtained by using binding-specific casting methods but is not necessarily expected to, as discussed in . This method also allow the implementation to provide specialized objects which do not support the Node interface.

**Specified by:**[getFeature](http://docs.google.com/org/w3c/dom/Node.html#getFeature(java.lang.String,%20java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**feature - The name of the feature requested. Note that any plus sign "+" prepended to the name of the feature will be ignored since it is not significant in the context of this method.version - This is the version number of the feature to test. **Returns:**Returns an object which implements the specialized APIs of the specified feature and version, if any, or null if there is no object which implements interfaces associated with that feature. If the DOMObject returned by this method implements the Node interface, it must delegate to the primary core Node and not return results inconsistent with the primary core Node such as attributes, childNodes, etc.

### isSameNode

public boolean **isSameNode**([Node](http://docs.google.com/org/w3c/dom/Node.html) node)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#isSameNode(org.w3c.dom.Node)) Returns whether this node is the same node as the given one.

This method provides a way to determine whether two Node references returned by the implementation reference the same object. When two Node references are references to the same object, even if through a proxy, the references may be used completely interchangeably, such that all attributes have the same values and calling the same DOM method on either reference always has exactly the same effect.

**Specified by:**[isSameNode](http://docs.google.com/org/w3c/dom/Node.html#isSameNode(org.w3c.dom.Node)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**node - The node to test against. **Returns:**Returns true if the nodes are the same, false otherwise.

### isEqualNode

public boolean **isEqualNode**([Node](http://docs.google.com/org/w3c/dom/Node.html) node)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#isEqualNode(org.w3c.dom.Node)) Tests whether two nodes are equal.

This method tests for equality of nodes, not sameness (i.e., whether the two nodes are references to the same object) which can be tested with Node.isSameNode(). All nodes that are the same will also be equal, though the reverse may not be true.

Two nodes are equal if and only if the following conditions are satisfied:

* The two nodes are of the same type.
* The following string attributes are equal: nodeName, localName, namespaceURI, prefix, nodeValue . This is: they are both null, or they have the same length and are character for character identical.
* The attributes NamedNodeMaps are equal. This is: they are both null, or they have the same length and for each node that exists in one map there is a node that exists in the other map and is equal, although not necessarily at the same index.
* The childNodes NodeLists are equal. This is: they are both null, or they have the same length and contain equal nodes at the same index. Note that normalization can affect equality; to avoid this, nodes should be normalized before being compared.

For two DocumentType nodes to be equal, the following conditions must also be satisfied:

* The following string attributes are equal: publicId, systemId, internalSubset.
* The entities NamedNodeMaps are equal.
* The notations NamedNodeMaps are equal.

On the other hand, the following do not affect equality: the ownerDocument, baseURI, and parentNode attributes, the specified attribute for Attr nodes, the schemaTypeInfo attribute for Attr and Element nodes, the Text.isElementContentWhitespace attribute for Text nodes, as well as any user data or event listeners registered on the nodes.

**Note:** As a general rule, anything not mentioned in the description above is not significant in consideration of equality checking. Note that future versions of this specification may take into account more attributes and implementations conform to this specification are expected to be updated accordingly.

**Specified by:**[isEqualNode](http://docs.google.com/org/w3c/dom/Node.html#isEqualNode(org.w3c.dom.Node)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**node - The node to compare equality with. **Returns:**Returns true if the nodes are equal, false otherwise.

### lookupNamespaceURI

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

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#lookupNamespaceURI(java.lang.String)) Look up the namespace URI associated to the given prefix, starting from this node.

See for details on the algorithm used by this method.

**Specified by:**[lookupNamespaceURI](http://docs.google.com/org/w3c/dom/Node.html#lookupNamespaceURI(java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**prefix - The prefix to look for. If this parameter is null, the method will return the default namespace URI if any. **Returns:**Returns the associated namespace URI or null if none is found.

### isDefaultNamespace

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

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#isDefaultNamespace(java.lang.String)) This method checks if the specified namespaceURI is the default namespace or not.

**Specified by:**[isDefaultNamespace](http://docs.google.com/org/w3c/dom/Node.html#isDefaultNamespace(java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**namespaceURI - The namespace URI to look for. **Returns:**Returns true if the specified namespaceURI is the default namespace, false otherwise.

### lookupPrefix

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

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#lookupPrefix(java.lang.String)) Look up the prefix associated to the given namespace URI, starting from this node. The default namespace declarations are ignored by this method.

See for details on the algorithm used by this method.

**Specified by:**[lookupPrefix](http://docs.google.com/org/w3c/dom/Node.html#lookupPrefix(java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**namespaceURI - The namespace URI to look for. **Returns:**Returns an associated namespace prefix if found or null if none is found. If more than one prefix are associated to the namespace prefix, the returned namespace prefix is implementation dependent.

### getTextContent

public [String](http://docs.google.com/java/lang/String.html) **getTextContent**()  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#getTextContent()) This attribute returns the text content of this node and its descendants. When it is defined to be null, setting it has no effect. On setting, any possible children this node may have are removed and, if it the new string is not empty or null, replaced by a single Text node containing the string this attribute is set to.

On getting, no serialization is performed, the returned string does not contain any markup. No whitespace normalization is performed and the returned string does not contain the white spaces in element content (see the attribute Text.isElementContentWhitespace). Similarly, on setting, no parsing is performed either, the input string is taken as pure textual content.

The string returned is made of the text content of this node depending on its type, as defined below:

| Node type | Content |
| --- | --- |
| ELEMENT\_NODE, ATTRIBUTE\_NODE, ENTITY\_NODE, ENTITY\_REFERENCE\_NODE, DOCUMENT\_FRAGMENT\_NODE | concatenation of the textContent attribute value of every child node, excluding COMMENT\_NODE and PROCESSING\_INSTRUCTION\_NODE nodes. This is the empty string if the node has no children. |
| TEXT\_NODE, CDATA\_SECTION\_NODE, COMMENT\_NODE, PROCESSING\_INSTRUCTION\_NODE | nodeValue |
| DOCUMENT\_NODE, DOCUMENT\_TYPE\_NODE, NOTATION\_NODE | *null* |

**Specified by:**[getTextContent](http://docs.google.com/org/w3c/dom/Node.html#getTextContent()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - DOMSTRING\_SIZE\_ERR: Raised when it would return more characters than fit in a DOMString variable on the implementation platform.

### setTextContent

public void **setTextContent**([String](http://docs.google.com/java/lang/String.html) textContent)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#setTextContent(java.lang.String)) This attribute returns the text content of this node and its descendants. When it is defined to be null, setting it has no effect. On setting, any possible children this node may have are removed and, if it the new string is not empty or null, replaced by a single Text node containing the string this attribute is set to.

On getting, no serialization is performed, the returned string does not contain any markup. No whitespace normalization is performed and the returned string does not contain the white spaces in element content (see the attribute Text.isElementContentWhitespace). Similarly, on setting, no parsing is performed either, the input string is taken as pure textual content.

The string returned is made of the text content of this node depending on its type, as defined below:

| Node type | Content |
| --- | --- |
| ELEMENT\_NODE, ATTRIBUTE\_NODE, ENTITY\_NODE, ENTITY\_REFERENCE\_NODE, DOCUMENT\_FRAGMENT\_NODE | concatenation of the textContent attribute value of every child node, excluding COMMENT\_NODE and PROCESSING\_INSTRUCTION\_NODE nodes. This is the empty string if the node has no children. |
| TEXT\_NODE, CDATA\_SECTION\_NODE, COMMENT\_NODE, PROCESSING\_INSTRUCTION\_NODE | nodeValue |
| DOCUMENT\_NODE, DOCUMENT\_TYPE\_NODE, NOTATION\_NODE | *null* |

**Specified by:**[setTextContent](http://docs.google.com/org/w3c/dom/Node.html#setTextContent(java.lang.String)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NO\_MODIFICATION\_ALLOWED\_ERR: Raised when the node is readonly.

### compareDocumentPosition

public short **compareDocumentPosition**([Node](http://docs.google.com/org/w3c/dom/Node.html) other)  
 throws [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html)

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#compareDocumentPosition(org.w3c.dom.Node)) Compares the reference node, i.e. the node on which this method is being called, with a node, i.e. the one passed as a parameter, with regard to their position in the document and according to the document order.

**Specified by:**[compareDocumentPosition](http://docs.google.com/org/w3c/dom/Node.html#compareDocumentPosition(org.w3c.dom.Node)) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html) **Parameters:**other - The node to compare against the reference node. **Returns:**Returns how the node is positioned relatively to the reference node. **Throws:** [DOMException](http://docs.google.com/org/w3c/dom/DOMException.html) - NOT\_SUPPORTED\_ERR: when the compared nodes are from different DOM implementations that do not coordinate to return consistent implementation-specific results.

### getBaseURI

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

**Description copied from interface:** [**Node**](http://docs.google.com/org/w3c/dom/Node.html#getBaseURI()) The absolute base URI of this node or null if the implementation wasn't able to obtain an absolute URI. This value is computed as described in . However, when the Document supports the feature "HTML" [[DOM Level 2 HTML](http://www.w3.org/TR/2003/REC-DOM-Level-2-HTML-20030109)] , the base URI is computed using first the value of the href attribute of the HTML BASE element if any, and the value of the documentURI attribute from the Document interface otherwise.

**Specified by:**[getBaseURI](http://docs.google.com/org/w3c/dom/Node.html#getBaseURI()) in interface [Node](http://docs.google.com/org/w3c/dom/Node.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/IIOMetadataNode.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/IIOMetadataFormatImpl.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/imageio/metadata/IIOMetadataNode.html)    [**NO FRAMES**](http://docs.google.com/IIOMetadataNode.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#3rdcrjn) |

[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).