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

## **javax.xml.stream**

Interface XMLEventWriter

**All Superinterfaces:** [XMLEventConsumer](http://docs.google.com/javax/xml/stream/util/XMLEventConsumer.html)

public interface **XMLEventWriter**extends [XMLEventConsumer](http://docs.google.com/javax/xml/stream/util/XMLEventConsumer.html)

This is the top level interface for writing XML documents. Instances of this interface are not required to validate the form of the XML.

**Since:** 1.6 **See Also:**[XMLEventReader](http://docs.google.com/javax/xml/stream/XMLEventReader.html), [XMLEvent](http://docs.google.com/javax/xml/stream/events/XMLEvent.html), [Characters](http://docs.google.com/javax/xml/stream/events/Characters.html), [ProcessingInstruction](http://docs.google.com/javax/xml/stream/events/ProcessingInstruction.html), [StartElement](http://docs.google.com/javax/xml/stream/events/StartElement.html), [EndElement](http://docs.google.com/javax/xml/stream/events/EndElement.html)

| **Method Summary** | |
| --- | --- |
| void | [**add**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#add(javax.xml.stream.events.XMLEvent))([XMLEvent](http://docs.google.com/javax/xml/stream/events/XMLEvent.html) event)            Add an event to the output stream Adding a START\_ELEMENT will open a new namespace scope that will be closed when the corresponding END\_ELEMENT is written. |
| void | [**add**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#add(javax.xml.stream.XMLEventReader))([XMLEventReader](http://docs.google.com/javax/xml/stream/XMLEventReader.html) reader)            Adds an entire stream to an output stream, calls next() on the inputStream argument until hasNext() returns false This should be treated as a convenience method that will perform the following loop over all the events in an event reader and call add on each event. |
| void | [**close**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#close())()            Frees any resources associated with this stream |
| void | [**flush**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#flush())()            Writes any cached events to the underlying output mechanism |
| [NamespaceContext](http://docs.google.com/javax/xml/namespace/NamespaceContext.html) | [**getNamespaceContext**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#getNamespaceContext())()            Returns the current namespace context. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPrefix**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#getPrefix(java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri)            Gets the prefix the uri is bound to |
| void | [**setDefaultNamespace**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#setDefaultNamespace(java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri)            Binds a URI to the default namespace This URI is bound in the scope of the current START\_ELEMENT / END\_ELEMENT pair. |
| void | [**setNamespaceContext**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#setNamespaceContext(javax.xml.namespace.NamespaceContext))([NamespaceContext](http://docs.google.com/javax/xml/namespace/NamespaceContext.html) context)            Sets the current namespace context for prefix and uri bindings. |
| void | [**setPrefix**](http://docs.google.com/javax/xml/stream/XMLEventWriter.html#setPrefix(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) prefix, [String](http://docs.google.com/java/lang/String.html) uri)            Sets the prefix the uri is bound to. |

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

### flush

void **flush**()  
 throws [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

Writes any cached events to the underlying output mechanism

**Throws:** [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

### close

void **close**()  
 throws [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

Frees any resources associated with this stream

**Throws:** [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

### add

void **add**([XMLEvent](http://docs.google.com/javax/xml/stream/events/XMLEvent.html) event)  
 throws [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

Add an event to the output stream Adding a START\_ELEMENT will open a new namespace scope that will be closed when the corresponding END\_ELEMENT is written.

| Required and optional fields for events added to the writer | |
| --- | --- |
| Event Type | Required Fields | Optional Fields | Required Behavior |
| START\_ELEMENT | QName name | namespaces , attributes | A START\_ELEMENT will be written by writing the name, namespaces, and attributes of the event in XML 1.0 valid syntax for START\_ELEMENTs. The name is written by looking up the prefix for the namespace uri. The writer can be configured to respect prefixes of QNames. If the writer is respecting prefixes it must use the prefix set on the QName. The default behavior is to lookup the value for the prefix on the EventWriter's internal namespace context. Each attribute (if any) is written using the behavior specified in the attribute section of this table. Each namespace (if any) is written using the behavior specified in the namespace section of this table. |
| END\_ELEMENT | Qname name | None | A well formed END\_ELEMENT tag is written. The name is written by looking up the prefix for the namespace uri. The writer can be configured to respect prefixes of QNames. If the writer is respecting prefixes it must use the prefix set on the QName. The default behavior is to lookup the value for the prefix on the EventWriter's internal namespace context. If the END\_ELEMENT name does not match the START\_ELEMENT name an XMLStreamException is thrown. |
| ATTRIBUTE | QName name , String value | QName type | An attribute is written using the same algorithm to find the lexical form as used in START\_ELEMENT. The default is to use double quotes to wrap attribute values and to escape any double quotes found in the value. The type value is ignored. |
| NAMESPACE | String prefix, String namespaceURI, boolean isDefaultNamespaceDeclaration | None | A namespace declaration is written. If the namespace is a default namespace declaration (isDefault is true) then xmlns="$namespaceURI" is written and the prefix is optional. If isDefault is false, the prefix must be declared and the writer must prepend xmlns to the prefix and write out a standard prefix declaration. |
| PROCESSING\_INSTRUCTION | None | String target, String data | The data does not need to be present and may be null. Target is required and many not be null. The writer will write data section directly after the target, enclosed in appropriate XML 1.0 syntax |
| COMMENT | None | String comment | If the comment is present (not null) it is written, otherwise an an empty comment is written |
| START\_DOCUMENT | None | String encoding , boolean standalone, String version | A START\_DOCUMENT event is not required to be written to the stream. If present the attributes are written inside the appropriate XML declaration syntax |
| END\_DOCUMENT | None | None | Nothing is written to the output |
| DTD | String DocumentTypeDefinition | None | The DocumentTypeDefinition is written to the output |

**Specified by:**[add](http://docs.google.com/javax/xml/stream/util/XMLEventConsumer.html#add(javax.xml.stream.events.XMLEvent)) in interface [XMLEventConsumer](http://docs.google.com/javax/xml/stream/util/XMLEventConsumer.html) **Parameters:**event - the event to be added **Throws:** [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

### add

void **add**([XMLEventReader](http://docs.google.com/javax/xml/stream/XMLEventReader.html) reader)  
 throws [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

Adds an entire stream to an output stream, calls next() on the inputStream argument until hasNext() returns false This should be treated as a convenience method that will perform the following loop over all the events in an event reader and call add on each event.

**Parameters:**reader - the event stream to add to the output **Throws:** [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

### getPrefix

[String](http://docs.google.com/java/lang/String.html) **getPrefix**([String](http://docs.google.com/java/lang/String.html) uri)  
 throws [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

Gets the prefix the uri is bound to

**Parameters:**uri - the uri to look up **Throws:** [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

### setPrefix

void **setPrefix**([String](http://docs.google.com/java/lang/String.html) prefix,  
 [String](http://docs.google.com/java/lang/String.html) uri)  
 throws [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

Sets the prefix the uri is bound to. This prefix is bound in the scope of the current START\_ELEMENT / END\_ELEMENT pair. If this method is called before a START\_ELEMENT has been written the prefix is bound in the root scope.

**Parameters:**prefix - the prefix to bind to the uriuri - the uri to bind to the prefix **Throws:** [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

### setDefaultNamespace

void **setDefaultNamespace**([String](http://docs.google.com/java/lang/String.html) uri)  
 throws [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

Binds a URI to the default namespace This URI is bound in the scope of the current START\_ELEMENT / END\_ELEMENT pair. If this method is called before a START\_ELEMENT has been written the uri is bound in the root scope.

**Parameters:**uri - the uri to bind to the default namespace **Throws:** [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

### setNamespaceContext

void **setNamespaceContext**([NamespaceContext](http://docs.google.com/javax/xml/namespace/NamespaceContext.html) context)  
 throws [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

Sets the current namespace context for prefix and uri bindings. This context becomes the root namespace context for writing and will replace the current root namespace context. Subsequent calls to setPrefix and setDefaultNamespace will bind namespaces using the context passed to the method as the root context for resolving namespaces.

**Parameters:**context - the namespace context to use for this writer **Throws:** [XMLStreamException](http://docs.google.com/javax/xml/stream/XMLStreamException.html)

### getNamespaceContext

[NamespaceContext](http://docs.google.com/javax/xml/namespace/NamespaceContext.html) **getNamespaceContext**()

Returns the current namespace context.

**Returns:**the current namespace context

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

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).