| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/xml/stream/util/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/xml/transform/dom/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/transform/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.xml.transform

This package defines the generic APIs for processing transformation instructions, and performing a transformation from source to result.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**ErrorListener**](http://docs.google.com/javax/xml/transform/ErrorListener.html) | To provide customized error handling, implement this interface and use the setErrorListener method to register an instance of the implmentation with the [Transformer](http://docs.google.com/javax/xml/transform/Transformer.html). |
| [**Result**](http://docs.google.com/javax/xml/transform/Result.html) | An object that implements this interface contains the information needed to build a transformation result tree. |
| [**Source**](http://docs.google.com/javax/xml/transform/Source.html) | An object that implements this interface contains the information needed to act as source input (XML source or transformation instructions). |
| [**SourceLocator**](http://docs.google.com/javax/xml/transform/SourceLocator.html) | This interface is primarily for the purposes of reporting where an error occurred in the XML source or transformation instructions. |
| [**Templates**](http://docs.google.com/javax/xml/transform/Templates.html) | An object that implements this interface is the runtime representation of processed transformation instructions. |
| [**URIResolver**](http://docs.google.com/javax/xml/transform/URIResolver.html) | An object that implements this interface that can be called by the processor to turn a URI used in document(), xsl:import, or xsl:include into a Source object. |

| **Class Summary** | |
| --- | --- |
| [**OutputKeys**](http://docs.google.com/javax/xml/transform/OutputKeys.html) | Provides string constants that can be used to set output properties for a Transformer, or to retrieve output properties from a Transformer or Templates object. |
| [**Transformer**](http://docs.google.com/javax/xml/transform/Transformer.html) | An instance of this abstract class can transform a source tree into a result tree. |
| [**TransformerFactory**](http://docs.google.com/javax/xml/transform/TransformerFactory.html) | A TransformerFactory instance can be used to create [Transformer](http://docs.google.com/javax/xml/transform/Transformer.html) and [Templates](http://docs.google.com/javax/xml/transform/Templates.html) objects. |

| **Exception Summary** | |
| --- | --- |
| [**TransformerConfigurationException**](http://docs.google.com/javax/xml/transform/TransformerConfigurationException.html) | Indicates a serious configuration error. |
| [**TransformerException**](http://docs.google.com/javax/xml/transform/TransformerException.html) | This class specifies an exceptional condition that occured during the transformation process. |

| **Error Summary** | |
| --- | --- |
| [**TransformerFactoryConfigurationError**](http://docs.google.com/javax/xml/transform/TransformerFactoryConfigurationError.html) | Thrown when a problem with configuration with the Transformer Factories exists. |

## Package javax.xml.transform Description

This package defines the generic APIs for processing transformation instructions, and performing a transformation from source to result. These interfaces have no dependencies on SAX or the DOM standard, and try to make as few assumptions as possible about the details of the source and result of a transformation. It achieves this by defining [Source](http://docs.google.com/javax/xml/transform/Source.html) and [Result](http://docs.google.com/javax/xml/transform/Result.html) interfaces.

To define concrete classes for the user, the API defines specializations of the interfaces found at the root level. These interfaces are found in [javax.xml.transform.sax](http://docs.google.com/javax/xml/transform/sax/package-summary.html), [javax.xml.transform.dom](http://docs.google.com/javax/xml/transform/dom/package-summary.html), and [javax.xml.transform.stream](http://docs.google.com/javax/xml/transform/stream/package-summary.html).

### Creating Objects

The API allows a concrete [TransformerFactory](http://docs.google.com/javax/xml/transform/TransformerFactory.html) object to be created from the static function [TransformerFactory.newInstance()](http://docs.google.com/javax/xml/transform/TransformerFactory.html#newInstance()).

### Specification of Inputs and Outputs

This API defines two interface objects called [Source](http://docs.google.com/javax/xml/transform/Source.html) and [Result](http://docs.google.com/javax/xml/transform/Result.html). In order to pass Source and Result objects to the interfaces, concrete classes must be used. Three concrete representations are defined for each of these objects: [StreamSource](http://docs.google.com/javax/xml/transform/stream/StreamSource.html) and [StreamResult](http://docs.google.com/javax/xml/transform/stream/StreamResult.html), [SAXSource](http://docs.google.com/javax/xml/transform/sax/SAXSource.html) and [SAXResult](http://docs.google.com/javax/xml/transform/sax/SAXResult.html), and [DOMSource](http://docs.google.com/javax/xml/transform/dom/DOMSource.html) and [DOMResult](http://docs.google.com/javax/xml/transform/dom/DOMResult.html). Each of these objects defines a FEATURE string (which is i the form of a URL), which can be passed into [TransformerFactory.getFeature(java.lang.String)](http://docs.google.com/javax/xml/transform/TransformerFactory.html#getFeature(java.lang.String)) to see if the given type of Source or Result object is supported. For instance, to test if a DOMSource and a StreamResult is supported, you can apply the following test.

TransformerFactory tfactory = TransformerFactory.newInstance();  
if (tfactory.getFeature(DOMSource.FEATURE) && tfactory.getFeature(StreamResult.FEATURE)) {  
...  
}

### Qualified Name Representation

[Namespaces](http://www.w3.org/TR/REC-xml-names) present something of a problem area when dealing with XML objects. Qualified Names appear in XML markup as prefixed names. But the prefixes themselves do not hold identity. Rather, it is the URIs that they contextually map to that hold the identity. Therefore, when passing a Qualified Name like "xyz:foo" among Java programs, one must provide a means to map "xyz" to a namespace.

One solution has been to create a "QName" object that holds the namespace URI, as well as the prefix and local name, but this is not always an optimal solution, as when, for example, you want to use unique strings as keys in a dictionary object. Not having a string representation also makes it difficult to specify a namespaced identity outside the context of an XML document.

In order to pass namespaced values to transformations, for instance when setting a property or a parameter on a [Transformer](http://docs.google.com/javax/xml/transform/Transformer.html) object, this specification defines that a String "qname" object parameter be passed as two-part string, the namespace URI enclosed in curly braces ({}), followed by the local name. If the qname has a null URI, then the String object only contains the local name. An application can safely check for a non-null URI by testing to see if the first character of the name is a '{' character.

For example, if a URI and local name were obtained from an element defined with <xyz:foo xmlns:xyz="http://xyz.foo.com/yada/baz.html"/>, then the Qualified Name would be "{http://xyz.foo.com/yada/baz.html}foo". Note that the prefix is lost.

### Result Tree Serialization

Serialization of the result tree to a stream can be controlled with the [Transformer.setOutputProperties(java.util.Properties)](http://docs.google.com/javax/xml/transform/Transformer.html#setOutputProperties(java.util.Properties)) and the [Transformer.setOutputProperty(java.lang.String, java.lang.String)](http://docs.google.com/javax/xml/transform/Transformer.html#setOutputProperty(java.lang.String,%20java.lang.String)) methods. These properties only apply to stream results, they have no effect when the result is a DOM tree or SAX event stream.

Strings that match the [XSLT specification for xsl:output attributes](http://www.w3.org/TR/xslt#output) can be referenced from the [OutputKeys](http://docs.google.com/javax/xml/transform/OutputKeys.html) class. Other strings can be specified as well. If the transformer does not recognize an output key, a [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) is thrown, unless the key name is [namespace qualified](#_2et92p0). Output key names that are namespace qualified are always allowed, although they may be ignored by some implementations.

If all that is desired is the simple identity transformation of a source to a result, then [TransformerFactory](http://docs.google.com/javax/xml/transform/TransformerFactory.html) provides a [TransformerFactory.newTransformer()](http://docs.google.com/javax/xml/transform/TransformerFactory.html#newTransformer()) method with no arguments. This method creates a Transformer that effectively copies the source to the result. This method may be used to create a DOM from SAX events or to create an XML or HTML stream from a DOM or SAX events.

### Exceptions and Error Reporting

The transformation API throw three types of specialized exceptions. A [TransformerFactoryConfigurationError](http://docs.google.com/javax/xml/transform/TransformerFactoryConfigurationError.html) is parallel to the [FactoryConfigurationError](http://docs.google.com/javax/xml/parsers/FactoryConfigurationError.html), and is thrown when a configuration problem with the TransformerFactory exists. This error will typically be thrown when the transformation factory class specified with the "javax.xml.transform.TransformerFactory" system property cannot be found or instantiated.

A [TransformerConfigurationException](http://docs.google.com/javax/xml/transform/TransformerConfigurationException.html) may be thrown if for any reason a Transformer can not be created. A TransformerConfigurationException may be thrown if there is a syntax error in the transformation instructions, for example when [TransformerFactory.newTransformer(javax.xml.transform.Source)](http://docs.google.com/javax/xml/transform/TransformerFactory.html#newTransformer(javax.xml.transform.Source)) is called.

[TransformerException](http://docs.google.com/javax/xml/transform/TransformerException.html) is a general exception that occurs during the course of a transformation. A transformer exception may wrap another exception, and if any of the [TransformerException.printStackTrace()](http://docs.google.com/javax/xml/transform/TransformerException.html#printStackTrace()) methods are called on it, it will produce a list of stack dumps, starting from the most recent. The transformer exception also provides a [SourceLocator](http://docs.google.com/javax/xml/transform/SourceLocator.html) object which indicates where in the source tree or transformation instructions the error occurred. [TransformerException.getMessageAndLocation()](http://docs.google.com/javax/xml/transform/TransformerException.html#getMessageAndLocation()) may be called to get an error message with location info, and [TransformerException.getLocationAsString()](http://docs.google.com/javax/xml/transform/TransformerException.html#getLocationAsString()) may be called to get just the location string.

Transformation warnings and errors are sent to an [ErrorListener](http://docs.google.com/javax/xml/transform/ErrorListener.html), at which point the application may decide to report the error or warning, and may decide to throw an Exception for a non-fatal error. The ErrorListener may be set via [TransformerFactory.setErrorListener(javax.xml.transform.ErrorListener)](http://docs.google.com/javax/xml/transform/TransformerFactory.html#setErrorListener(javax.xml.transform.ErrorListener)) for reporting errors that have to do with syntax errors in the transformation instructions, or via [Transformer.setErrorListener(javax.xml.transform.ErrorListener)](http://docs.google.com/javax/xml/transform/Transformer.html#setErrorListener(javax.xml.transform.ErrorListener)) to report errors that occur during the transformation. The ErrorListener on both objects will always be valid and non-null, whether set by the application or a default implementation provided by the processor. The default implementation provided by the processor will report all warnings and errors to System.err and does not throw any Exceptions. Applications are *strongly* encouraged to register and use ErrorListeners that insure proper behavior for warnings and errors.

### Resolution of URIs within a transformation

The API provides a way for URIs referenced from within the stylesheet instructions or within the transformation to be resolved by the calling application. This can be done by creating a class that implements the [URIResolver](http://docs.google.com/javax/xml/transform/URIResolver.html) interface, with its one method, [URIResolver.resolve(java.lang.String, java.lang.String)](http://docs.google.com/javax/xml/transform/URIResolver.html#resolve(java.lang.String,%20java.lang.String)), and use this class to set the URI resolution for the transformation instructions or transformation with [TransformerFactory.setURIResolver(javax.xml.transform.URIResolver)](http://docs.google.com/javax/xml/transform/TransformerFactory.html#setURIResolver(javax.xml.transform.URIResolver)) or [Transformer.setURIResolver(javax.xml.transform.URIResolver)](http://docs.google.com/javax/xml/transform/Transformer.html#setURIResolver(javax.xml.transform.URIResolver)). The URIResolver.resolve method takes two String arguments, the URI found in the stylesheet instructions or built as part of the transformation process, and the base URI against which the first argument will be made absolute if the absolute URI is required. The returned [Source](http://docs.google.com/javax/xml/transform/Source.html) object must be usable by the transformer, as specified in its implemented features.

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/xml/stream/util/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/xml/transform/dom/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/transform/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).