| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NamespaceSupport.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/org/xml/sax/helpers/LocatorImpl.html)   [**NEXT CLASS**](http://docs.google.com/org/xml/sax/helpers/ParserAdapter.html) | [**FRAMES**](http://docs.google.com/index.html?org/xml/sax/helpers/NamespaceSupport.html)    [**NO FRAMES**](http://docs.google.com/NamespaceSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **org.xml.sax.helpers**

Class NamespaceSupport

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **org.xml.sax.helpers.NamespaceSupport**

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

Encapsulate Namespace logic for use by applications using SAX, or internally by SAX drivers.

*This module, both source code and documentation, is in the Public Domain, and comes with* ***NO WARRANTY****.* See <http://www.saxproject.org> for further information.

This class encapsulates the logic of Namespace processing: it tracks the declarations currently in force for each context and automatically processes qualified XML names into their Namespace parts; it can also be used in reverse for generating XML qnames from Namespaces.

Namespace support objects are reusable, but the reset method must be invoked between each session.

Here is a simple session:

String parts[] = new String[3];  
 NamespaceSupport support = new NamespaceSupport();  
  
 support.pushContext();  
 support.declarePrefix("", "http://www.w3.org/1999/xhtml");  
 support.declarePrefix("dc", "http://www.purl.org/dc#");  
  
 parts = support.processName("p", parts, false);  
 System.out.println("Namespace URI: " + parts[0]);  
 System.out.println("Local name: " + parts[1]);  
 System.out.println("Raw name: " + parts[2]);  
  
 parts = support.processName("dc:title", parts, false);  
 System.out.println("Namespace URI: " + parts[0]);  
 System.out.println("Local name: " + parts[1]);  
 System.out.println("Raw name: " + parts[2]);  
  
 support.popContext();

Note that this class is optimized for the use case where most elements do not contain Namespace declarations: if the same prefix/URI mapping is repeated for each context (for example), this class will be somewhat less efficient.

Although SAX drivers (parsers) may choose to use this class to implement namespace handling, they are not required to do so. Applications must track namespace information themselves if they want to use namespace information.

**Since:** SAX 2.0

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**NSDECL**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#NSDECL)            The namespace declaration URI as a constant. |
| static [String](http://docs.google.com/java/lang/String.html) | [**XMLNS**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#XMLNS)            The XML Namespace URI as a constant. |

| **Constructor Summary** | |
| --- | --- |
| [**NamespaceSupport**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#NamespaceSupport())()            Create a new Namespace support object. |

| **Method Summary** | |
| --- | --- |
| boolean | [**declarePrefix**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#declarePrefix(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)            Declare a Namespace prefix. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html) | [**getDeclaredPrefixes**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getDeclaredPrefixes())()            Return an enumeration of all prefixes declared in this context. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPrefix**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefix(java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri)            Return one of the prefixes mapped to a Namespace URI. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html) | [**getPrefixes**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefixes())()            Return an enumeration of all prefixes whose declarations are active in the current context. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html) | [**getPrefixes**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefixes(java.lang.String))([String](http://docs.google.com/java/lang/String.html) uri)            Return an enumeration of all prefixes for a given URI whose declarations are active in the current context. |
| [String](http://docs.google.com/java/lang/String.html) | [**getURI**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String))([String](http://docs.google.com/java/lang/String.html) prefix)            Look up a prefix and get the currently-mapped Namespace URI. |
| boolean | [**isNamespaceDeclUris**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#isNamespaceDeclUris())()            Returns true if namespace declaration attributes are placed into a namespace. |
| void | [**popContext**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#popContext())()            Revert to the previous Namespace context. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**processName**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#processName(java.lang.String,%20java.lang.String%5B%5D,%20boolean))([String](http://docs.google.com/java/lang/String.html) qName, [String](http://docs.google.com/java/lang/String.html)[] parts, boolean isAttribute)            Process a raw XML qualified name, after all declarations in the current context have been handled by [declarePrefix()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#declarePrefix(java.lang.String,%20java.lang.String)). |
| void | [**pushContext**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#pushContext())()            Start a new Namespace context. |
| void | [**reset**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#reset())()            Reset this Namespace support object for reuse. |
| void | [**setNamespaceDeclUris**](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#setNamespaceDeclUris(boolean))(boolean value)            Controls whether namespace declaration attributes are placed into the [NSDECL](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#NSDECL) namespace by [processName()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#processName(java.lang.String,%20java.lang.String%5B%5D,%20boolean)). |

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

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

### XMLNS

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

The XML Namespace URI as a constant. The value is http://www.w3.org/XML/1998/namespace as defined in the "Namespaces in XML" \* recommendation.

This is the Namespace URI that is automatically mapped to the "xml" prefix.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#org.xml.sax.helpers.NamespaceSupport.XMLNS)

### NSDECL

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

The namespace declaration URI as a constant. The value is http://www.w3.org/xmlns/2000/, as defined in a backwards-incompatible erratum to the "Namespaces in XML" recommendation. Because that erratum postdated SAX2, SAX2 defaults to the original recommendation, and does not normally use this URI.

This is the Namespace URI that is optionally applied to *xmlns* and *xmlns:\** attributes, which are used to declare namespaces.

**Since:** SAX 2.1alpha **See Also:**[setNamespaceDeclUris(boolean)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#setNamespaceDeclUris(boolean)), [isNamespaceDeclUris()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#isNamespaceDeclUris()), [Constant Field Values](http://docs.google.com/constant-values.html#org.xml.sax.helpers.NamespaceSupport.NSDECL)

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

### NamespaceSupport

public **NamespaceSupport**()

Create a new Namespace support object.

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

### reset

public void **reset**()

Reset this Namespace support object for reuse.

It is necessary to invoke this method before reusing the Namespace support object for a new session. If namespace declaration URIs are to be supported, that flag must also be set to a non-default value.

**See Also:**[setNamespaceDeclUris(boolean)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#setNamespaceDeclUris(boolean))

### pushContext

public void **pushContext**()

Start a new Namespace context. The new context will automatically inherit the declarations of its parent context, but it will also keep track of which declarations were made within this context.

Event callback code should start a new context once per element. This means being ready to call this in either of two places. For elements that don't include namespace declarations, the *ContentHandler.startElement()* callback is the right place. For elements with such a declaration, it'd done in the first *ContentHandler.startPrefixMapping()* callback. A boolean flag can be used to track whether a context has been started yet. When either of those methods is called, it checks the flag to see if a new context needs to be started. If so, it starts the context and sets the flag. After *ContentHandler.startElement()* does that, it always clears the flag.

Normally, SAX drivers would push a new context at the beginning of each XML element. Then they perform a first pass over the attributes to process all namespace declarations, making *ContentHandler.startPrefixMapping()* callbacks. Then a second pass is made, to determine the namespace-qualified names for all attributes and for the element name. Finally all the information for the *ContentHandler.startElement()* callback is available, so it can then be made.

The Namespace support object always starts with a base context already in force: in this context, only the "xml" prefix is declared.

**See Also:**[ContentHandler](http://docs.google.com/org/xml/sax/ContentHandler.html), [popContext()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#popContext())

### popContext

public void **popContext**()

Revert to the previous Namespace context.

Normally, you should pop the context at the end of each XML element. After popping the context, all Namespace prefix mappings that were previously in force are restored.

You must not attempt to declare additional Namespace prefixes after popping a context, unless you push another context first.

**See Also:**[pushContext()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#pushContext())

### declarePrefix

public boolean **declarePrefix**([String](http://docs.google.com/java/lang/String.html) prefix,  
 [String](http://docs.google.com/java/lang/String.html) uri)

Declare a Namespace prefix. All prefixes must be declared before they are referenced. For example, a SAX driver (parser) would scan an element's attributes in two passes: first for namespace declarations, then a second pass using [processName()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#processName(java.lang.String,%20java.lang.String%5B%5D,%20boolean)) to interpret prefixes against (potentially redefined) prefixes.

This method declares a prefix in the current Namespace context; the prefix will remain in force until this context is popped, unless it is shadowed in a descendant context.

To declare the default element Namespace, use the empty string as the prefix.

Note that there is an asymmetry in this library: [getPrefix](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefix(java.lang.String)) will not return the "" prefix, even if you have declared a default element namespace. To check for a default namespace, you have to look it up explicitly using [getURI](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String)). This asymmetry exists to make it easier to look up prefixes for attribute names, where the default prefix is not allowed.

**Parameters:**prefix - The prefix to declare, or the empty string to indicate the default element namespace. This may never have the value "xml" or "xmlns".uri - The Namespace URI to associate with the prefix. **Returns:**true if the prefix was legal, false otherwise**See Also:**[processName(java.lang.String, java.lang.String[], boolean)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#processName(java.lang.String,%20java.lang.String%5B%5D,%20boolean)), [getURI(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String)), [getPrefix(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefix(java.lang.String))

### processName

public [String](http://docs.google.com/java/lang/String.html)[] **processName**([String](http://docs.google.com/java/lang/String.html) qName,  
 [String](http://docs.google.com/java/lang/String.html)[] parts,  
 boolean isAttribute)

Process a raw XML qualified name, after all declarations in the current context have been handled by [declarePrefix()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#declarePrefix(java.lang.String,%20java.lang.String)).

This method processes a raw XML qualified name in the current context by removing the prefix and looking it up among the prefixes currently declared. The return value will be the array supplied by the caller, filled in as follows:

parts[0] The Namespace URI, or an empty string if none is in use. parts[1] The local name (without prefix). parts[2] The original raw name.

All of the strings in the array will be internalized. If the raw name has a prefix that has not been declared, then the return value will be null.

Note that attribute names are processed differently than element names: an unprefixed element name will receive the default Namespace (if any), while an unprefixed attribute name will not.

**Parameters:**qName - The XML qualified name to be processed.parts - An array supplied by the caller, capable of holding at least three members.isAttribute - A flag indicating whether this is an attribute name (true) or an element name (false). **Returns:**The supplied array holding three internalized strings representing the Namespace URI (or empty string), the local name, and the XML qualified name; or null if there is an undeclared prefix.**See Also:**[declarePrefix(java.lang.String, java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#declarePrefix(java.lang.String,%20java.lang.String)), [String.intern()](http://docs.google.com/java/lang/String.html#intern())

### getURI

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

Look up a prefix and get the currently-mapped Namespace URI.

This method looks up the prefix in the current context. Use the empty string ("") for the default Namespace.

**Parameters:**prefix - The prefix to look up. **Returns:**The associated Namespace URI, or null if the prefix is undeclared in this context.**See Also:**[getPrefix(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefix(java.lang.String)), [getPrefixes()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefixes())

### getPrefixes

public [Enumeration](http://docs.google.com/java/util/Enumeration.html) **getPrefixes**()

Return an enumeration of all prefixes whose declarations are active in the current context. This includes declarations from parent contexts that have not been overridden.

**Note:** if there is a default prefix, it will not be returned in this enumeration; check for the default prefix using the [getURI](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String)) with an argument of "".

**Returns:**An enumeration of prefixes (never empty).**See Also:**[getDeclaredPrefixes()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getDeclaredPrefixes()), [getURI(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String))

### getPrefix

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

Return one of the prefixes mapped to a Namespace URI.

If more than one prefix is currently mapped to the same URI, this method will make an arbitrary selection; if you want all of the prefixes, use the [getPrefixes()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefixes()) method instead.

**Note:** this will never return the empty (default) prefix; to check for a default prefix, use the [getURI](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String)) method with an argument of "".

**Parameters:**uri - the namespace URI **Returns:**one of the prefixes currently mapped to the URI supplied, or null if none is mapped or if the URI is assigned to the default namespace**See Also:**[getPrefixes(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefixes(java.lang.String)), [getURI(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String))

### getPrefixes

public [Enumeration](http://docs.google.com/java/util/Enumeration.html) **getPrefixes**([String](http://docs.google.com/java/lang/String.html) uri)

Return an enumeration of all prefixes for a given URI whose declarations are active in the current context. This includes declarations from parent contexts that have not been overridden.

This method returns prefixes mapped to a specific Namespace URI. The xml: prefix will be included. If you want only one prefix that's mapped to the Namespace URI, and you don't care which one you get, use the [getPrefix](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefix(java.lang.String)) method instead.

**Note:** the empty (default) prefix is *never* included in this enumeration; to check for the presence of a default Namespace, use the [getURI](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String)) method with an argument of "".

**Parameters:**uri - The Namespace URI. **Returns:**An enumeration of prefixes (never empty).**See Also:**[getPrefix(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefix(java.lang.String)), [getDeclaredPrefixes()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getDeclaredPrefixes()), [getURI(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String))

### getDeclaredPrefixes

public [Enumeration](http://docs.google.com/java/util/Enumeration.html) **getDeclaredPrefixes**()

Return an enumeration of all prefixes declared in this context.

The empty (default) prefix will be included in this enumeration; note that this behaviour differs from that of [getPrefix(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefix(java.lang.String)) and [getPrefixes()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefixes()).

**Returns:**An enumeration of all prefixes declared in this context.**See Also:**[getPrefixes()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getPrefixes()), [getURI(java.lang.String)](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#getURI(java.lang.String))

### setNamespaceDeclUris

public void **setNamespaceDeclUris**(boolean value)

Controls whether namespace declaration attributes are placed into the [NSDECL](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#NSDECL) namespace by [processName()](http://docs.google.com/org/xml/sax/helpers/NamespaceSupport.html#processName(java.lang.String,%20java.lang.String%5B%5D,%20boolean)). This may only be changed before any contexts have been pushed.

**Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - when attempting to set this after any context has been pushed.**Since:** SAX 2.1alpha

### isNamespaceDeclUris

public boolean **isNamespaceDeclUris**()

Returns true if namespace declaration attributes are placed into a namespace. This behavior is not the default.

**Since:** SAX 2.1alpha

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NamespaceSupport.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/org/xml/sax/helpers/LocatorImpl.html)   [**NEXT CLASS**](http://docs.google.com/org/xml/sax/helpers/ParserAdapter.html) | [**FRAMES**](http://docs.google.com/index.html?org/xml/sax/helpers/NamespaceSupport.html)    [**NO FRAMES**](http://docs.google.com/NamespaceSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).