

XML

Namespaces

Namespaces are a mechanism by which element and attribute names can be assigned to groups. They are most often used when combining different vocabularies in the same document.

The following example shows that the part-catalog element contains two namespaces which are declared by the attributes `xmlns:nw` and `xmlns`. The elements inside part-catalog and their attributes belong to one or other namespace. Those in the first namespace can be identified by the prefix `nw`.

XML

```
<?xml version="1.0"?>
- <part-catalog xmlns="http://www.bobco.com/" xmlns:nw="http://www.nutware.com/">
  - <nw:entry nw:number="1327">
    <nw:description>torque-balancing hexnut</nw:description>
  </nw:entry>
  - <part id="555">
    <name>type 4 wingnut</name>
  </part>
</part-catalog>
```

The attributes of part-catalog are called namespace declarations. The general form of a namespace declaration is to start with the keyword `xmlns:` followed by the namespace prefix, and equals sign, and a namespace identifier in quotes.

XML

In a special form of the declaration, the colon and namespace prefix are left out, creating an implicit (unnamed) namespace. The first namespace declaration in the previous example is an implicit namespace.

part-catalog and any of its descendants without the namespace prefix `nw:` belong to the implicit namespace.

To include an element or attribute in a namespace other than the implicit namespace, you must do so as follows:

XML

Fully qualified name:

To the left of the colon is the namespace prefix, and to the right of the colon is the local name.

namespace prefix : local name

<nw:description>

Namespaces only affect a limited area in the document. The element containing the declaration and all of its descendants are in the scope of the namespace.

XML

Namespace identifiers are assigned a URL. This is not a requirement, however. The XML parser does not lookup any information located at that site. The site may not even exist.

So why use the URL?

The namespace must have a unique identifier. URLs are unique. They often contain information about the company or organisation so it makes a good candidate.

Still many have made the point that URLs are not really meant to be used as identifiers. Resources are often moved and URLs change. But since there is currently no alternative it looks like the practice is here to stay.

XML Namespaces

XML allows authors to create custom elements. This can result in naming collisions among elements in an XML document.

For example, we may use the element “part” to mark up data about computer parts, whereas “part” could also be used to mark up data about car parts. If these two files were combined in a product listing there would be a naming collision, making it difficult to determine which kind of data each element contains.

An XML namespace is a collection of element and attribute names. XML namespaces provide a means for authors to refer to elements with the same name.

XML Namespaces

This XML carries html table information

```
<table>  
  <tr>  
    <td>Apples</td>  
    <td>Bananas</td>  
  </tr>  
</table>
```

This XML carries furniture table information

```
<table>  
  <name>African Coffee Table</name>  
  <width>80</width>  
  <length>120</length>  
</table>
```

XML Namespaces

If these XML fragments were added together, there would be a name conflict. Both contain a `<table>` element, but the elements have different content and meaning.

A user or an XML application will not know how to handle these differences.

XML Namespaces

Name conflicts in XML can easily be avoided using a name prefix.

This XML carries information about an HTML table, and a piece of furniture:

```
<h:table>
```

```
  <h:tr>
```

```
    <h:td>Apples</h:td>
```

```
    <h:td>Bananas</h:td>
```

```
  </h:tr>
```

```
</h:table>
```

```
<f:table>
```

```
  <f:name>African Coffee Table</f:name>
```

```
  <f:width>80</f:width>
```

```
  <f:length>120</f:length>
```

```
</f:table>
```

XML Namespaces

<subject>Geometry</subject>

<subject>Cardiology</subject>

The first subject is something one studies in school, whereas the second the subject is a field of medicine.

A namespace can differentiate these two elements.

<second_level:subject>Geometry</second_level:subject>

<medical_school:subject>Cardiology</medical_school:subject>

XML Namespaces

Both `second_level:` and `medical_school:` are namespace prefixes. These specify the namespace each element belongs to.

Document authors create their own namespace prefixes using virtually any name. However you must ensure your namespace is unique.

XML Namespaces

When using prefixes in XML, a so-called **namespace** for the prefix must be defined.

The namespace is defined by the **xmlns attribute** in the start tag of an element.

Syntax:

`xmlns:prefix="URI"`

XML Namespaces

- In the example above, the xmlns attribute in the <table> tag give the h: and f: prefixes a qualified namespace.
- When a namespace is defined for an element, all child elements with the same prefix are associated with the same namespace.
- Namespaces can be declared in the elements where they are used or in the XML root element:

XML Namespaces

- `<catalog xmlns:h="http://www.w3.org/TR/html4/"
xmlns:f="http://www.w3schools.com/furniture">`

```
<h:table>
```

```
<h:tr>
```

```
<h:td>Apples</h:td>
```

```
<h:td>Bananas</h:td>
```

```
</h:tr>
```

```
</h:table>
```

```
<f:table>
```

```
<f:name>African Coffee Table</f:name>
```

```
<f:width>80</f:width>
```

```
<f:length>120</f:length>
```

```
</f:table>
```

```
</root>
```

XML Namespaces

A common method is to use a URL, which specifies the location of a file or a resource on the Internet.

```
xmlns:h="http://www.w3.org/TR/html4"
```

The parser does not visit the URL, nor do these URLs need to refer to actual web pages. They each simply represent a unique series of characters used to differentiate URI names.

XML Namespaces

Any string could represent a namespace.

```
xmlns:text = "IJYYHE998eKHGH"
```

When you then use the prefix in an element you must match the opening and closing tags, the same as you do in any xml document.

```
<text:image filename="happy.jpg"  
  <text:description>Happy Face</text:description>  
</text:image>
```


XML Namespaces

Attributes do not require namespace prefixes, because each attribute is already part of an element that specifies the namespace prefix.

Specifying a default namespace:

To eliminate the need to place namespace prefixes in each element, document authors may specify a default namespace for an element and its children.

The xmlns definition for the default namespace does not include a prefix but does have a URI, once this is defined that namespace is used for all elements unless another prefix is used.

XML Namespaces

```
<catalog xmlns="http://www.w3.org/TR/html4/"
xmlns:f="http://www.w3schools.com/furniture">
  <table>
    <tr>
      <td>Apples</td>
      <td>Bananas</td>
    </tr>
  </table>
  <f:table>
    <f:name>African Coffee Table</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table></root>
```

XML Namespaces

The default namespace refers to the <catalog> element and all the child elements that are not qualified with a namespace prefix.

You have already seen namespaces being used in your code.

xsl:value-of

xsl:for-each

To refer to these xslt instructions we use the xsl namespace prefix. If we look at our stylesheet declaration it contains:

```
<xsl:stylesheet version="1.0"
```

```
xmlns:xsl="http://www.w3.org/1999/Transform">
```

XML uses in practice.

RSS stands for RDF (Resource Description Framework) Site Summary, also known as Rich Site Summary and Really Simple Syndication.

RSS is an XML format used to syndicate website content, this format makes the content available to readers.

The content is packaged into what is known as an RSS feed or RSS channel. To see this content, you subscribe to an RSS feed. Then you can view the RSS feed using an RSS reader or an RSS aggregator.

The RSS orange icon is now well known, showing up on most websites, and blog sites. RSS allows you to subscribe to news articles, calendar events, blog postings, podcasts and photostreams.



XML uses in practice.

RSS readers are freely available from Google and Yahoo!
The major web browsers now have RSS readers built in making it more convenient to subscribe and view RSS feeds.

RSS 2.0 specification can be found:

<http://cyber.law.harvard.edu/rss/rss.html>

In this way an xml file of the web content is generated and then can be interpreted and presented by an RSS reader.

XML uses in practice.

RSS makes it possible to distribute up to date web content from one web site to thousands of other web sites around the world. RSS allows fast browsing for news and updates in areas that you subscribe to and are thus interested in.

Without RSS, users will have to check your site daily for new updates. This may be too time-consuming for many users. With an RSS feed they can check your site faster using an RSS aggregator (a site or program that gathers and sorts out RSS feeds).

Since RSS data is small and fast-loading, it can easily be used with services like cell phones or PDA's.

XML uses in practice.

With RSS you get to choose your news.

Reduce unwanted information and spam.

Increase network traffic to your web site.

All RSS feeds are written in XML, so they must follow the same rules

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

```
<rss version="2.0">
```

```
<channel>
```

```
  <title>W3Schools Home Page</title>
```

```
  <link>http://www.w3schools.com</link>
```

```
  <description>Free web building tutorials</description>
```

```
  <item>
```

```
    <title>RSS Tutorial</title>
```

```
    <link>http://www.w3schools.com/rss</link>
```

```
    <description>New RSS tutorial on W3Schools</description>
```

```
  </item>
```

```
  <item>
```

```
    <title>XML Tutorial</title>
```

```
    <link>http://www.w3schools.com/xml</link>
```

```
    <description>New XML tutorial on W3Schools</description>
```

```
  </item>
```

```
</channel>
```

```
</rss>
```