

Remove Namespaces from XML Documents

Finnbarr P. Murphy

(fpm@fpmurphy.com)

A frequent requirement when transforming XML documents is to remove some or all of a document namespaces. XML namespaces provide a simple and straightforward way to distinguish element and attribute names used in XML documents by associating them with namespaces identified by URI references.

Consider the following example XML document:

```
<work-order-list>
  <wo:job xmlns:wo="http:hello" freshness:timestamp="2006-01-12" xmlns:freshness="http://freshness"
    history:timestamp="2006" xmlns:history="http:history" xmlns:mnr="http:mnr">
    <wo:work-order id="1" status="k" freshness:timestamp="2006-01-13" />
  </wo:job>
  <wo:job xmlns:wo="http:hello" freshness:timestamp="2006-01-13" xmlns:freshness="http://freshness"
    history:timestamp="2006" xmlns:history="http:history" xmlns:mnr="http:mnr">
    <wo:work-order id="2" status="k" freshness:timestamp="2006-01-14" />
  </wo:job>
  <wo:job xmlns:wo="http:hello" freshness:timestamp="2006-01-14" xmlns:freshness="http://freshness"
    history:timestamp="2006" xmlns:history="http:history" xmlns:mnr="http:mnr">
    <wo:work-order id="3" status="k" freshness:timestamp="2006-01-15" />
  </wo:job>
</work-order-list>
```

It has several namespaces in it including *wo*, *freshness*, *history* and *mnr*. Some of those namespaces are declared but not used.

The following stylesheet can be removed to remove all namespaces from a document. I did not write this particular stylesheet. It is available at [Dave Pawson's](#) website and elsewhere.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" indent="no"/>
  <xsl:template match="*/comment()|processing-instruction()">
    <xsl:copy>
      <!-- go process children (applies to root node only) -->
      <xsl:apply-templates/>
    </xsl:copy>
  </xsl:template>
  <xsl:template match="*">
    <xsl:element name="{local-name()}">
      <!-- go process attributes and children -->
      <xsl:apply-templates select="@*|node()"/>
    </xsl:element>
  </xsl:template>
  <xsl:template match="@*">
    <xsl:attribute name="{local-name()}">
      <xsl:value-of select="."/>
    </xsl:attribute>
  </xsl:template>
```

```

</xsl:attribute>
</xsl:template>
</xsl:stylesheet>

```

It is an extension of the standard XSL `match=@*|node()` identity template which is discussed [here](#) and on lots of other websites. The key concept is that an identity transformation copies a source document to the destination document without change. In our case we need to remove all namespaces but otherwise leave the document as is. The function `local-name()` returns the local part of the expanded name of the node. Thus the namespace prefixes get dropped. If the argument is empty or the node has no expanded-name, an empty string is returned.

Here is the output produced when the above stylesheet is used to transform the example document:

```

$ xsltproc example.xsl example.xml
<?xml version="1.0"?>
<work-order-list>
  <job timestamp="2006">
    <work-order id="1" status="k" timestamp="2006-01-13"/>
  </job>
  <job timestamp="2006">
    <work-order id="2" status="k" timestamp="2006-01-14"/>
  </job>
  <job timestamp="2006">
    <work-order id="3" status="k" timestamp="2006-01-15"/>
  </job>
</work-order-list>

```

Suppose that instead of removing all namespaces, we wanted to only remove certain namespaces. Using our example document, suppose we wished to remove all namespaces except *wo*, *freshness* and *history*. In this case we have to add namespace declarations for *wo*, *freshness* and *history* and add extra templates to handle these namespaces when used with both elements and attributes.

Here is a stylesheet which meet the requirement:

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:freshness="http://freshness"
  xmlns:history="http://history"
  xmlns:wo="http://hello">
  <xsl:output method="xml" indent="no"/>
  <xsl:template match="/*|comment()|processing-instruction()">
    <xsl:copy>
      <!-- go process children (applies to root node only) -->
      <xsl:apply-templates/>
    </xsl:copy>
  </xsl:template>
  <xsl:template match="*">
    <xsl:element name="{local-name()}">
      <!-- go process attributes and children -->
      <xsl:apply-templates select="@*|node()"/>
    </xsl:element>
  </xsl:template>
  <xsl:template match="wo:*">
    <xsl:element name="{name()}" namespace="{namespace-uri()}">
      <!-- go process attributes and children -->
      <xsl:apply-templates select="@*|node()"/>
    </xsl:element>
  </xsl:template>

```

```
<xsl:template match="@history:*">
  <xsl:attribute name="{name()}" namespace="{namespace-uri()}">
    <xsl:value-of select="."/>
  </xsl:attribute>
</xsl:template>
<xsl:template match="@freshness:*">
  <xsl:attribute name="{name()}" namespace="{namespace-uri()}">
    <xsl:value-of select="."/>
  </xsl:attribute>
</xsl:template>
<xsl:template match="@*">
  <xsl:attribute name="{local-name()}">
    <xsl:value-of select="."/>
  </xsl:attribute>
</xsl:template>
</xsl:stylesheet>
```

The required namespace prefixes and declarations are retained by means of the `{name()}" namespace="{namespace-uri()}"` construct.

Here is the output when this stylesheet is applied to our example document:

```
<?xml version="1.0"?>
<work-order-list>
  <wo:job xmlns:wo="http:hello" xmlns:freshness="http://freshness" xmlns:history="http:hi
story" freshness:timestamp="20
06-01-12" history:timestamp="2006">
    <wo:work-order id="1" status="k" freshness:timestamp="2006-01-13"/>
  </wo:job>
  <wo:job xmlns:wo="http:hello" xmlns:freshness="http://freshness" xmlns:history="http:hi
story" freshness:timestamp="20
06-01-13" history:timestamp="2006">
    <wo:work-order id="2" status="k" freshness:timestamp="2006-01-14"/>
  </wo:job>
  <wo:job xmlns:wo="http:hello" xmlns:freshness="http://freshness" xmlns:history="http:hi
story" freshness:timestamp="20
06-01-14" history:timestamp="2006">
    <wo:work-order id="3" status="k" freshness:timestamp="2006-01-15"/>
  </wo:job>
</work-order-list>
```

As you can see the specified namespaces are retained and the others are no longer present in the document.

Hope this post helps someone somewhere figure out how to selectively remove namespaces using a stylesheet.