Tip: Computing word count in XML documents

Use XSLT and other tools to navigate around XML tags

Uche Ogbuji

September 29, 2005

XML is text and yet more than just text -- sometimes you want to work with just the content rather than the tags and other markup. In this tip, Uche Ogbuji demonstrates simple techniques for counting the words in XML content using XSLT with or without additional tools.

View more content in this series

One of the best things about XML is that it's just text, and you can use many general-purpose text processing tools to work with it. Occasionally, however, this doesn't work so well because the tags get in the way. As an author for developerWorks I have to submit my work in an XML template, and I usually want to get a word count for my efforts. The tags and other markup in the XML format don't count, so I need to figure out how to process this information from the content alone. In this tip, I show you how to use XSLT in conjunction with other tools -- or even by itself -- to solve this problem.

Just the text, ma'am

Listing 1 is an example of a well-formed XML document that's based on the draft of this article.

Listing 1. Sample XML document

```xml
  <!-- ARTICLE TITLE -->
  <title>Computing word count in XML documents</title>
  <docbody>
    <p>One of the best things about XML is that it's just text, and you can use many general-purpose text processing tools to work with it. Occasionally, however, this doesn't work so well because the tags get in the way.</p>
    <heading refname="" type="major" toc="yes" alttoc="">Just the text, ma'am</heading>
    <p>Listing 1 is an example of a well-formed XML document that's based on the draft of this article.</p>
  </docbody>
</dw-document>
```

If I run this XML file as it is through a tool such as UNIX's `wc` command, I get a word count of 82. This is rather high, though, as it includes the following sources of extraneous words:
• All the start and end elements
• The namespace declaration on the top element
• The comment within the top element
• The attribute names and values

The actual developerWorks templates include a lot more extraneous material, which can often double the apparent word count. However, the simplest possible XSLT 1.0 transform does something quite useful -- it strips all this extraneous material, leaving only the critical content. Listing 2 is a small variation on the minimal XSLT script:

Listing 2. Minimal XSLT that serves as a mark-up stripper (stripmarkup.xslt)

```xml
<xsl:transform
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   version="1.0">
   <xsl:output omit-xml-declaration="yes"/>
</xsl:transform>
```

The instruction `<xsl:output omit-xml-declaration="yes"/>` (the only instruction in the transform) ensures that the output does not include an XML declaration. If you apply this transform to Listing 1, you get the result in Listing 3.

Listing 3. Listing 1 stripped of markup using the transform in Listing 2

```
Computing word count in XML documents

One of the best things about XML is that it's just text, and you can use many general-purpose text processing tools to work with it. Occasionally, however, this doesn't work so well because the tags get in the way.

Just the text, ma'am

Listing 1 is an example of a well-formed XML document that's based on the draft of this article.
```

To check your word counts, use a command-line processor to pipe this stripped content directly to the UNIX `wc` command. Here's an example using 4Suite's XSLT processor (see Related topics):

```
$ 4xslt listing1.xml stripmarkup.xslt | wc
16 67 408
```

The middle number in the output, "67", is the word count (the other two numbers are the line and character counts, respectively). So the naïve count (82) was off by 15, or almost 25%. Again, this error can be much higher with some real-world document templates. If you're using an operating system without a built-in word count utility such as `wc`, you should still be able to find a third-party version, or you can just do the whole job in XSLT.

**Pure XSLT solution**

A pure XSLT word count solution might be handy if you don't want to use an external tool, or don't have one available. Perhaps you want to embed a word counting routine into a larger body of
XSLT. Such a routine is not possible in pure XSLT 1.0, although you can approximate the word count by counting the space characters after normalizing them. Using EXSLT -- the community standard in XSLT extensions -- does make this possible (see Related topics). Listing 4 is a pure XSLT and EXSLT word counting script.

**Listing 4. XSLT transform that computes the word count (wordcount.xslt)**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:str="http://exslt.org/strings">

    <xsl:variable name="ws" select="'\x20;\xA;\x9;\r;'"/>

    <xsl:template match="/">
        <xsl:text>XML content word count: </xsl:text>
        <xsl:value-of select="count(str:tokenize(string(.), $ws))"/>
    </xsl:template>

</xsl:transform>
```

You can see the declaration of the EXSLT string module, which uses the prefix str. The variable ws is just a string comprising all the standard whitespace characters. If you want to add some nuance, such as counting hyphenated words as two or more separate words, you could add the relevant characters to this string. The main action occurs in the one template where the str:tokenize function is used to split the text content into a node set of separate words. The number of token elements in this node set -- obtained using count -- is the desired word count.

```
$ 4xslt listing1.xml wordcount.xslt
<?xml version="1.0" encoding="UTF-8"?>
XML content word count: 67
```

**Wrap-up**

XML and the best XML technologies are very intelligent in the way they differentiate between elements, attributes, and text content. Most well-designed XML formats take advantage of this to make useful tasks such as content word counting a straightforward matter. Whenever you need to process some content that you have in XML form, you can use the basic techniques from this article to get started quickly.
Related topics

- Learn more about EXSLT, the community standard for useful and widely supported XSLT extension functions and elements. The string module has a number of functions, including str:tokenize.
- Make sure your XML design does not get in the way of normal content processing. See Uche's developerWorks series "Principles of XML design."
- Find more XML resources on the developerWorks XML zone, including a wide array of XML tips.
- Experiment with the code samples yourself. The stylesheet processor used in the examples is 4XSLT, part of 4Suite, which Uche Ogbuji co-develops.
- Find out how you can become an IBM Certified Developer.