

Re: Editing XML

Source: <http://coding.derkeiler.com/Archive/Delphi/comp.lang.pascal.delphi.misc/2007-06/msg00125.html>

- *From:* Hans-Peter Diettrich <DrDiettrich1@xxxxxxx>
 - *Date:* Tue, 12 Jun 2007 16:37:59 +0200
-

Rob Kennedy wrote:

I wonder how text and child nodes are distinguished in the XML text. Or can a element have either text or child nodes, but not both at the same time?

A text node is just another kind of child node. The "*" location path selects element children, and the "text()" location path selects text children. The "@*" location path selects all attributes of the context node, which must be an element node.

Thanks, I confused the text property, of the nodes, with the text children. There exist more confusing (redundant) properties, which I have to sort out...

Consider this XML document:

```
<?xml version="1.0"?>
<tag>Hello, world!</tag>
```

There are three nodes there. The first is the processing instruction, which you'll typically ignore in XSLT. The next is the element node named "tag" and the last is the text node. (There could really be more nodes -- you're allowed to have two consecutive text nodes -- but they usually get combined into a single node at some point.)

My problem was a mix of text and embedded nodes, quite a common structure in documents. When every piece of text is treated as a text node of it's own, everything fits together.

XSLT is XSLT. There's not much choice of another syntax. If Microsoft used anything else, it wouldn't be XSLT anymore.

Since Microsoft uses MSXML, why not also MSXSLT? ;-)

Re: Editing XML

There must exist a difference. When I start walking through the children of .Node, I get 3 xml nodes, but only 2 when starting with .DocumentElement. The last of these nodes seems to be the /xml tag, so I have no idea yet, what the Delphi classes or the MS interface understand as "node", "element" and so on :-(

I still don't understand why you're getting </xml> at all. It's not a node. It's the closing tag, which defines the end of the same element node that <xml> began.

Most probably it's the implementation of MSXML, where closing tags seem to translate into their own nodes, with the name and type of the opening tag, they only don't have child nodes and attributes. I came across this when counting the number of 'xml' nodes in the help files, in order to stop processing of the document after the xml element. Now that I know how to start processing with the document element, I can stop after processing the children of the first (and only) xml element. (see below)

In my example above, the element named "tag" is the document element. It is also the root element. There is always exactly one root element in a valid XML document.

I could figure out that the document root node is XMLDoc.Node, it's children are .ChildNodes, and their last element node is the ..DocumentElement. The ads in this case stem from Borland, they are not part of the interface.

There is also a document root, which in XPath is the parent of the root element. In the XML spec, I think it's the same as the "document entity." It serves as the parent for not only the root element but also any processing instructions and doctype declarations that appear outside the root element.

But note that the name "xml" is reserved. You're not supposed to use that as a name for an element or attribute unless some W3C standard defines it.

The HTML Help 2 documents start with

```
<?xml...?>  
<HTML ...>  
<HEAD>
```

and HEAD can contain an <xml>...</xml> part, containing keywords and other Help information, depending on the file type.

DoDi

Re: Editing XML

Re: Editing XML