Subject: Re: Another XML Question
Posted by Michael Wallace on Tue, 22 Mar 2005 22:07:23 GMT
View Forum Message <> Reply to Message

> OK, I'm lost again. :-(

You should have turned left at Alberquerque.

> I cannot figure out how to pull out the information from
> the CAMPAIGN_ID element for example:
>
>   name: CAMPAIGN_ID
>   value: 00
>   units: n/a
>
> Without knowing in my code that the element CAMPAIGN_ID exists.
>
> I guess my question is this: How can I find all the sub-elements
> of CONFIGDATA in a generic way? (I can find ALL the elements in the
> file, but what I want is the first sub-elements of CONFIGDATA, then
> all of *its* sub-elements, etc.) I find the documentation uh, vague. :-(

So, if I understand your question, you want to walk the node tree
without prior knowledge of what nodes you may encounter on your journey.
The only way I know to do this is with good ol' recursion. I could
explain the recursion, but an example is much better than an explanation
in this case. In the vague IDL documentation there's a pretty good
example of tree-walking. So, I guess the question is, what about this
doesn't do what you want it to?

From the IDL Documentation...

The following code traverses a DOM tree using pre-order traversal.

```
PRO sample_recurse, oNode, indent

   ; "Visit" the node by printing its name and value
   PRINT, indent GT 0 ? STRJOIN(REPLICATE(' ', indent)) : '', $
     oNode->GetNodeName(), ':', oNode->GetNodeValue()

   ; Visit children
   oSibling = oNode->GetFirstChild()
   WHILE OBJ_VALID(oSibling) DO BEGIN
     SAMPLE_RECURSE, oSibling, indent+3
     oSibling = oSibling->GetNextSibling()
   ENDWHILE
```

```
END

PRO sample
   oDoc = OBJ_NEW('IDLffXMLDOMDocument')
   oDoc->Load, FILENAME="sample.xml"
   SAMPLE_RECURSE, oDoc, 0
   OBJ_DESTROY, oDoc
END
```

-Mike

Subject: Re: Another XML Question
Posted by David Fanning on Tue, 22 Mar 2005 22:40:01 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> So, if I understand your question, you want to walk the node tree
> without prior knowledge of what nodes you may encounter on your journey.
>   The only way I know to do this is with good ol' recursion. I could
> explain the recursion, but an example is much better than an explanation
> in this case. In the vague IDL documentation there's a pretty good
> example of tree-walking. So, I guess the question is, what about this
> doesn't do what you want it to?

I think my problem is that my tree has a hell of a lot
of "children" that aren't what I expect them to be. They
seem to be text nodes with end-of-line strings or some
damn thing in them. I don't know where the hell they
come from. :-(

Once I realized this, and threw these objects out,
I had no difficulty traversing the tree and building
a set of structures and corresponding widgets to change
the values. Now I'm trying to figure out how to put
the values back in the XML file without having to completely
re-write the file.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Another XML Question
Posted by Michael Wallace on Wed, 23 Mar 2005 00:36:41 GMT
View Forum Message <> Reply to Message

> I think my problem is that my tree has a hell of a lot
> of "children" that aren't what I expect them to be. They
> seem to be text nodes with end-of-line strings or some
> damn thing in them. I don't know where the hell they
> come from. :-(

Was everything created and running under Windows? I've had end-of-line problems before when someone creates a text file on Windows and then ships it over to UNIX to be processed. Due to the end-of-line differences, all my lines on UNIX systems would end with ^M. Sometimes this causes a problem, sometimes not. Of course, if memory serves, you use Windows for just about everything, so it can't be that.

So, when you say there's an "end of line string" or something in there, what exactly are you talking about? I'm just curious what kind of gremlin or banshee you're dealing with.

> Once I realized this, and threw these objects out,
> I had no difficulty traversing the tree and building
> a set of structures and corresponding widgets to change
> the values. Now I'm trying to figure out how to put
> the values back in the XML file without having to completely
> re-write the file.
>
> Cheers,
>
> David
>

Subject: Re: Another XML Question
Posted by David Fanning on Wed, 23 Mar 2005 03:52:12 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> So, when you say there's an "end of line string" or something in there,
> what exactly are you talking about? I'm just curious what kind of
> gremlin or banshee you're dealing with.

Well, here is my XML file:

```
<CONFIGDATA>
 <CAMPAIGN_ID>
```

```
  <TYPE>INT</TYPE>
  <VALUE>00</VALUE>
 </CAMPAIGN_ID>
 <SPAM_WAIT>
  <TYPE>INT</TYPE>
  <VALUE>60</VALUE>
  <UNITS>Seconds</UNITS>
 </SPAM_WAIT>
</CONFIGDATA >
```

Here is my code:

```
   doc = Obj_New('IDLffXMLDOMDocument')
   doc -> Load, Filename=filename, /Exclude_Ignorable_Whitespace

   tags = doc -> GetElementsByTagName('CONFIGDATA')
   node = tags -> Item(0)
   children = node -> GetChildNodes()
    FOR j=0,children->GetLength()-1 DO BEGIN
      child = children -> Item(j)
      Help, child
   ENDFOR
```

And here is the result:

```
CHILD           OBJREF    = <ObjHeapVar43440(IDLFFXMLDOMTEXT)>
CHILD           OBJREF    = <ObjHeapVar43443(IDLFFXMLDOMELEMENT)>
CHILD           OBJREF    = <ObjHeapVar43445(IDLFFXMLDOMTEXT)>
CHILD           OBJREF    = <ObjHeapVar43447(IDLFFXMLDOMELEMENT)>
CHILD           OBJREF    = <ObjHeapVar43449(IDLFFXMLDOMTEXT)>
```

Only child 2 and 4 are the elements I'm looking for: CAMPAIGN_ID
and SPAM_WAIT. The other three children are some kind of white
space thingy. If I get the value of the TEXT objects, I see a single
quote on one line and a single quote on the next line. No text.

If I get the children of the CAMPAIGN_ID element, there are 9 of
them, and only 3 I care about.

Go figure!

Cheers,

David

--
David Fanning, Ph.D.

Subject: Re: Another XML Question
Posted by Karl Schultz on Wed, 23 Mar 2005 17:16:00 GMT
View Forum Message <> Reply to Message

On Tue, 22 Mar 2005 20:52:12 -0700, David Fanning wrote:

```
> Michael Wallace writes:
>
>> So, when you say there's an "end of line string" or something in there,
>> what exactly are you talking about?  I'm just curious what kind of
>> gremlin or banshee you're dealing with.
>
> Well, here is my XML file:
>
> <CONFIGDATA>
>  <CAMPAIGN_ID>
>   <TYPE>INT</TYPE>
>   <VALUE>00</VALUE>
>  </CAMPAIGN_ID>
>  <SPAM_WAIT>
>   <TYPE>INT</TYPE>
>   <VALUE>60</VALUE>
>   <UNITS>Seconds</UNITS>
>  </SPAM_WAIT>
> </CONFIGDATA >
>
> Here is my code:
>
>    doc = Obj_New('IDLffXMLDOMDocument')
>    doc -> Load, Filename=filename, /Exclude_Ignorable_Whitespace
>
>    tags = doc -> GetElementsByTagName('CONFIGDATA')
>    node = tags -> Item(0)
>    children = node -> GetChildNodes()
>     FOR j=0,children->GetLength()-1 DO BEGIN
>       child = children -> Item(j)
>       Help, child
>    ENDFOR
>
> And here is the result:
>
> CHILD          OBJREF    = <ObjHeapVar43440(IDLFFXMLDOMTEXT)>
> CHILD          OBJREF    = <ObjHeapVar43443(IDLFFXMLDOMELEMENT)>
> CHILD          OBJREF    = <ObjHeapVar43445(IDLFFXMLDOMTEXT)>
>  CHILD          OBJREF    = <ObjHeapVar43447(IDLFFXMLDOMELEMENT)>
>  CHILD          OBJREF    = <ObjHeapVar43449(IDLFFXMLDOMTEXT)>
```
>
> Only child 2 and 4 are the elements I'm looking for: CAMPAIGN_ID
> and SPAM_WAIT. The other three children are some kind of white
> space thingy. If I get the value of the TEXT objects, I see a single
> quote on one line and a single quote on the next line. No text.
>
> If I get the children of the CAMPAIGN_ID element, there are 9 of
> them, and only 3 I care about.
>
> Go figure!
>
> Cheers,
>
> David

In XML, whitespace is often considered significant, even in places where
you think it may not be.

For example,

```
<CAMPAIGN_ID>
 <TYPE>INT</TYPE>
 <VALUE>00</VALUE>
</CAMPAIGN_ID>
```

contains significant whitespace between the CAMPAIGN_ID start and end
tags. There are three newlines that correspond to the text nodes you
discovered above.

From an XML point of view, the above is QUITE different from:

```
<CAMPAIGN_ID><TYPE>INT</TYPE><VALUE>00</VALUE></CAMPAIGN_ID >
```

In this case, those three text nodes would not be in the DOM tree.

The XML folks wanted the whitespace to be detectable by the parser in case
there was an application need for that sort of information.

In order to teach the XML parser which whitespace is ignoreable and which
is not, you need to create a DTD or schema and specify the
EXCLUDE_IGNORABLE_WHITESPACE keyword. It is NOT sufficient to specify the
keyword without the DTD. (And the docs are *not* vague here :-).)

If you do make the DTD, you will not see those TEXT nodes containing
newlines or linefeeds. And the Windows vs Unix line terminator discussion
has nothing to do with this. You'd get the same result on either platform

and with either line terminator scheme. And that's by design.

If you don't want to make a DTD (it is worth doing, IMHO), you'd have to
beef up your parser to tolerate and skip over text nodes containing only
whitespace. Keep in mind too that an input file may contain:

<TYPE>INT</TYPE>

or

<TYPE>

INT

</TYPE>

and so you'd have to also deal with whitespace within an element where you
might expect none.

If you make a DTD, your parser becomes MUCH simpler.

The IDL docs won't tell you how to make a DTD. For that, and for most
other background XML expertise, you'll have to consult XML books, etc.

Karl

Subject: Re: Another XML Question
Posted by David Fanning on Wed, 23 Mar 2005 17:37:47 GMT
View Forum Message <> Reply to Message

Karl Schultz writes:

> In order to teach the XML parser which whitespace is ignoreable and which
> is not, you need to create a DTD or schema and specify the
> EXCLUDE_IGNORABLE_WHITESPACE keyword. It is NOT sufficient to specify the
> keyword without the DTD. (And the docs are *not* vague here :-).)

Ah, well, I guess it *is* clear if you know what the
hell a "DTD or schema" means. I find reading this XML
documentation similar to reading the German newspapers
last fall in Munich. I understand every third or
fourth word. But I'm improving! :-)

> The IDL docs won't tell you how to make a DTD. For that, and for most
> other background XML expertise, you'll have to consult XML books, etc.

Well, naturally. But who has the time for that. I have

deadlines and it's the client who is giving me these damn
XML files. Let *him* go read up on it! :-(

Thanks for the help. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Another XML Question
Posted by Michael Wallace on Wed, 23 Mar 2005 17:49:20 GMT
View Forum Message <> Reply to Message

> Ah, well, I guess it *is* clear if you know what the
> hell a "DTD or schema" means. I find reading this XML
> documentation similar to reading the German newspapers
> last fall in Munich. I understand every third or
> fourth word. But I'm improving! :-)

No wonder you were having problems. The actual XML file is only one
part of a larger puzzle -- 9 times out of 10 you'll need the DTD. Once
you have a DTD, parsing the file and working with it in general become
much easier.

-Mike

Subject: Re: Another XML Question
Posted by David Fanning on Wed, 23 Mar 2005 18:30:09 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> No wonder you were having problems. The actual XML file is only one
> part of a larger puzzle -- 9 times out of 10 you'll need the DTD. Once
> you have a DTD, parsing the file and working with it in general become
> much easier.

Oh, of course. Now what the hell is a DTD again?

Cheers,

David

P.S. Nearly all my life (in Colorado and Arizona) I've been
firmly in the minority. No reason to change now, I guess,
just to make life easier. :-)

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Another XML Question
Posted by jargoogle on Wed, 23 Mar 2005 19:43:08 GMT
View Forum Message <> Reply to Message

I happen to be learning this xml stuff right now too. For what it's
worth, I've so far avoided climbing the DTD/schema learning curve by
saving my XMLDomDocument without the pretty_print keyword set. I'll
tackle that mountain another day.

John.

Subject: Re: Another XML Question
Posted by Robert Barnett on Wed, 23 Mar 2005 22:54:46 GMT
View Forum Message <> Reply to Message

Hello there,

I know this might sound callous, but there are some who argue that you
haven't actually produced a valid XML document until you have specified
a document type definition (DTD) or similar. DTD and XSLs are common
ways of expressing the contraints on an XML document. You may have
realised from the barrage of acronyms that there are actually quite a
few alternatives to the document type definition (DTD). However, if your
documents are simple enough then a DTD should be more than sufficent.

I learnt XML and DTDs from a few excellent tutorials on the web. Either
that, or you could try getting a book on the subject. I can understand
that you might be tempted to skip out on DTDs. I guess this is ok, but,
don't expect anyone else to use your XML if you do not provide some
kind of definition of the structure of the XML document.

Here are an example of how I serialise IDL structures into XML and back
again. This is a parital implementation of the XML Remote Procedure Call

standard (XML-RPC - www.xmlrpc.com). I have provided a DTD as well.

I wish to serialise the following structure

configdata = {CAMPAIGN_ID: 0, SPAM_WAIT: 60}

This is an excerpt from the resulting XML

```
<struct>
<member>
<name>CAMPAIGN_ID</name>
<value><i4>0</i4></value>
</member>
<member>
<name>SPAM_WAIT</name>
<value><i4>60</i4></value>
</member>
</struct>
```

This is an excerpt from the DTD

```
<!ELEMENT i4 (#PCDATA)>

<!ELEMENT value (i4)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT member (name,value)>

<!ELEMENT struct (member+)>
```

I have taken advantage of borrowing elements from the existing XML-RPC DTD. This means that my code will be interoperable with many other implementations of XML-RPC. For example, I could import this XML into python as a dictionary in only a few lines of code.

The structure of your XML depends upon what you are using it for. In your case it appears that you are only using XML as a convenient way to get settings and configuration data into your IDL application. An alternative to your current XML dilemma might be to formulate some kind of general document definition (DTD or not) which would cover most of the possible ways that settings or configuration could be expressed. Who knows? You may find an minor extention of the example above more than sufficent.

For those interested, I'm slowly working on an implementation of XMLRPC for IDL 6.0.

I hope this clears things up a little. Happy easter everyone.

--

nrb@
Robbie Barnett
imag
Research Assistant
wsahs
Nuclear Medicine & Ultrasound
nsw
Westmead Hospital
gov
Sydney Australia
au
+61 2 9845 7223

Subject: Re: Another XML Question
Posted by Michael Wallace on Thu, 24 Mar 2005 00:37:17 GMT
View Forum Message <> Reply to Message

> Oh, of course. Now what the hell is a DTD again?

Document Type Definition.

It all begins with SGML, Standard Generalized Markup Language, which is
what other markup languages (XML, HTML, XHTML, etc) are based on. SGML
defines the basics of markup and sets a few rules, but not many. XML
was created by the W3C to be a simplified version of SGML suitable for
sharing data over the Internet. There is nothing that limits XML to
Internet-only work -- it just happened this was the original audience,
but it has now grown into a number of different worlds. Anyway, XML
simplified SGML by adding some extra structure, but didn't limit the
possible tags someone could define.

Even though XML is simpler, it could still be really ugly to work with
because the tags used are arbitrary. It'd be difficult to try to work
with a file without some prior knowledge about acceptable tags and
attributes, as our friend David has discovered. To get around the
problems of arbitrary definitions, a DTD is used. A DTD simply defines
all of the acceptable tags, nesting rules, acceptable attributes for
given tags and so on. Having such a definition allows data to be
interpreted and transfered much easier as you don't have to guess at
what's a valid tag or attribute and don't have to guess about what comes
next. There are plenty of XML parsers which only need a valid DTD and

they can read any XML file corresponding to that DTD. It's much easier
than rolling your own each time.

And if you ever here Schema or XSD mentioned along with XML, that's a
reference to the successor of DTD, but DTDs are still in high use, so
it's worthwhile to know how they work. Plus, it's easier to learn DTDs
first and then pick up XSD. Anyway, I saw another message in this
thread with an example of a DTD, so I won't bother with throwing up
another one. Also, one of the best resources I've found for getting
started on stuff like this is the W3C's www.w3schools.com site.
There's a lot of good info on XML in all it's variations.

-Mike

Subject: Re: Another XML Question
Posted by David Fanning on Thu, 24 Mar 2005 01:45:06 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> Plus, it's easier to learn DTDs first and then pick up XSD.

I overheard someone in Starbucks this afternoon talking about an
STD. I guess that's another one you can pick up. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Another XML Question
Posted by David Fanning on Thu, 24 Mar 2005 02:02:54 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> Also, one of the best resources I've found for getting
> started on stuff like this is the W3C's www.w3schools.com site.
> There's a lot of good info on XML in all it's variations.

I hope I haven't come across as sounding down on XML. The
truth is, I think it is terrific. What gets me testy is
having to learn something new with the clock sounding
like a metronome while it ticks down the minutes to the
impending deadline. (Especially so when on the day *before*
the big demo the client wants a feature that requires
a re-write of the fundamental program design. Yikes!)

And while the object nature of the XML parser is wonderful
in principle, it can be enormously frustrating when you
are trying to understand what is going on. Isn't there
a simple way to look *inside* those suckers!? But I've
muddled my way though, and I'm learning a lot. I've already
thought up three or four more uses for XML. And I greatly
appreciate all the help.

Cheers,

David

P.S. This really is a great place for XML info. But
after sitting in front of a computer for hours on end,
I could really use a good book to take to bed. And
suggestions?

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Another XML Question
Posted by James Kuyper on Thu, 24 Mar 2005 03:44:03 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:
>> Oh, of course. Now what the hell is a DTD again?
>
> Document Type Definition.
> ...
> [Lots of good general-purpose information about DTDs]

David's questions got me curious, so I spent a lot of time this
afternoon wading through the IDL documentation for it's XML objects. At
the end of that survey, I still couldn't figure out how to use a DTD in
connection with those objects. There's lots of things that say "if a
DTD is provided", but I couldn't find anything that indicated how to
provide a DTD. Could someone give some actual IDL code for this?

Subject: Re: Another XML Question
Posted by Karl Schultz on Thu, 24 Mar 2005 16:19:39 GMT
View Forum Message <> Reply to Message

On Wed, 23 Mar 2005 19:44:03 -0800, kuyper wrote:

> Michael Wallace wrote:
>>> Oh, of course. Now what the hell is a DTD again?
>>
>> Document Type Definition.
>> ...
>> [Lots of good general-purpose information about DTDs]
>
> David's questions got me curious, so I spent a lot of time this
> afternoon wading through the IDL documentation for it's XML objects. At
> the end of that survey, I still couldn't figure out how to use a DTD in
> connection with those objects. There's lots of things that say "if a
> DTD is provided", but I couldn't find anything that indicated how to
> provide a DTD. Could someone give some actual IDL code for this?

It is actually rather automatic and you don't really need IDL code for it.

DTD's are actually "provided" by the XML doc itself referencing a DTD. If
it does so and the VALIDATION_MODE keyword to IDLffXMLDOMDocument::Init or
Load is set to 1, then the parser will validate the document using the
DTD, if one is available.

In the example below, note the DOCTYPE element. It refers to a file named
slideshow.dtd that contains the DTD, also shown below. The DTD info can
actually be in the XML file itself, within the DOCTYPE element, but it is
common to put the DTD in a different file for easier reuse.

When VALIDATION_MODE is 1, the parser will check the XML against the DTD
and throw a parse error if the XML does not conform to the DTD. This is a
HUGE deal when writing IDL code to parse the XML. If you know the XML doc
passed the validation step, you can make tons of safe assumptions in your
parser code and get to the job at hand. If you don't validate, your
parser needs to be robust enough to deal with what might be a totally
random XML file. Like, I could pass an XML file describing the channel
lineup for my PVR to David's app that is expecting some special type of
configuration data. It would be easier to let the parser throw a parse
error than to write IDL code to discover the mistake.

Hope this helps,
Karl

++++++

Here is the example :

```
<?xml version='1.0' encoding='us-ascii'?>

<!--  A SAMPLE set of slides  -->

<!DOCTYPE slideshow SYSTEM "slideshow.dtd">

<slideshow
    title="Sample Slide Show"
    date="Date of publication"
    author="Yours Truly"
    >

    <!-- PROCESSING INSTRUCTION -->
    <?my.presentation.Program QUERY="exec, tech, all"?>

    <!-- TITLE SLIDE -->
    <slide type="all">
       <title>Wake up to WonderWidgets!</title>
    </slide>

    <!-- OVERVIEW -->
    <slide type="all">
      <title>Overview</title>
      <item>Why WonderWidgets are great</item>
      <item/>
      <item>Who buys WonderWidgets</item>
    </slide>

    <slide type="exec">
      <title>Financial Forecast</title>
      <item>Market Size &lt; predicted!</item>
      <item>Anticipated Penetration</item>
      <item>Expected Revenues</item>
      <item>Profit Margin </item>
    </slide>

</slideshow>
```

The DTD (in file slideshow.dtd):

```
<?xml version='1.0' encoding='us-ascii'?>

<!--
    DTD for a simple "slide show".
-->
```

```
<!ELEMENT slideshow (slide+)>
<!ATTLIST slideshow
          title    CDATA    #REQUIRED
          date     CDATA    #IMPLIED
          author   CDATA    "unknown"
>
<!ELEMENT slide (image?, title, item*)>
<!ATTLIST slide
          type   (tech | exec | all) #IMPLIED
>
<!ELEMENT title (#PCDATA)>
<!ELEMENT item (#PCDATA | item)* >
<!ELEMENT image EMPTY>
<!ATTLIST image
          alt    CDATA    #IMPLIED
          src    CDATA    #REQUIRED
          type   CDATA    "image/gif"
>
```

Subject: Re: Another XML Question
Posted by Michael Wallace on Thu, 24 Mar 2005 22:56:09 GMT
View Forum Message <> Reply to Message

> P.S. This really is a great place for XML info. But
> after sitting in front of a computer for hours on end,
> I could really use a good book to take to bed. And
> suggestions?

Well, I don't have a suggestion for a specific XML book. The only thing
I can recommend is to stay away from the books that cover XML with
respect to the gazillion other related technologies, especially if
you're not going to do web development. The majority of XML books are
written from a web perspective -- the typical audience for the books are
web site developers and the like. Look for something where the majority
of the book is spent on XML itself, DTD, maybe some Schema. And maybe
where the latter chapters introduce you to XSL, XSLT, XPath, XHTML, etc.
I hate to say it, but you'll need to go to Amazon and start reading
book reviews to find the right XML book for the job.

-Mike
