Subject: loop for XML
Posted by skymaxwell@gmail.com on Sat, 06 Dec 2014 13:17:10 GMT
View Forum Message <> Reply to Message

Hi

I need read the XML file. Here is the fragment of file

```
<root>
   <Key1>
      <p1>13</p1>
      <tb1>10</tb1>
   </Key1>
   <KeyMeta>
      <ax>21</ax>
      <ay>59</ay>
      <az>26.7</az>
   </KeyMeta>
</root>
```

I was looking here something about XML, I find simple example with single tag
There used these functions (oValue->GetFirstChild())->GetNodeValue())

How build construction for multiple tags as my case ?
I didn't find any functions in IDL help, which i could use in my case.
May be I don't understand function Item(i).

Here part of code

```
==============
p=OBJ_NEW('IDLffXMLDOMDocument')
p->Load,FILENAME=filenameXML
oTopLevel=p->GetDocumentElement() ;return root IDLffXMLDOMDocument
; tag with name
mainTagsList=oTopLevel->GetElementsByTagName("KeyMeta")
; how many KeyMeta tags in documents
PRINT,"Length=",mainTagsList->GetLength()
;loop
FOR i=0,mainTagsList->GetLength()-1 DO BEGIN
 ;How to do correct loop ???
ENDFOR

==============
```

thanks

Subject: Re: loop for XML
Posted by Michael Galloy on Sat, 06 Dec 2014 18:57:30 GMT
View Forum Message <> Reply to Message

On 12/6/14, 6:17 am, skymaxwell wrote:
> Hi
>
> I need read the XML file. Here is the fragment of file
>
> <root>
> <Key1>
> <p1>13</p1>
> <tb1>10</tb1>
> </Key1>
> <KeyMeta>
> <ax>21</ax>
> <ay>59</ay>
> <az>26.7</az>
> </KeyMeta>
> </root>
>
>
> I was looking here something about XML, I find simple example with single tag
> There used these functions (oValue->GetFirstChild())->GetNodeValue())
>
>
> How build construction for multiple tags as my case ?
> I didn't find any functions in IDL help, which i could use in my case.
> May be I don't understand function Item(i).
>
> Here part of code
> ==========
> p=OBJ_NEW('IDLffXMLDOMDocument')
> p->Load,FILENAME=filenameXML
> oTopLevel=p->GetDocumentElement() ;return root IDLffXMLDOMDocument
> ; tag with name
> mainTagsList=oTopLevel->GetElementsByTagName("KeyMeta")
> ; how many KeyMeta tags in documents
> PRINT,"Length=",mainTagsList->GetLength()
> ;loop
> FOR i=0,mainTagsList->GetLength()-1 DO BEGIN
> ;How to do correct loop ???
> ENDFOR
>
> ==========
>
>
> thanks
>

Try this:

```
p = obj_new('IDLffXMLDOMDocument')
p->load, filename=filenameXML

oTopLevel = p->getDocumentElement() ;return root IDLffXMLDOMDocument

; tag with name
metaList = oTopLevel->getElementsByTagName('KeyMeta')
; how many KeyMeta tags in documents
metaElement = metaList->item(0)
metaChildrenList = metaElement->getElementsByTagname('*')
for c = 0L, metaChildrenList->getLength() - 1L do begin
   item = metaChildrenList->item(c)
   name = item->getTagname()
   dataList = item->getFirstChild()
   value = dataList->getNodeValue()
   print, name, value, format='(%"%s = %s")'
endfor
```

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: loop for XML
Posted by skymaxwell@gmail.com on Sun, 07 Dec 2014 20:35:32 GMT
View Forum Message <> Reply to Message

Thanks a lot ! it works!

Subject: Re: loop for XML
Posted by mick.mitanirc3 on Mon, 08 Dec 2014 16:41:30 GMT
View Forum Message <> Reply to Message

I have a question about XML object functions. I've not used XML object functions since until recently we only had IDL 5.5 so I tried the code from Mike on one of the weather data files we parse. It aborts when it reaches the first missing field so the tag is <urn:Value/>. Weather data is always missing one or more fields so the XML files are full of such tags. I had to write a lot of code to handle all of the possible ways missing data is marked. How do I handle this with object functions?

Subject: Re: loop for XML
Posted by Jim Pendleton on Tue, 09 Dec 2014 02:30:10 GMT
View Forum Message <> Reply to Message

On Monday, December 8, 2014 9:41:33 AM UTC-7, mick.mi...@gmail.com wrote:
> I have a question about XML object functions. I've not used XML object functions since until recently we only had IDL 5.5 so I tried the code from Mike on one of the weather data files we parse. It aborts when it reaches the first missing field so the tag is <urn:Value/>. Weather data is always missing one or more fields so the XML files are full of such tags. I had to write a lot of code to handle all of the possible ways missing data is marked. How do I handle this with object functions?

If you had a DTD file, that could solve some of your issues. The DTD would contain the rules for the proper syntax of the XML.

Another approach would be to use a node iterator and turn off schema checking.

```
pro untitled_2
p=OBJ_NEW('IDLffXMLDOMDocument')
p->Load,FILENAME=test.xml', Schema_Checking = 0
oTopLevel=p->GetDocumentElement() ;return root IDLffXMLDOMDocument
oIt = p->CreateNodeIterator(oTopLevel)
Node = oIt->NextNode()
Repeat Begin
  Case Obj_Class(Node) of
    'IDLFFXMLDOMTEXT' : Begin
      text = Node->GetData()
      if ((text.tobyte())[0] ne 10) then print, tn, ': ', text
      End
    'IDLFFXMLDOMELEMENT' : Begin
      tn = Node->GetTagName()
      if ((tn.tobyte())[0] ne 10) then print, 'tag = ', tn
      End
    Else:
  EndCase
  Node = oIt->NextNode()
EndRep Until (Node eq !null)
End
```

Inserting a new "<urn:Value/>" line into the previous example,

```
<root>
<Key1>
<urn:Value/>
<p1>13</p1>
<tb1>10</tb1>
</Key1>
<KeyMeta>
<ax>21</ax>
<ay>59</ay>
```

<az>26.7</az>
</KeyMeta>
</root>

Execution of the code now produces:
tag = root
tag = Key1
tag = urn:Value
tag = p1
p1: 13
tag = tb1
tb1: 10
tag = KeyMeta
tag = ax
ax: 21
tag = ay
ay: 59
tag = az
az: 26.7

Jim P.
"I still work for Exelis"

Subject: Re: loop for XML
Posted by mick.mitanirc3 on Tue, 09 Dec 2014 16:49:56 GMT
View Forum Message <> Reply to Message

Thanks, Jim. I'll have to examine that code to see what it does. I'm working with Joint METOC JMBL output and there is no DTD file available. I know what tags I will get from the database and I know how the database may provide closing tags for missing data, but there are at least three different ways and up to a dozen different places that they could appear. I have a serviceable parser solution I wrote in IDL but it isn't speedy by any measure. That is why I'm interested in the native XML objects. Nearly all of the code base in my office is written for IDL 5.5 or older. Whenever possible I've been rewriting code to use the IDL Way to improve performance. At the same time, as the only trained programmer in my section I've been trying to clean up and standardize the format of the code that my scientist coworkers have written over the years.
