

---

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
olt = p->CreateNodeIterator(oTopLevel)
Node = olt->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 = olt->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"

---