Subject: Re: XML Question
Posted by Robert Barnett on Wed, 23 Feb 2005 00:29:56 GMT
View Forum Message <> Reply to Message

Well DOM is a specification which covers many, *many* different languages. Data typing has to be very generic and most structural information is expressed in OO jargon.

Sometimes the easiest way serialise something as a DOM is to actually inherit the DOM::Element class itself. Read and write everything to the DOM rather than storing stuff in your own OO fields. This option is not really available to you because you have already written your code.

If you want to learn DOM and XML quickly I'd actually recommend looking at tutorials in another language that is familiar to you. Something like perl, python, java or c++ might be ideal.

Occasionally, you might find that XML SAX is a bit easier to use.

David Fanning wrote:

> Folks,
>
> I think I am beginning to understand how people feel when
> they start to learn about objects for the first time.
> That is to say, the IDL documentation is just about
> impossible.
>
> I'm trying to get up to speed on XML files very quickly
> (that is, I needed this yesterday). I'm looking at the
> documentation and my anxiety is increasing exponentially.
> Does anyone have a fairly simple example of reading an
> XML file and doing something useful with the information?
>
> I would like, for example, to read a configuration file
> with parameters and their associated values. I would like
> to build a widget that would allow the user to change the
> values, and finally I would like to write the configuration
> file back to an XML file. Does anyone have anything remotely
> like that?
>
> I see to be floundering in IDLffDOM minutiae. :-(
>
> Cheers,
>
> David
>

>

--

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

Subject: Re: XML Question
Posted by Michael Wallace on Wed, 23 Feb 2005 00:31:45 GMT
View Forum Message <> Reply to Message

> I'm trying to get up to speed on XML files very quickly
> (that is, I needed this yesterday). I'm looking at the
> documentation and my anxiety is increasing exponentially.

Uh oh.

> Does anyone have a fairly simple example of reading an
> XML file and doing something useful with the information?

No.

> I would like, for example, to read a configuration file
> with parameters and their associated values. I would like
> to build a widget that would allow the user to change the
> values, and finally I would like to write the configuration
> file back to an XML file. Does anyone have anything remotely
> like that?

I've worked with XML before, but not with IDL. I don't know how their classes are set up to handle things. I hate to say it, but I took a quick look at the documentation, and the "Using the XML DOM classes" looks like what you need -- there are examples, but in classic IDL fashion, they write the examples using a sample.xml file which they don't show you. So it's pretty hard to understand things without knowing the format of the file they're using. You might try loading a

very simple XML file and playing with the methods they use. That's about all I can recommend. I didn't even know before now that IDL even supported XML. I'm going to have to look harder at this in the future.

-Mike

Subject: Re: XML Question
Posted by Karl Schultz on Wed, 23 Feb 2005 17:23:54 GMT
View Forum Message <> Reply to Message

On Tue, 22 Feb 2005 16:25:53 -0700, David Fanning wrote:

> Folks,
>
> I think I am beginning to understand how people feel when
> they start to learn about objects for the first time.
> That is to say, the IDL documentation is just about
> impossible.
>
> I'm trying to get up to speed on XML files very quickly
> (that is, I needed this yesterday). I'm looking at the
> documentation and my anxiety is increasing exponentially.
> Does anyone have a fairly simple example of reading an
> XML file and doing something useful with the information?
>
> I would like, for example, to read a configuration file
> with parameters and their associated values. I would like
> to build a widget that would allow the user to change the
> values, and finally I would like to write the configuration
> file back to an XML file. Does anyone have anything remotely
> like that?
>
> I see to be floundering in IDLffDOM minutiae. :-(
>
> Cheers,
>
> David

XML DOM is pretty complicated, so it takes awhile to learn.

Rather than tackle the "write-back" part at the same time, let's just work on building the widget app from a config file.

Here's a sample config file. To really make it solid from an XML point of view, one needs to create a DTD or a schema to describe the data and enforce it during parsing. If you let the parser enforce the schema, you can greatly simplify your code that walks the DOM tree, since you don't

have to prepare for every possible input possibility. For now, we'll assume that the input xml file is "legal" and we'll make a lot of assumptions when parsing it.

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

<demo title="XMLDOM Sample">

 <widget type="slider" min="0" max="20">
  <value>13</value>
 </widget>

 <widget type="text">
  <value>Hello</value>
 </widget>
</demo>
```

I purposely made the "value" tags XML Elements instead of attributes on the "widget" Element because this would allow widget data to be specified in a more flexible way. For example, you can make put an "array" tag instead of a "value" tag that specifies the contents for a table widget.

Here's an IDL program that reads the above config file and builds a super-simple widget app.

```
pro demo
   oDoc = OBJ_NEW('IDLffXMLDOMDocument', FILENAME='demo.xml')
   oTopLevel = oDoc->GetDocumentElement()
   oWidgetList = oTopLevel->GetElementsByTagName("widget")

   wBase = WIDGET_BASE(/COL, TITLE=oTopLevel->GetAttribute("title"))

   for i=0, oWidgetList->GetLength()-1 do begin
      oWidget = oWidgetList->Item(i)
      case oWidget->GetAttribute("type") OF
      'slider': begin
         wSlider = WIDGET_SLIDER(wBase, $
                        MINIMUM=FIX(oWidget->GetAttribute("min")), $
                        MAXIMUM=FIX(oWidget->GetAttribute("max")))
         oValueList = oWidget->GetElementsByTagName("value")
         oValue = oValueList->Item(0)
         WIDGET_CONTROL, wSlider, $
            SET_VALUE=FIX((oValue->GetFirstChild())->GetNodeValue())
         end
      'text': begin
         wText = WIDGET_TEXT(wBase)
```

```
        oValueList = oWidget->GetElementsByTagName("value")
        oValue = oValueList->Item(0)
        WIDGET_CONTROL, wText, $
           SET_VALUE=(oValue->GetFirstChild())->GetNodeValue()
        end
      else: print, "Unknown widget type"
      endcase
   endfor

   OBJ_DESTROY, oDoc
   WIDGET_CONTROL, wBase, /REALIZE
end
```

Hope this helps. I can help further, if needed.

Karl

Subject: Re: XML Question
Posted by David Fanning on Wed, 23 Feb 2005 17:33:08 GMT
View Forum Message <> Reply to Message

Karl Schultz writes:

> Hope this helps. I can help further, if needed.

Oh, that *really* helps. Thanks!

I've been reading my w3schools.com tutorials, and working my
way through the relevant sections of Building Applications
and it is beginning to make a lot of sense to me now. In fact,
like most of the objects I work with, it generates more good
ideas than I know what to do with! I can see all *kinds* of
possibilities! :-)

Cheers,

David

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

Subject: Re: XML Question
Posted by Karl Schultz on Wed, 23 Feb 2005 18:03:54 GMT

On Wed, 23 Feb 2005 10:33:08 -0700, David Fanning wrote:

> Karl Schultz writes:
>
>> Hope this helps. I can help further, if needed.
>
> Oh, that *really* helps. Thanks!

Great!

> I've been reading my w3schools.com tutorials, and working my
> way through the relevant sections of Building Applications
> and it is beginning to make a lot of sense to me now. In fact,
> like most of the objects I work with, it generates more good
> ideas than I know what to do with! I can see all *kinds* of
> possibilities! :-)

That's exactly the thing to do. The IDL XMLDOM support is built on top of the Xerces DOM library, so after you trip across that, the IDL stuff will seem pretty familiar.

Here's an update to the program that lets you modify the widget values. As the widget values change, the DOM tree contents are modified. When the application is closed, the DOM tree is written back out to the XML file. When you start the app again, you see your updated values in the widgets.

This code is really bare-bones for newsgroup-level brevity with a lot of ugly shortcuts, but you'll get the idea.

Karl

```
pro demo_cleanup, w
   WIDGET_CONTROL, w, GET_UVALUE=state
   state.oDoc->Save, FILENAME='demo.xml'
   OBJ_DESTROY, state.oDoc
end

pro demo_event, ev
   WIDGET_CONTROL, ev.id, GET_VALUE=value, GET_UVALUE=oValue
   oValue->SetNodeValue, STRING(value[0])
end


pro demo
   oDoc = OBJ_NEW('IDLffXMLDOMDocument', FILENAME='demo.xml')
   oTopLevel = oDoc->GetDocumentElement()
```

```
   oWidgetList = oTopLevel->GetElementsByTagName("widget")

   wBase = WIDGET_BASE(/COL, TITLE=oTopLevel->GetAttribute("title"))

   for i=0, oWidgetList->GetLength()-1 do begin
      oWidget = oWidgetList->Item(i)
      case oWidget->GetAttribute("type") OF
      'slider': begin
         wSlider = WIDGET_SLIDER(wBase, $
                           MINIMUM=FIX(oWidget->GetAttribute("min")), $
                           MAXIMUM=FIX(oWidget->GetAttribute("max")))
         oValueList = oWidget->GetElementsByTagName("value")
         oValue = oValueList->Item(0)
         WIDGET_CONTROL, wSlider, $
            SET_VALUE=FIX((oValue->GetFirstChild())->GetNodeValue()), $
            SET_UVALUE=oValue->GetFirstChild()
         end
      'text': begin
         wText = WIDGET_TEXT(wBase, /EDITABLE)
         oValueList = oWidget->GetElementsByTagName("value")
         oValue = oValueList->Item(0)
         WIDGET_CONTROL, wText, $
            SET_VALUE=(oValue->GetFirstChild())->GetNodeValue(), $
            SET_UVALUE=oValue->GetFirstChild()
         end
       else: print, "Unknown widget type"
       endcase
   endfor

   state = { $
      oDoc: oDoc }

   WIDGET_CONTROL, wBase, SET_UVALUE=state
   WIDGET_CONTROL, wBase, /REALIZE

   XMANAGER, 'demo', wBase, CLEANUP='demo_cleanup', /NO_BLOCK
end
```