Subject: Re: Another XML Question
Posted by JD Smith on Thu, 24 Mar 2005 17:21:16 GMT
View Forum Message <> Reply to Message

On Thu, 24 Mar 2005 09:19:39 -0700, Karl Schultz wrote:

> 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.

I can't help but wonder here how much more quickly David would have
progressed if his input file had been a simple text file like:

file config.txt

```
====================
CAMPAIGN_ID: 0
SPAM_WAIT: 60
HEARTBEAT_TIME: 60
====================
```

I bet he could have had that file parsing in 10 seconds or less. XML is nice. Overusing XML to be buzzword compliant, not so nice.

JD

```
>  ++++++
>
>  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"
>>
```