

Re: What is the best way to parse and validate an arbitrary XML document

Source: <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.xml/2006-08/msg00211.html>

- *From:* "John Saunders" <john.saunders at trizetto.com>
 - *Date:* Fri, 11 Aug 2006 13:58:03 -0400
-

"Confused XML hacker" <Confused XML hacker@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:F7576F6D-6775-4319-A1F8-CD963532AD7B@xxxxxxxxxxxxxxxxxxxx

My application needs to be able to parse and validate either a DTD or schema based document without knowing in advance which form of grammar a document is using. (New documents presented to my system are schema based while the older ones are DTD – conversion is not an option as these document represent legally binding contracts and they must be processed as is).

In the .Net 1.1 version of my code I used a XmlValidatingReader instance configured with ValidationType.Auto which handled both document types. Now that I am porting the code to 2.0 I am trying to use the new XmlReader.Create method but these readers must be configured as either DTD validating or schema validating and can not do both at once.

I found a several examples on the web of nested XmlReaders and have tried this approach (see following code) but the data in the <!DOCTYPE> (my application needs the PUBLIC name intact to determine the grammar version) node of a DTD based instance seems to be lost as it passed through the readers.

```
XmlReaderSettings settings;

settings = new XmlReaderSettings ();
settings.XmlResolver = GetResolver ();
settings.ValidationType = ValidationType.DTD;
settings.ProhibitDtd = false;

if (eventHandler != null)
    settings.ValidationEventHandler += eventHandler;

XmlReader inner = XmlReader.Create (stream, settings);

settings = new XmlReaderSettings ();
settings.Schemas = GetSchemaSet ();
```

Re: What is the best way to parse and validate an arbitrary XML document

```
settings.ValidationType = ValidationType.Schema;  
settings.ProhibitDtd = false;
```

```
if (eventHandler != null)  
settings.ValidationEventHandler += eventHandler;
```

```
XmlReader reader = XmlReader.Create (inner, settings);
```

```
document.Load (reader);  
return (document);
```

Can someone please confirm if I am using the right approach and whether the problems with DOCTYPE are bugs, and if so is there a work around. If not then it looks like I will have to remain with the old and now 'obsolete' XmlValidatingReader interface which handles this scenario correctly.

The changes to the XML API in 2.0 seem to forget that there is a substantial amount of information coded using DTD based grammars that will be with us for many years. Some of the contracts I process represent 10 or 20 year deals. Not everyone using XML is writing webservice where the XML is simply a transient wire format. Efficient support for legacy documents and mixed DTD/schema environments is commercially very important.

Could you first attempt it using a schema, and if that fails, using DTD? Or should I read your post more carefully?

John

.