
Subject: Re: GENERIC XML READER

Posted by [Karl\[1\]](#) on Wed, 10 Dec 2008 21:31:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Dec 10, 9:34 am, Bennett <juggernaut...@gmail.com> wrote:

> On Dec 9, 7:43 am, bernat <bernat.puigdomen...@gmail.com> wrote:

>

>> Hello people,

>

>> I'm trying to write a generic xml reader. Since today I used to read

>> the xml using the elements name like:

>

>> oDesc = (oltem->GetElementsByTagName('description'))

>

>> Is it posible to create a generic xml reader? Anybody have any kinds

>> of examples or codes?

>

>> Thanks,

>> bernat

>

> What do you mean by generic? There are ways to just read out the

> entire xml file but I would think you would be looking for something

> specific to extract from an xml file in order to do anything useful

> with it. So do you mean a function that would search an xml file for

> a specific tag/text/attribute/etc....?

The original request is a bit vague. Usually, an application reads an XML file to extract very specific information. So a "generic" reader is a bit undefined.

If you simply want to read the XML and store the result in memory, that is exactly what the DOM model does for you. You would just walk the tree and create objects for each DOM element. As a result, you would then have read the entire file and have access to all the data via the objects.

Since XML files are just text, one could write a program to walk the DOM tree and simply "print out" every XML element. You'd have to decide what "print out" means for each XML element type, but I think that is possible. I imagine one could get a bit creative and generate a graphical visualization of the DOM tree by drawing a tree with a node for each Element. You could hang attributes off of each node that has attributes, etc. Another approach would be to dump the tree in a text format, indenting the text based on tree depth. It would be a challenge to handle both deep and wide trees well.

I mentioned "walking" the tree twice above. I'm certain that there are several examples in the IDL XMLDOM documentation that show you how

to write IDL code to walk the entire DOM tree. These routines would generically "read" the file for you without regard to context or content. It would then be up to you what to do with each node.

Karl
