Subject: Bug in the SAX parser?
Posted by mattf on Tue, 09 Jan 2007 14:34:58 GMT
View Forum Message <> Reply to Message

Just ran into a, uh, interesting problem with the SAX parser in IDL. At
one point in processing an XML file, I have to remove the ignorable
white space-- so I wrote a 'remove_white' subclass of the SAX parser to
do that (a good exercise, btw, for anyone who's learning about this
stuff). However, it turns out that the file I want to process has
two-byte characters-- the 'remove_white' parser isn't bothered by that,
but it appears to have a problem with recognizing a Windows newline
(i.e., linefeed + carriage return) in two-byte character form.
Specifically, it spots the carriage return, but not the linefeed-- so,
my output file still has all the linefeeds. This is easy enough to deal
with, in the subclassed 'Chars' method I test for

chars[0] ne string(10B)

before sending the string to the output file. But I think it's a bug.

Subject: Re: Bug in the SAX parser?
Posted by Karl Schultz on Wed, 10 Jan 2007 22:22:42 GMT
View Forum Message <> Reply to Message

On Wed, 10 Jan 2007 21:49:26 +0000, Matt Feinstein wrote:

> In article <09012007162923276%mfein@clark.net>, Matt Feinstein
> <mfein@clark.net> wrote:
>
>> In article <12q7hcegpr0gm6a@corp.supernews.com>, Karl Schultz
>> <k_remove_schultz@ittvis.com> wrote:
>>
>>> On Tue, 09 Jan 2007 06:34:58 -0800, mattf wrote:
>>>
>>>> Just ran into a, uh, interesting problem with the SAX parser in IDL. At
>>>> one point in processing an XML file, I have to remove the ignorable
>>>> white space-- so I wrote a 'remove_white' subclass of the SAX parser to
>>>> do that (a good exercise, btw, for anyone who's learning about this
>>>> stuff). However, it turns out that the file I want to process has
>>>> two-byte characters-- the 'remove_white' parser isn't bothered by that,
>>>> but it appears to have a problem with recognizing a Windows newline
>>>> (i.e., linefeed + carriage return) in two-byte character form.
>>>> Specifically, it spots the carriage return, but not the linefeed-- so,
>>>> my output file still has all the linefeeds. This is easy enough to deal
>>>> with, in the subclassed 'Chars' method I test for
>>>>
>>>> chars[0] ne string(10B)

>>>>
>>>> before sending the string to the output file. But I think it's a bug.
>>>
>>> If I look at a single-byte text file on Windows, the newline sequence is
>>> carriage return + linefeed, not LF + CR, as you indicate.
>>>
>>> This might be enough for the parser to behave as you describe.
>>>
>>> Did the file undergo any newline translation as it may have been moved
>>> to/from a Unix box? (It may have been treated as a binary file, and NOT
>>> get the correct newline translation)
>>>
>>> Does reversing the newline sequence help?
>>
>> Interesting suggestion. I'll look at that tomorrow
>
> I checked the original file, and it looks like the carriage return and
> line feed are in the correct order, so the binary form of the four-byte
> new line is:
>
> 00 0D 00 0A

OK, that was worth checking.

IDL uses Xerces to implement the SAX and DOM parsers and so any decisions about whitespace are made down in the Xerces library. It is possible that the Xerces library has a bug. You might want to file a problem report with ITTVIS technical support. If there is a newer version of Xerces available that fixes the bug, then we could potentially upgrade the library.

I assume that your file has a correct encoding attribute in the top tag, since the rest of the file seems to parse OK.

I also assume that you've tested your program with a file with a different encoding, like plain old 1-byte ASCII, to make sure it handles the white space the way you want.

A final suggestion, which is a lot of work for you, would be to try to use the IDLffXMLDOM parser with the whitespace filter turned on, and see if you get any Text nodes with only linefeeds in them. This is a longshot since the SAX and DOM parsers share a lot of code, but perhaps the DOM parser gets this right.

Karl