
Subject: EOL problem
Posted by [gridem](#) on Sat, 06 Jun 2009 19:12:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Using XML serialization I found the problem with EOL symbols: XML file was generated using '\n' on Windows platform. This is not a problem in many situations. But this file is opened in Windows notepad incorrectly.

So to correct this behavior on various platforms I prepared the patch which fixes my and other problems with EOL. I think it will be useful to include such patch into the main repository.

Patch was created on 1270 revision.

File Attachments

1) [upp_EOL_fix.patch](#), downloaded 382 times

Subject: Re: EOL problem
Posted by [mirek](#) on Sun, 07 Jun 2009 19:51:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

gridem wrote on Sat, 06 June 2009 15:12 Using XML serialization I found the problem with EOL symbols: XML file was generated using '\n' on Windows platform. This is not a problem in many situations. But this file is opened in Windows notepad incorrectly.

So to correct this behavior on various platforms I prepared the patch which fixes my and other problems with EOL. I think it will be useful to include such patch into the main repository.

Patch was created on 1270 revision.

I am not sure this is the right approach. I believe that this will make Win32 code emit "\r\n" while Linux code will emit "\n".

Anyway, that is not too good for interchangeable format, is not it?

Maybe the better approach would be to generate "\r\n" always?

(So much trouble for single stupid notepad.exe...)

Mirek

Subject: Re: EOL problem
Posted by [mirek](#) on Sun, 07 Jun 2009 19:52:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

<http://www.velocityreviews.com/forums/t167618-lf-cr-handling.html>

Mirek

Subject: Re: EOL problem

Posted by [gridem](#) on Tue, 09 Jun 2009 07:51:27 GMT

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

XML specification says:

2.11 End-of-Line Handling

XML parsed entities are often stored in computer files which, for editing convenience, are organized into lines. These lines are typically separated by some combination of the characters CARRIAGE RETURN (#xD) and LINE FEED (#xA).

To simplify the tasks of applications, the XML processor **MUST** behave as if it normalized all line breaks in external parsed entities (including the document entity) on input, before parsing, by translating both the two-character sequence #xD #xA and any #xD that is not followed by #xA to a single #xA character.

So this paragraph denotes how the line will be interpreted during parsing the XML file. As I understand:

CR LF -> LF

LF CR -> LF

CR -> LF

LF -> LF

But this line doesn't tell us what we should do in case of saving the file in XML format. Also the problem appears not from parsing the entity but from saving the spaces between tags itself.

I think that the saving using Windows EOL format will be good solution, because this approach had been used in QTF project.

Sorry for my bad English.
