The Hammer - Feature #4

TH3: Support for source with line endings other as CRLF

03/20/2017 05:38 PM - Wil van Antwerpen

Assignee: % Done: 0% Category: Estimated time: 0.00 hour Target version: Description 0.00 hour Currently the hammer works best with source code that has CR+LF as end of line characters. A.k.a. windows style texts. The editor controls (codemax/scintilla) by themselves have support for all common formats (LF / CRLF / CR). However for scintilla we have to read the text via DataFlex and as such we have to use binary reads as that's the way to preserve to tab symbol when using the readin command. We now detect the eol format by inspecting the first line and can thereby detect either LF or CRLF. If line endings are CRLF we strip the CR before passing the text on as CRLF to scintilla. Another reason we currently have to hardwire to CRLF is because the parser that is used for the code explorer (among others) only understands CRLF. It would be great if we can support all formats. FWIW: Windows = CRLF Linux/Unix = LF	Status:	New	Start date:	03/20/2017
Category: Estimated time: 0.00 hour Target version: Description Currently the hammer works best with source code that has CR+LF as end of line characters. A.k.a. windows style texts. The editor controls (codemax/scintilla) by themselves have support for all common formats (LF / CRLF / CR). However for scintilla we have to read the text via DataFlex and as such we have to use binary reads as that's the way to preserve to tab symbol when using the readin command. We now detect the eol format by inspecting the first line and can thereby detect either LF or CRLF. If line endings are CRLF we strip the CR before passing the text on as CRLF to scintilla. Another reason we currently have to hardwire to CRLF is because the parser that is used for the code explorer (among others) only understands CRLF. It would be great if we can support all formats. FWIW: Windows = CRLF Linux/Unix = LF	Priority:	Normal	Due date:	
Target version: Description Currently the hammer works best with source code that has CR+LF as end of line characters. A.k.a. windows style texts. The editor controls (codemax/scintilla) by themselves have support for all common formats (LF / CRLF / CR). However for scintilla we have to read the text via DataFlex and as such we have to use binary reads as that's the way to preserve that symbol when using the readIn command. We now detect the eol format by inspecting the first line and can thereby detect either LF or CRLF. If line endings are CRLF we strip the CR before passing the text on as CRLF to scintilla. Another reason we currently have to hardwire to CRLF is because the parser that is used for the code explorer (among others) only understands CRLF. It would be great if we can support all formats. FWIW: Windows = CRLF Linux/Unix = LF	Assignee:		% Done:	0%
Description Currently the hammer works best with source code that has CR+LF as end of line characters. A.k.a. windows style texts. The editor controls (codemax/scintilla) by themselves have support for all common formats (LF / CRLF / CR). However for scintilla we have to read the text via DataFlex and as such we have to use binary reads as that's the way to preserve that symbol when using the readIn command. We now detect the eol format by inspecting the first line and can thereby detect either LF or CRLF. If line endings are CRLF we strip the CR before passing the text on as CRLF to scintilla. Another reason we currently have to hardwire to CRLF is because the parser that is used for the code explorer (among others) only understands CRLF. It would be great if we can support all formats. FWIW: Windows = CRLF Linux/Unix = LF	Category:		Estimated time:	0.00 hour
Currently the hammer works best with source code that has CR+LF as end of line characters. A.k.a. windows style texts. The editor controls (codemax/scintilla) by themselves have support for all common formats (LF / CRLF / CR). However for scintilla we have to read the text via DataFlex and as such we have to use binary reads as that's the way to preserve the tab symbol when using the readIn command. We now detect the eol format by inspecting the first line and can thereby detect either LF or CRLF. If line endings are CRLF we strip the CR before passing the text on as CRLF to scintilla. Another reason we currently have to hardwire to CRLF is because the parser that is used for the code explorer (among others) only understands CRLF. It would be great if we can support all formats. FWIW: Windows = CRLF Linux/Unix = LF	Target version:			
 A.k.a. windows style texts. The editor controls (codemax/scintilla) by themselves have support for all common formats (LF / CRLF / CR). However for scintilla we have to read the text via DataFlex and as such we have to use binary reads as that's the way to preserve to tab symbol when using the readIn command. We now detect the eol format by inspecting the first line and can thereby detect either LF or CRLF. If line endings are CRLF we strip the CR before passing the text on as CRLF to scintilla. Another reason we currently have to hardwire to CRLF is because the parser that is used for the code explorer (among others) only 	Description			
 tab symbol when using the readln command. We now detect the eol format by inspecting the first line and can thereby detect either LF or CRLF. If line endings are CRLF we strip the CR before passing the text on as CRLF to scintilla. Another reason we currently have to hardwire to CRLF is because the parser that is used for the code explorer (among others) only understands CRLF. It would be great if we can support all formats. FWIW: Windows = CRLF Linux/Unix = LF 	A.k.a. windows styl	e texts.		
understands CRLF. It would be great if we can support all formats. FWIW: Windows = CRLF Linux/Unix = LF	tab symbol when us LF or CRLF.	sing the readIn command. We	now detect the eol format by inspectin	g the first line and can thereby detect either
FWIW: Windows = CRLF Linux/Unix = LF			CRLF is because the parser that is use	ed for the code explorer (among others) only
Windows = CRLF Linux/Unix = LF	It would be great if	we can support all formats.		
	Windows = CRLF Linux/Unix = LF	X) = CR		
I don't think we have to support only CR as line endings as it isn't common at all.	I don't think we hav	e to support only CR as line e	ndings as it isn't common at all.	