Base Language - Bug #2429

fix null character handling

10/27/2014 01:54 PM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

History

#1 - 10/27/2014 02:17 PM - Greg Shah

- File ges_upd20141002a.zip added

Our first pass at null character handling has been found to be incorrect. The original idea was that when one specified an escaped null character in a string literal, that the 4GL preprocessor would "spacify" the null character and all following chars. Thus one could not embed a null character in a string... or so it was thought.

Because of this mistaken belief, the STRING rule of src/com/goldencode/p2j/preproc/text.g was modified to call character.progressSpacifyNull().

After looking more carefully, it turns out that the 4GL preproc may allow escaped null characters into a string literal, we made our previous mistake because the preprocessor's listing output made it seem like the null wasn't there since it did not appear in that output.

Interestingly enough, even if the null chars can be placed there using escape sequences, trying to assign these to a character variable or otherwise access the characters will often spacify the results, leading to the impression that the escape chars aren't there.

The most important finding so far is that the one can get null characters into the character variable by using the 3 parameter GET-STRING function on a MEMPTR or RAW that has null characters. The specified length (3rd parm) must not be -1 and must extend past the location of the null character(s). No way has yet been found to get null characters into a longchar variable.

Once you have the null char in a character variable, it turns out that the 4GL behavior in text processing is quite inconsistent. I started a set of testcases to explore this set of problems.

The following list of questions remains:

- can native library calls that directly access character/longchar types return data that includes embedded nulls?
- test file loading via copy-lob, XML processing, editor:load()
- finish tests for how various 4GL text processing features deal with null characters (see builtins_and_operators_react_to_nulls.p)
- test output using UI (e.g. DISPLAY) and IO statements (e.g. PUT or EXPORT) when the variables being output really have null characters inside
- check if string literals with escaped null chars will cause the same results as usage of character variables that have null data

See testscases/uast/null_character/.

Also attached is the change needed to stop the preprocessor from "spacifying" nulls. If this is checked into P2J, then all the other text processing locations will have to have been fixed up to get the behavior correct. I have included some fixes already, but many places are not yet tested/fixed. For that reason, I'm not including this change until more research is done.

Files

ges_upd20141002a.zip	15.4 KB	10/27/2014	Greg Shah