

Base Language - Bug #8139

OS-COMMAND fails on windows if not in "batch mode"

12/20/2023 04:16 AM - Marian Edu

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

History

#1 - 12/20/2023 04:48 AM - Marian Edu

We're trying to start a 'ui simulator' based on sikuli using OS-COMMAND and this is always failing if the client is not running on batch mode, at least on Windows. The "pid" returned is always -2 and looking at the windows native implementation (process_win.c) that is caused by the fact that in background mode the STDIN is expected to be a TTY.

Now, I don't know the thin difference between batch and background but it does look like the background setting is only applicable to the BatchDriver and remains false for all other types of drivers. Since we're trying to test some user interaction here it is impossible to run in batch/background :(

Just for the sake of completeness the OS-COMMAND works fine when the client is started in batch and background is set to true - this is the only combination that I've found it works for Windows.

#2 - 12/20/2023 05:12 AM - Marian Edu

For what it's worth forcing the SwingGuiDriver to take into account the background option does solve the issue and the Os-COMMAND seems to work just fine so no idea why the TTY enforcement on the STDIN is used.

#3 - 01/31/2024 09:39 AM - Greg Shah

- Assignee set to Eugenie Lyzenko

#4 - 02/01/2024 07:18 PM - Eugenie Lyzenko

Let me clarify the issue details.

The short description is the converted OS-COMMAND statement does not load interactive UI application on Windows as it happens in 4GL. Is it correct understanding?

#5 - 02/02/2024 03:03 AM - Marian Edu

Eugenie Lyzenko wrote:

Let me clarify the issue details.

The short description is the converted OS-COMMAND statement does not load interactive UI application on Windows as it happens in 4GL. Is it correct understanding?

I don't see how at that point one could tell if the application to start has any interactive UI or not and is not about this. There is a condition in the native code that is based on the `background` mode set on the client, it requires the `STDIN` to be a `TTY` in that case else it just doesn't start the application - the fact that it fails silently is less important now.

This is related to the fact that the 4GL GUI client can also be started in `batch` mode (no STDIN/STDOUT), this is not supported in FWD right now and we've tried to accommodate `background` mode in Swing client (gui/chui drivers) and not only in the console driver. There are statements in 4GL that works differently when in batch mode - aka trying to send anything to STDIN throws error for one.

The issue we're trying to solve here is to be able to run some Sikuli tests for GUI/CHUI, for that we need to be able to start the Sikuli script only that is failing on Windows with both Swing GUI/CHUI clients because of that enforcement of the `TTY` STDIN in native code. Changing the code to lift that check is the easy part, problem is why the restriction was put there in the first place or it taking it out could break something else :)