

Base Language - Feature #7875

Interactive child process launching is broken in GUI

10/04/2023 03:55 PM - Roger Borrello

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

History

#2 - 10/04/2023 05:50 PM - Roger Borrello

- File 7676_unix_action.mkv added

- File 7875_windows_action.mkv added

A good amount of background can be found in #7676. Make sure to go through it first.

This required a simple testcase, similar to:

```
unix echo "Hello unix world!".
dos echo "Hello dos world!".
os-command echo "Hello world!".
```

In OE Windows, all three work the same way, whether ChUI or GUI. See 7875_windows_action.mkv video. No keystrokes are required to move to the next statement, just exiting the child CMD.EXE (both ChUI and GUI).

In OE Unix, there isn't a GUI, but the dos command is flagged with an error (** On UNIX, the UNIX statement must be used. (261)). See 7676_unix_action.mkv video.

In FWD on Windows, ChUI hesitates for a keystroke before displaying the child CMD.EXE. Using the mouse to click on the CMD.EXE that launches the child CMD.EXE also seems to display the child CMD.EXE. The Swing client also works, creating the Swing console prior to launching the child CMD.EXE processes. No additional keystrokes are required, just exiting the child CMD.EXE and the next statement executes.

In FWD on Linux, our behavior is different for ChUI and broken for GUI. In ChUI, we do not launch an independent shell, but drop into a child process within the current ChUI shell. Keystrokes are required to launch the child process and move onto the next statement. In GUI, we throw a hotspot error when we reach the pseudoTerminalWait JNI, since we always execute the NCURSE call. No use case passes silent=true:

```
if (!silent)
{
    // in interactive mode we're calling functions in NCURSES directly,
    // as opposed to executing command line utilities for the same purpose
    // (NCURSES library is a requirement anyway so we want to have a more
    // efficient and more controlled use of that interface as well, while
    // also avoiding the hard requirement that other utility programs be
    // installed in addition to P2J)

    // put "Press space bar to continue." message on the last line

    // move cursor to the beginning of the last terminal line
    char *s;
    s = tparm(tigetstr("cup"), tigetnum("lines"), 0, 0, 0, 0, 0, 0, 0, 0);
    putp(s);

    // display prompt
    printf("Press space bar to continue.");
```

```
// move cursor to the end of the last terminal line (as in Progress)
s = tparm(tigetstr("cup"), tigetnum("lines"), tigetnum("cols"),
          0, 0, 0, 0, 0, 0, 0, 0);
putp(s);
}
```

Files

7875_windows_action.mkv	447 KB	10/04/2023	Roger Borrello
7676_unix_action.mkv	195 KB	10/04/2023	Roger Borrello