

Limitations when using SSH jobs for Windows

Scope

- When using SSH jobs users might experience limitations that are specific for Windows operating systems. One reason for such limitations is the fact that SSH is not available with the OS but is implemented by third party products.
 - A complete re-write of the JITL SSH Job is provided by

 [JITL-423](#) - JITL SSH Job should switch from Trilead to JSch implementation by JCraft RELEASED

- FEATURE AVAILABILITY STARTING FROM RELEASE 1.9
- When using this JITL job please consider our experiences concerning limitations with some SSH products for Windows. Before posting a support request please check the following behavior of your SSH server and verify by use of a command line tool if possible SSH issues were related to the SSH server or to the JITL job.
- The list of products and limitations is not complete, please contact your SSH server vendor for details.

Copssh

[Copssh](#) is a Windows SSH Service based on the Linux SSH implementation OpenSSH provided by Itefix in collaboration with Cygwin. We experienced the following limitations when using Copssh 5.0.3:

- Interactive Shell
 - Problem
 - The Windows shell can only be used as an interactive shell and does not support command or script execution.
 - The reason for this behavior is the fact that CopSSH does not support the non-interactive shell `cmd .sh`.
 - For details see <https://www.itefix.net/content/windows-shell-ftp>.
 - Solution
 - Using the "Bash and sFTP shell" with Copssh will work for Windows systems when using the cygwin supported Linux commands.
 - Examples for Limitations
 - Limitation: the Windows command `set` will not be recognized.
 - Workaround: use `export` instead.
 - Limitation: the `%` special character to mark an environment variable such as `%VARIABLENAME%` will not be recognized.
 - Workaround: use `$VARIABLENAME` instead.
 - Hint: be careful when using mixed case spelling for environment variables.
 - Limitation: executing chained Windows commands in one line with the command delimiters `&` or `&&` will result in executing only the first command.
 - Workaround: use the Linux command delimiter `;` together with the proper Linux command instead.
 - Limitation: prefixing a command with `cmd /C` does not execute the specified command.
 - Workaround:
 - check: <http://stackoverflow.com/questions/17582778/how-to-execute-a-bat-file-from-cygwin-bash-that-uses-the-windows-find-command>

probably link to the article, is

```
cmd /C start "clean shell" /I "${EXPLORER_WINPATH}" "${SCRIPT_WINPATH}"
```

working?

- example: start "My Notepad" /I c:/tmp/hallo.cmd
- tested:
 - with `cygstart -w launcher.bat ./test.cmd`
 - `cygstart` has to be available, it is not integrated in the Copssh installation, but available through the standard cygwin installation
 - nothing happens in the ssh connection
 - JUnit Test fails with exit code 127
 - with `cmd /C start /I ./test.cmd`
 - nothing happens in the ssh connection
 - JUnit Test hangs as soon as the command will be executed remotely

bitwise SSH Server

- Environment Variables
 - Limitation: setting environment variables with `set` would not allow using this variable in further commands.
 - Example: unsuccessful use of an environment variable in a subsequent command:
 - `set TESTFILE=test.txt`
 - `echo MYPARAM=myTestParam >> %TESTFILE%`

- This example shows that %TESTFILE% is not substituted, instead a file with the name %TESTFILE% is being created and **MYPARAM=myTestParam** is being written to that file.