RXSHELL V2.1

RXSHELL is a TSO utility to run z/OS UNIX shell sessions in an interactive mode from foreground ISPF. It is not supported and not needed in background. This routine is based on the old RXSHELL utility of USSTools of the UNIX Tools disk with many enhancements as described following.

RXSHELL updates in V2.0

- RXSHELL picks up the terminal emulation code page from ISPF and the user can enter the commands as if the session is set up with code page 01047.
- You can provide an MVS PS data set named '*tsouser*.RXSHELL.ENVVARS' for RXSHELL to specify your desired settings for z/OS UNIX environment variables. The HLQ is your TSO userid. By default the PATH variable is set to value "PATH=/usr/local/bin:/bin".
- RXSHELL waits for end of output data and then prints a prompt (**>>**) and allows to enter a next command. Sometimes it may happen that the output is not complete. Simply press ENTER to show additional output.

RXSHELL updates in V2.1

- On entering a command starting with tso -o ... then the following command is addressed to the TSO environment where you called RXSHELL. This allows to get back to TSO/ISPF temporarily, for example when using utility GO being part of the USSTools package.

Syntax of command RXSHELL

Following the syntax of RXSHELL is shown:

```
rxshell [shell_cmd]
```

- By default RXSHELL opens the shell program as found in the OMVS segment. If this is /bin/sh or /bin/tcsh a full login shell is opened. You can run any valid z/OS UNIX command that accepts line input data and that does not need extended terminal functions and simply provides line data output.
- When the UNIX environment is closed the last UNIX return code is reflected in a final message.
- When the environment gets into a hang-up situation you can try to use the escape key and enter command hi to stop the RXSHELL processing.
- Unlike the OMVS shell, for example, RXSHELL still works fine when the user's home directory does not exist.

Sample RXSHELL sessions using default and explicit UNIX commands

Following sample rxshell sessions run from ISPF as references...

```
rxshell
 >>
pwd; invalid_command; id
 /u/hering
 invalid_command: FSUM7351 not found
 uid=888(HERING) gid=2(SYS1) groups=1047(USSTEST)
 >>
exit
 RXSHL009I The shell session ended normally.
 ***
```

```
rxshell sh -c invalid_command
 invalid_command: FSUM7351 not found
 RXSHL010E Errors occurred; shell gave RC= 127.
 ***


rxshell sh -c id && pwd
 uid=888(HERING) gid=2(SYS1) groups=1047(USSTEST)
 /u/hering
 RXSHL009I The shell session ended normally.
 ***

rxshell rexx
 >>
s "getcwd cwd"; say cwd
 Rx> s "getcwd cwd"; say cwd
 /u/hering
 OMVS Return Value (retval) = 9
 >>
exit
 Rx> exit
 RXSHL010E Errors occurred; shell gave RC= 255.
 ***

rxshell
 >>
cn "d omvs,o" | grep "SYSPLEX " | awk '{ print $1 $2 $3 }'
 >>
[ enter ]
 SYSPLEX=YES
 >>
exit
 RXSHL009I The shell session ended normally.
 ***
```

How to handle a possible hang-up situation

If you get into a hang-up situation you can try to use the escape or pa1 key plus command HI to quit the RXSHELL environment. Following a sample with stopping the hang-up is shown for reference...

```
rxshell
 >>
sleep 600
 >>
exit
 >>
pwd
 >>
[ escape ]
 !
 IRX0920I ENTER HI TO END, A NULL LINE TO CONTINUE, OR AN IMMEDIATE …
hi
[ enter ]
 RXSHL014W Processing halted; exiting...
```

***

This is a pseudo hang-up situation as we started for waiting 10 minutes. Entering new commands are simply queued. After pressing the escape or PA1 key you are prompted for an action. You should enter "hi" to halt processing and exit.