

## rexx

REXX procedure REXX is supported in ISPF/TSO (using command `tso rexx`) and UNIX System Services shell environments (using `rexx`). It allows you to run TSO, SYSCALL, UNIX System Services Shell and REXX commands interactively. It was originally provided with the USSTools package. Here is the most current version for easy access to it.

If you enter an empty or blank line in interactive mode, help information is displayed describing the usage as follows.

The REXX command input syntax is: `[ prefix ] cmd_input`

The prefix can be specified as one of the following:

<b>SH</b>	The input data <code>cmd_input</code> is run as a UNIX command.
<b>Syscall</b>	The input data <code>cmd_input</code> is run as a SYSCALL command.
<b>TSO</b>	The input data <code>cmd_input</code> line is run as a TSO command.
<b>MVS</b>	The command is run with addressing environment MVS.

If you do not specify a prefix, then the command input is only interpreted by REXX. The command input data may contain several commands, separated by a semi-colon (;). You can extend the amount of data to multiple lines by appending a dash (-) at the end of the current line.

Important: You must use an explicit `Exit` statement to end processing. This is needed especially in batch mode; otherwise, the REXX will not stop.

Some examples:

```
$> id
uid=888(HERING) gid=2(SYS1) groups=1047(USSTEST)
$> rexx
SH> s "getuid"
OMVS Return Value (retval) = 888
SH> sh "id"
uid=888(HERING) gid=2(SYS1) groups=1047(USSTEST)
SH> s "setreuid 0 0"
OMVS Return Value (retval) = 0
SH> s "getlogin mvssuid"
OMVS Return Value (retval) = 621915156
SH> Say mvssuid
HERING
$> id
uid=888(HERING) gid=2(SYS1) groups=1047(USSTEST)
$>
```