RXSBATCH V1.6

The REXX procedure RXSBATCH has been created to run z/OS UNIX commands similar to RXBATCH. It does not use BPXBATCH but shell programs explicitly. It should provide all the functions of RXBATCH and more:

- You can use DD name STDIN for STDIN data as desired and can write the data like a UNIX shell script.
- You can work with DD names in UNIX commands as it is assured that you stay in the same ASID.
- You can code your UNIX STDIN data using your preferred code page. This makes it easy to write the commands and they are much better readable.
- You can use the back slash ("\") if you need to continue the command on a next line. This works even if you are using record format FB for your JCL library.
- The combination of several advantages mentioned above allows to use MVS data sets or members, created in your code page as scripts for usage as STDIN data.
- You can switch to other sub shells or interactive command processors, like using "su", "su -s" or the interactive "rexx".
- All the added functions of RXBATCH are supported as well. Especially final error messages are written to the job message log. Also the first ten cases of truncated STDOUT lines are mentioned there.
- Unlike standard BPXBATCH, for example, utility RXSBATCH still works fine when the user's home directory does not exist. BPXBATCH shows message **`FSUM1004 Cannot change to directory </home_dir>.`** and ends.

New function in RXSBATCH V1.1

- A problem has been corrected with the current working directory if the home directory does not exist.

New functions in RXSBATCH V1.2

- If establishing the SYSCALL environment fails the error code is provided with the message shown.
- The names of temporary files used are made better unique to avoid using the same name twice in high parallel processing environments.
- It is allowed to set a fixed error return code instead of using the final z/OS UNIX return code. The UNIX error code is then shown at the end of the z/OS UNIX output data or within the job message log.

New functions in RXSBATCH V1.3

- Support for using explicit STDERR ddname has been added.
- Support for using "bash" has been added.

New functions in RXSBATCH V1.4

- A small typo, found by Manfred Lotz during testing, has been corrected.

New functions in RXSBATCH V1.5

- You can use environment variable TMPDIR to provide a directory that is used for temporary files by z/OS UNIX instead of the default "/tmp". Placing this setting to STDENV in a batch job using RXSBATCH allows to use this temporary files directory for RXSBATCH itself as well as for the z/OS UNIX environment.

New functions in RXSBATCH V1.6

- Support has been added to allow that just STDERR is used and no STDOUT. All STDERR messages are shown in STDERR in this case just like BPXBATCH is doing it.

Functions of RXSBATCH over RXBATCH

- It supports the standard shell (/bin/sh), the C shell and the "bash".
- It supports output conversion from the standard UNIX EBCDIC code page to the input or another code page.
- The exit error code from the shell is reflected as the job return code. No extra RC handling is needed.
- It does more internal checking for STDIN, STDOUT and STDERR ddnames.

Some important remarks

- The following LE run-time options should be set at least in order to run RXSBATCH successfully: **`ALL31(ON),STACK(128K,128K,ANYWHERE,KEEP,512K,128K)`**
- Low region sizes can lead to abends (**`USER COMPLETION CODE=4088 REASON CODE=000003EC`**). Best is to set **`REGION=0M`**.

RXSBATCH variables

The procedure supports a set of four RXSBATCH variables to set some specific control switches for processing (similar as RXBATCH does).

_RXBATCH_SWSU	Do (value 1) or do not (value 0) try to switch to superuser mode. The default is 0 (do not switch).
_RXBATCH_LOGIN	Use a login shell (value 1) or do not (value 0). The default is 1.
_RXBATCH_PGM	Specify the shell program to be used as "/bin/sh", "/bin/tcsh" or "bash". The default is "/bin/sh".
_RXBATCH_CPIN	Specify the code page for the STDIN data. The default code page is IBM-1047.
_RXBATCH_CPOUT	Specify the code page for the STDOUT data. The default is the STDIN code page. Specify "CPIN" to explicitly use the default value.
_RXBATCH_ERRRC	Use the value as the return code if it is a whole positive number. Otherwise use the z/OS UNIX return code. The latter is the default.
_RXBATCH_ERRMS	If a fixed RC is requested in case of an error show the z/OS UNIX RC in the job message log (by specifying "JMSGLG") or at the end of the UNIX output data. The latter is the default.

RXSBATCH sample job and output

Following you find a sample JCL for using RXSBATCH:

```
//USSJOB   JOB ,'USS Commands',NOTIFY=&SYSUID.,REGION=0M,SYSNAME=xxxx
//* ------------------------------------------------------------------
//  SET REXXLIB=HERING.UNIX.REXX.EXEC             <=== SYSEXEC library
//* ------------------------------------------------------------------
//RXSBATCH EXEC PGM=IKJEFT01,PARM=RXSBATCH
//SYSEXEC  DD DSNAME=&REXXLIB.,DISP=SHR
//STDIN    DD DATA,DLM=##
echo aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaazzzz | cat
cp //DD:TEST /dev/fd1
rexx
do 2; say Copies("xX",100); end
exit 0
_BPX_SHAREAS=YES sftp; RC=$?
pwd
exit $RC
//STDENV   DD DATA,DLM=##
PATH=/usr/local/bin:/bin
##
//RXBPARM  DD DATA,DLM=##
 _RXBATCH_SWSU=0        < 0= no switch (default), 1= switch to SU mode
_RXBATCH_LOGIN=0        < 0= no login shell, 1= login shell (default)
```

```
  _RXBATCH_PGM=/bin/sh  < shell pgm "sh" or "tcsh", default is "/bin/sh"
_RXBATCH_CPIN=IBM-273   < code page of STDIN data, default is IBM-1047
_RXBATCH_CPOUT=CPIN     < STDOUT code page, default is STDIN CP (=CPIN)
_RXBATCH_ERRRC=08       < Use as job RC if a number else USS RC (def)
##
//TEST      DD DATA,DLM=##
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz
##
//STDOUT    DD SYSOUT=*,LRECL=136,RECFM=VB
//SYSTSIN   DD DUMMY
//SYSTSPRT DD DUMMY    < Use "SYSOUT=*" instead in case of problems
//* ------------------------------------------------------------------
```

Following the joblog output written to STDOUT...

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaazzzz
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz
Rx> do 2; say Copies("xX",100); end
xXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxX
xXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxX
xXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxX
xXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxXxX
Rx> exit 0
usage: sftp [-1246Cpqrv] [-B buffer_size] [-b batchfile] [-c cipher]
.        [-D sftp_server_path] [-F ssh_config] [-i identity_file] [-l limit]
.        [-o ssh_option] [-P port] [-R num_requests] [-S program]
.        [-s subsystem | sftp_server] host
.   sftp [user@]host[:file ...]
.   sftp [user@]host[:dir[/]]
       sftp -b batchfile [user@]host
/u/hering

Shell ended with UNIX return code 1.
```

The job return code is 8 (forced by _RXBATCH_ERRRC=08). The UNIX return code is 1, which is the code we get when entering sftp with no parameters.

Using file or data set for STDOUT

You can also use an MVS data set or a UNIX file for STDOUT or STDERR.

```
//STDOUT    DD PATH='/u/hering/rxsbatch.stdout',
//             PATHOPTS=(OWRONLY,OTRUNC,OCREAT),
//             PATHMODE=(SIRUSR,SIWUSR)
```