rxsubmit V1.1

The USS REXX rxsubmit allows to submit and monitor a JES2 batch job. JES3 is not supported! It is provided in a ZIP file, is in ASCII (ccsid 819) and needs to be converted to EBCDIC (ccsid 1047 or 037) during transfer to z/OS UNIX or afterwards when transferring it it in binary mode.

- It retrieves the jobname from the job member or file and gets the jobid back from the USS REXX submit() call.
- It uses the REXX SDSF APIs to monitor the job and to retrieve job output data sets to the USS shell.
- It uses the JESMSGLG data set to see where we are in the job and especially how and when the job ends.
- JESJCL and JESYSMSG are ignored. All other data sets are seen as job output data and are retrieved to shell STDOUT.

To be sure to handle the output in JESMSGLG correctly you need to adapt the following lines in rxsubmit that are based on the follwing output data structure.

```
 08.45.46 JOB28016 ---- WEDNESDAY, 27 JAN 2021 ----
 08.45.46 JOB28016  IRR010I  USERID HERING   IS ASSIGNED TO THIS JOB.
 08.45.46 JOB28016  ICH70001I HERING  LAST ACCESS AT 08:36:02 ON WEDNESDAY, JANUARY 27, 2021
 08.45.46 JOB28016  $HASP373 UNIXJOB  STARTED - INIT 1    - CLASS A        - SYS SC70
 08.45.46 JOB28016  IEF403I UNIXJOB - STARTED - TIME=08.45.46 - ASID=001F - SC70
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
 08.45.51 JOB28016  Jobname  Procstep Stepname  CPU Time      EXCPs     RC
 08.45.51 JOB28016  UNIXJOB  --None-- UNIXCMDS  00:00:00        400    256
 08.45.57 JOB28016  UNIXJOB  --None-- UNIXCMD2  00:00:00        392     00
 08.45.57 JOB28016  IEF404I UNIXJOB - ENDED - TIME=08.45.57 - ASID=001F - SC70
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
 08.45.57 JOB28016  $HASP395 UNIXJOB  ENDED - RC=0256
```

Here are the lines that you might need to adapt perhaps.

```
Do ln=1 To line.0
  line = Strip(line.ln)
  Parse Var line . ljobid ljobname . 1 . . msgnum .,
    step_name . . retc .
  Select
    When ljobid=jobid & msgnum="$HASP373" Then Do
      Say "Job" jobname "with Jobid" jobid "has been",
        "started."
      job_started = 1
    End
    When ljobid=jobid & ljobname=jobname & job_started &,
      Verify(retc,"1234567890")=0 Then Do
        job_step = job_step+1
        Say "Jobstep" Right(job_step,2,"0") step_name,
          "ended with rc" Right(retc,4) !! "."
    End
    When ljobid=jobid & msgnum="$HASP395" Then Do
      RC_info = Word(line,7)
      Say "Job" jobname "with Jobid" jobid "ended with",
        RC_info !! "."
      job_running = 0
      Leave ln
    End
    Otherwise Nop
  End
End ln
```

Following examples are shown how to start "rxsubmit".

- Using a data set or data set member as parameter...
  `rxsubmit "//'hering.job.cntl(unixsamp)'"`
- Using a z/OS UNIX file as parameter...
  `rxsubmit ./my.private.job`
- Reading job data from STDIN...
  `"cat "//'hering.job.cntl(unixsamp)'" | rxsubmit`

It can happen, that you start "rxsubmit" without specifying STDIN data or a parameter as useful. Then you can use <ctl-v> or <ctl-d> to end processing. However, you could also enter the complete job data by hand, although that is really not ideal. Best is to use <ctl-d> which means End-Of-Input or end of STDIN data.

```
$> rxsubmit
<ctl-d>
The job data does not contain any lines. This is invalid.
Rc(2)
$>
```

Next a sample shell session is shown.

```
$> rxsubmit "//'hering.job.cntl(unixsamp)'"
Job UNIXJOB with Jobid JOB28016 has been started.
This is a test job.
Jobstep 01 UNIXCMDS ended with rc  256.
We are done...
xleep: FSUM7351 not found
This is a test job.
We are done...
Jobstep 02 UNIXCMD2 ended with rc   00.
Job UNIXJOB with Jobid JOB28016 ended with RC=0256.
$>
```

Note, that you also can send the command to BG by appending an ampersand (&) at the end of the command. This frees the shell for running other commands in parallel but keep in mind that output data from the job and commands entered can get shown mixed on the screen.