

REXX Users Panel

Sample REXX Usage @ Hanes Companies

Steve Weifenbach
steve.weifenbach@hanesfinishing.com
2003/04/25

Four REXX Procs

- Utility Report Processing
 - Create utility report that is reformatted into JCL submitted to maintain a debug source file
- PCL forms conversion from EBCDIC to ASCII
 - FTPed JCL that creates duplicate PCL macros for use by VTAM defined and TCP/ip defined printers
- Email notification of job cancellation
 - Send an email to a cell phone when a job cancels
- VSE Shutdown

Utility Report Processing

We have more CICS source programs than I have allotted DASD for storing the compile source and consequently the VSAM file fills up preventing the programmers from debugging their code interactively. I needed a way to empty the file of older, unused, dead space source.

```
JCL
* $$ LST CLASS=L,DISP=D
// DLBL ABNLDFL,'CSS75.LISTING.FILE',,VSAM,CAT=VSESPUC
// EXEC DBDDSUTL,SIZE=256K
D
/*
* $$ PUN CLASS=R,DISP=I
* $$ LST CLASS=L,DISP=D
// EXEC REXXBAT,SIZE=REXXBAT,PARM='DBDDSUTL'
/*
```

Utility Report Processing *(continued)*

Report Output

ENTER UTILITY COMMAND:

D

PROGRAM	LIST NUMBER	COMP DATE	TIME	RC	LANGUAGE	SIZE(K)
KORH	(L) 16721	25 APR 2003	11.21.49	00	DOS COBOL	1533
...						
ZPSE	(L) 16387	09 APR 2003	15.11.32	00	DOS COBOL	1917
KIVB	(L) 16319	07 APR 2003	09.45.35	00	DOS COBOL	1022

ENTER UTILITY COMMAND:

*** ALL FUNCTIONS SUCCESSFULLY COMPLETED

REXX output to the RDR queue for submission

```
* $$ JOB JNM=DBDDSUTL,CLASS=A,DISP=D,USER=00004
* $$ LST CLASS=L,DISP=D
// JOB DBDDSUTL
/*
// DLBL ABNLDFL,'CSS75.LISTING.FILE',,VSAM,CAT=VSESPUC
// EXEC DBDDSUTL,SIZE=256K
DELETE DN=16319
/*
/&
* $$ EOJ
```

PCL Forms Conversion

We needed to duplicate the PCL forms overlay for both VTAM and TCP/ip defined printers. The protocol conversion process is different between the printers. Sometimes the steps to produce forms were not followed. Consequently, the forms would be different, creating user consternation.

JCL

```
* $$ PUN CLASS=R,DISP=I
// EXEC REXXBAT,SIZE=REXXBAT,PARM='M4VAFLSH A101'
1B451B266C316F36633565304C1B2661354C
1B2A63323044
1B2838551B2873307031362E363768382E3576307330623054
1B2A633646
1B28323058
/*
```

PCL Forms Conversion (continued)

Intermediate REXX output to the RDR queue for submission

```
// OPTION CATAL,NODECK
// EXEC ASSEMBLY,SIZE=72K
      PUNCH ' PHASE M4VAA101,+0,NOAUTO'
      TITLE ' M4VAA101 FOR PCL PRINTERS'
M4VAA101 CSECT
      DC X'4A5B' * CENT SIGN AND QUESTION
      DC C'1B451B266C316F36633565304C1B2661354C1B2A633230441B28'
      DC C'38551B2873307031362E363768382E35763073306230541B2A63'
      DC C'36461B28323058'
      DC X'4A' * CENT SIGN
      END

/*
// EXEC LNKEDT
/*
// LIBDEF PROC,SEARCH=ALIB.TEST
// EXEC REXX=REXXFLSH,SIZE=1024K,PARM='ALIB.PROD A101 X101 '
/*
```

PCL Forms Conversion (continued)

Final REXX output to the RDR queue for submission

```
* $$ JOB JNM=M4VAFLSH,DISP=H,PRI=3,CLASS=R,USER='00004'
* $$ LST CLASS=L,DISP=D
// JOB M4VAX101
/*
// LIBDEF PHASE,CATALOG=ALIB.PROD,TEMP
// OPTION CATAL,NODECK
// EXEC ASSEMBLY,SIZE=72K
      PUNCH ' PHASE M4VAX101 ,+0,NOAUTO'
      TITLE ' M4VAX101 FOR PCL PRINTERS'
M4VAX101 CSECT
      DC X'27C5275093F196F683F585F0D3275081F5D3275C83F2F0C4274D'
      DC X'F8E4274DA2F097F1F64BF6F788F84BF5A5F0A2F082F0E3275C83'
      DC X'F6C6274DF2F0E7'
      END

/*
// EXEC LNKEDT
/*
/&
* $$ EOJ
```

Email Notification

When converting to TCP/ip, we removed all the local 3174s. Our problem notification process still used a coax connection breaking notification. It could not be changed to TCP/ip and we had to find an alternative method of notification. Via TCP/ip we can send emails to the on-call person's cell phone.

```
JCL
// ON $CANCEL GOTO JOBEND
// ON $ABEND GOTO JOBEND
// ON $RC>=16 GOTO JOBEND
//          SETPARAM CX=1
//          SETPARAM JX=ADPRALRT
//          SETPARAM MX='CANCELED, SEND MAIL TO SYSTEM PGMR'
...
// GOTO ENDJOB
/. JOBEND
// EXEC PROC=JOBEND,CX,JX,MX
/*
/. ENDJOB
```

Email Notification (continued)

```
JOBEND.PROC
// PROC UX='SYS',CX='1',JX='UNKNOWN',                                X
//          MX='CANCELED, NOTIFICATION AUTOMATED.'
// ON $RC>=16 CONTINUE
//          SETPARAM S='S'
//          SETPARAM A='A'
//          SETPARAM B='B'
//          SETPARAM C='C'
//          SETPARAM PH='PH'
//          SET VSE S PHSYS
//          SET VSE A PHHCI
//          SET VSE B PHHDF
//          SET VSE C PHHCC
// EXEC REXX=REXXDIV,SIZE=1M,PARM='&A &B &C &S',PH
// EXEC REXX=REXXMAIL,SIZE=1M,PARM='&PH &MX'
```

Email Notification (continued)

Behind the scenes, FAQs variables contain the current on-call person's cell phone number for each division, maintained by the help desk. REXXDIV is passed the current numbers, determines which division is responsible for the job that cancelled and passes that phone number to REXXMAIL. REXXMAIL generates JCL that is submitted to send notification via email to that cell phone.

Rich Smrcina supplied the original source for REXXMAIL.

Email Notification (continued)

Final REXX output to the RDR queue for submission

```
* $$ JOB JNM=ADPRMAIL,DISP=D,PRI=3,CLASS=S,USER='00004'  
* $$ LST CLASS=L,DISP=D  
// JOB ADPRMAIL  
// LIBDEF *,SEARCH=(PRD2.CONFIG),TEMP  
// EXEC BSTMTMTPC,SIZE=BSTMTMTPC  
ID 01  
OPEN 192.168.1.6  
HELO mail.whitehouse.gov  
MAIL From: mainframe@whitehouse.gov  
RCPT To: 8775551212@messaging.cellcarrier.com  
SUBJ Subject:  
ORGA Organization: Whitehouse  
DATA  
QUIT  
/*  
CICSICCF HAS TERMINATED UNEXPECTEDLY.  
/*  
/&  
  
* $$ EOJ
```

VSE System Shutdown

```
/* This REXX IMOD will issue all commands for the shutting down */
/* of our system. */

parse upper arg pend
pid = pidlist('B') /* get busy partitions and jobnames */
j = (length(pid)%2) - 1
n = 0
do i = 0 to j
  p = substr(pid,i*2+1,2)
  jnm = jobname(p)
  if jnm ^= 'NO NAME' then do
    n = n + 1
    x.n = jnm /* jobname & pid into stem vars */
    y.n = p
  end
end
end

(continued)
```

Example: Shutdown (continued)

```
do i = 1 to n
  if substr(x.i,1,4) == 'CICS' then do
    call shutcics x.i PEND
    if result='1' then do
      say 'ZFAQS: halting termination,' x.i 'still active' TIME()
      exit
    end
  end
  if x.i == "M4VAPRNT" then m4va = x.i
  if x.i == "EXPLCICS" then do
    expc = x.i
    expcp = y.i
  end
  if x.i == "NETTST02" then nettst = x.i
  if x.i == "NETFTP01" then netftp = x.i
  if x.i == "EVSEIBAT" then evse = x.i
  if x.i == "NETTEL00" then nettel = x.i
end

(continued)
```

Example: Shutdown (continued)

```
CALL SEVENT DELETE ABNDVPRT /* delete event that monitors */
if m4va == "M4VAPRNT" then call shutvprt
if expc == "EXPLCICS" then ADDRESS CONSOLE 'MSG' expcp
if netst == "NETTST02" then call shuttcp netst
if netftp == "NETFTP01" then call shuttcp netftp
call TERMINAT NOTIFY /* start FAQS shutdown */
ADDRESS CONSOLE 'Z NET,QUICK' /* terminate vtam */
if evse == "EVSEIBAT" then ADDRESS CONSOLE 'A RDR,EVSEPBAT,DISP=K,CLASS=S'
if nettel == "NETTEL00" then call shuttcp nettel
ADDRESS CONSOLE 'V NET,INACT,ID=H04NCP0,FORCE' /* sometimes NCP */
/* hangs VTAM */
ADDRESS CONSOLE 'PEND' /* terminate POWER */

ADDRESS CONSOLE 'SIR'
ADDRESS CONSOLE 'QUERY TD'
ADDRESS CONSOLE 'SYSDEF TD,RESETCNT'

say 'ZFAQS: done! ... at' TIME()
call TERMINAT SHUT /* finish FAQS shutdown */

EXIT
```

Example: SHUTTCP imod

```
/* SHUTTCP REXX IMOD WILL find the partition running 'jobname' */
/* and issue the commands to shut that job down */

Parse upper arg jnm junk
jnm = strip(jnm)

pid = pidlist('B')
j = (length(pid)%2) - 1
do i = 0 to j
  y = substr(pid,i*2+1,2)
  x = jobname(y)
  if rc = 0 then do
    if x = jnm then do /* if jnm then shut it down */
      say 'SHUTTCP: continuing at' TIME()
      x = replid(y) /* is there a message outstanding */
      if rc = 0 then do
        if x = '' then do /* no, then request one */
          ADDRESS CONSOLE 'MSG' y
        end
      end
    end
  end
end
(continued)
```


Example: SHUTTCP imod

```
if rc = 0 then do
  call getptnid      /* try until there is a message */
  z = substr(x,5,3)
  ADDRESS CONSOLE z 'SHUTDOWN'
  if rc = 0 then do
    call getptnid      /* try until there is a message */
    z = substr(x,5,3)
    ADDRESS CONSOLE z 'YES'
    EXIT
  end
end
else do
  say 'REPLID failed rc='||rc
  EXIT
end
end
```

(continued)

Example: SHUTTCP imod

```
else do
  say 'REPLID failed rc='||rc
  EXIT
end
end
end
end
EXIT

getptnid:
do forever until x = '' /* try until there is a msg */
  x = wait(2)
  x = replid(y)      /* is there a msg outstanding now */
end
return
```

Hanes Examples

Beendet - Irgendwelche fragen?

Finished - Any questions?