

# Introduction To REXX

WAVV 2006 –Chattanooga

By Charles Rice

# What Is REXX?

- ◆ REXX - restructured extended executor
- ◆ REXX is a general purpose programming language
- ◆ A very powerful high level programming tool
- ◆ A language processor runs REXX programs
- ◆ REXX has the usual structured-programming instructions--IF, SELECT, DO WHILE, LEAVE, etc. And very useful built-in functions for accessing and using system interfaces and facilities

# Parts of REXX

- ◆ REXX KERNEL - SAA procedure language.
  - REXX is common to all IBM processing systems.
  - VSE partial SSA implementation.
- ◆ REXX/370 LIBRARY - common to VSE/VM/MVS.
  - Compiler supported for VM/MVS only.
    - I.E.. Compile on VM/MVS run on VSE.
  - Or interpretive execution on VSE/VM/MVS.
- ◆ VSE system dependent interfaces - provides access to VSE/ESA resources and subsystems.

# REXX Execution

- ◆ Catalogue REXX program into VSE library as a proc
- ◆ Run the program with a
  - `// EXEC REXX=XXXXXXXXXX`

# REXX VSE System Dependent

- ◆ Job control language interface.
- ◆ Console interface.
- ◆ Application framework.
- ◆ Batch job management capabilities.
- ◆ POWER interface.

# REXX VSE System Dependent

- ◆ VSE batch services like LIBR/IDCAMS.
- ◆ I/O functions for VSE library and VSE files.
- ◆ CICS interface with CICS/TS.
- ◆ TCP/IP sockets interface.

# ADDRESS JCL

- ◆ REXX extends the capabilities of VSE Job Control and offers a new dimension for JCL
  - Logical creation of JCL on the fly
  - JCL loops
  - Conditional checking using REXX facilities while building JCL
  - Issuing commands, trapping output, reformatting output and then customized output
  - The possibilities are endless

# ADDRESS JCL

- ◆ Use a REXX program to create JCL in the Stack to pass to VSE
- ◆ Example:
  - /\* REXX STACK FULL OF JCL \*/
  - PUSH '// assgn sys001,181'
  - PUSH '// assgn sys002,182'
  - PUSH '// assgn sys003,183'
  - EXIT 0



# ADDRESS CONSOLE

- ◆ Enables you to automate and make more productive the operation of your VSE/ESA console.
- ◆ You activate & deactivate one or more console sessions.

# ADDRESS CONSOLE

ADDRESS CONSOLE

'ACTIVATE NAME VSECONA PROFILE REXNORC'

'MAP'

RC=GETMSG(MSG. , 'RESP' , , , 5)

# vseconx

```
// JOB CATAL
// EXEC LIBR
A S=VENDOR.REXX
CATALOG VSECONX.PROC                      REPLACE=YES
/* ----- */
/* ISSUE MAP COMMAND FOR STATIC PARTITIONS */
/* ----- */
FC=SYSVAR('SYSPID')
CONNAME='VSECON' || SYSPID                /* MAKE CONSOLE NAME UNIQUE */
ADDRESS CONSOLE
COM1='ACTIVATE NAME' CONNAME 'PROFILE REXNORC'
COM1                                       /* ACTIVATE THE CONSOLE */
'MAP'
RC=GETMSG(MSG., 'RESP', , , 5)
/* ----- */
/* BUILD PARTITION TABLE FOR STATIC PARTITIONS */
/* ----- */
PART. 0=0
DO I=4 TO 15
  PARSE VAR MSG. I F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11
/* MOVE FIELDS FOR MULTIPLE PARTITIONS IN SAME ADDRESS SPACE */
  IF F4 = "*" THEN DO
    F4 = F5
    F5 = F6
    F6 = F7
    F7 = F8
    F8 = F9
    F9 = F10
    F10 = F11
  END
/* ADD TO PARTITION TABLE IF ACTIVE JOB IN PARTITION */
  IF F10 \= " " THEN DO
    J=PART. 0+1
    PART. J = F4
    JOB. J = F10
    PART. 0 = J
    JOB. 0=J
  END
END
/* ----- */
/* GET LIST OF ALL DYNAMIC CLASSES */
/* ----- */
'MAP CLASS=ALL'
RC=GETMSG(MSG., 'RESP', , , 5)
CLASS. 0=0
DO I = 2 TO MSG. 0-1
  PARSE VAR MSG. I F1 F2 F3 F4
  J=CLASS. 0+1
  CLASS. J = F3
  CLASS. 0 = J
END
/* ----- */
/* ADD TO PARTITION TABLE FOR ACTIVE DYNAMIC PARTITIONS */
/* ----- */
DO I = 1 TO CLASS. 0
  'MAP CLASS=' || CLASS. I
  RC=GETMSG(MSG., 'RESP', , , 5)
  PARSE VAR MSG. 4 F1 F2 F3 F4
  IF F3 \= "NO" THEN DO
    DO K= 6 TO MSG. 0-1
      PARSE VAR MSG. K F1 F2 F3 F4 F5
      J = PART. 0 + 1
      PART. J = F3
    END
  END
END
```

```

JOB. J = F4
PART. 0 = J
JOB. 0 = J
END
END
END
/* ----- */
/* LIST PARTITION TABLE */
/* ----- */
DO I = 1 TO PART. 0
  SAY PART. I JOB. I
END
/* ----- */
/* GET COMMAND FROM INPUT AND SEND COMMAND TO CONSOLE */
/* ----- */
DO FOREVER
  PULL CMD
  IF CMD=" " THEN LEAVE
/* IF SLEEP COMMAND THE WAIT FOR NUMBER OF SECONDS */
  IF SUBSTR(CMD, 1, 5)="SLEEP" THEN DO
    CALL SLEEP STRIP(SUBSTR(CMD, 7, 5))
  END
  ELSE DO
/* REPLACE XXXX WITH REPLID FROM LAST MSG COMMAND */
    IF SUBSTR(CMD, 1, 4)="XXXX" THEN DO
      IF LENGTH(CMD) = 4 THEN CMD=PARTID
      ELSE
        CMD=PARTID || SUBSTR(CMD, 5, LENGTH(CMD)-4)
      END
    IF SUBSTR(CMD, 1, 3)='MSG' THEN DO
      DO I = 1 TO PART. 0
        IF SUBSTR(CMD, 5, LENGTH(JOB. I)) = JOB. I THEN DO
          CMD=SUBSTR(CMD, 1, 4) || PART. I || SUBSTR(CMD, 5+LENGTH(JOB. I), 40)
          LEAVE
        END
      END
    END
    IF SUBSTR(CMD, 1, 3)='RSP' THEN DO
      DO I = 1 TO PART. 0
        IF SUBSTR(CMD, 5, LENGTH(JOB. I)) = JOB. I THEN DO
          LEAVE
        END
      END
    END
    "REPLID"
    RC=GETMSG(MSG. , 'RESP' , , , 5)
    LOC=POS(PART. I , MSG. 1)
    PARTID=SUBSTR(MSG. 1, LOC+3, 4)
    CMD=PARTID || SUBSTR(CMD, 5+LENGTH(JOB. I), 40)
  END
  SAY CMD
  CMD
  RC = GETMSG(MSG. , 'RESP' , , , 5)
  I = 1
  DO WHILE I <= MSG. 0
    SAY MSG. I
    I=I+1
  END
  IF SUBSTR(CMD, 1, 3) = "MSG" THEN DO
    CALL SLEEP 5
    "REPLID"
    RC=GETMSG(MSG. , 'RESP' , , , 5)
    LOC=POS(SUBSTR(CMD, 5, 2), MSG. 1)
    PARTID=SUBSTR(MSG. 1, LOC+3, 4)

```

```

                                vseconx
                                SAY PARTID
                                END
                                END
END
COM1=' DEACTIVATE' CONNAME
COM1                                /* DEACTIVATE THE CONSOLE */
EXIT
/+
/*
/&

```

```

// JOB VSECON
// LIBDEF PROC, SEARCH=VENDOR. REXX
// EXEC REXX=VSECONX
MSG PDF
XXXX SHUTDOWN
MSG TCPIP
XXXX SHUTDOWN
SLEEP 5
XXXX YES
RSP JCLSCHEDE &TERMINATE
MSG FAQSVTAM
XXXX YES
MSG FAQSMAIN
XXXX YES
MSG FAQSCONS
XXXX YES
MSG FTPJCL, DATA=END
MSG CICSICCF
XXXX CEMT P SHUT, I
MSG CICSSTEST
XXXX CEMT P SHUT, I
MSG CICSPROD
XXXX CEMT P SHUT, I
Z NET, QUICK
PEND
/*
/&

```

# ADDRESS POWER

- ◆ Provides for VSE/POWER spool-access services requests, GET, PUT, and CTL

# ADDRESS POWER

```
myj ob. 0 = 9
myj ob. 1 = "* $$ JOB JNM=MYJOB1, CLASS=Y, PRI =8, DI SP=D"
myj ob. 2 = "* $$ LST CLASS=Q, DI SP=D"
myj ob. 3 = "// JOB MYJOB1"
myj ob. 4 = "// EXEC LI BR"
myj ob. 5 = "ACC S=PRD1. BASE"
myj ob. 6 = "LD ARX*. PHASE"
myj ob. 7 = "/*"
myj ob. 8 = "/&"
myj ob. 9 = "* $$ EOJ"
ADDRESS POWER
' PUTQE RDR STEM MYJOB. '
```

# PRI NTPDF

```
// JOB PRI NTPDF
// EXEC LIBR
A S=VENDOR. REXX
CATALOG PRI NTPDF. PROC      EOD=/+      REPLACE=YES
/* AUTO EMAIL PDF REPORTS */
JCL. 0=23
JCL. 1="* $$ JOB USER='CRR', DISP=D, CLASS=S, PRI =5, JNM=EMAILPDF"
JCL. 2="* $$ LST CLASS=X, DISP=D, FNO=DEL, FCB=DEL"
JCL. 3='// JOB EMAILPDF'
JCL. 4='// LIBDEF *, SEARCH=VENDOR. TCPIP'
JCL. 5='// EXEC EMAIL, SIZE=EMAIL'
JCL. 6='EXEC EMAILLOG. L'
JCL. 7='SET FROM=CRICE@CAROLINASTEEL.COM'
JCL. 8='SET TO='
JCL. 9='SET SUBJECT=REPORT '
JCL. 10='SET PDF='
JCL. 11='SET CC=ON'
JCL. 12='SET DISP=DELETE'
JCL. 13='SET NOEJECT=OFF'
JCL. 14='TEXT'
JCL. 15='ATTACHED IS YOUR REPORT IN ADOBE PDF FORMAT.'
JCL. 16=' '
JCL. 17=' /+'
JCL. 18='CD POWER. LST. A'
JCL. 19='SEND '
JCL. 20='QUIT'
JCL. 21='/*'
JCL. 22='/&'
JCL. 23='* $$ EOJ'
RESULT = OUTTRAP(PWR.)
ADDRESS POWER 'PDISPLAY LST, RJE, 200, FULL=YES'
IF RC = 0 THEN
  DO I = 2 TO PWR. 0 BY 2
    J=I+1
    PARSE VAR PWR. I . JNAME JNUM PRI DISP CLASS PAGES D2 FORM .
    PARSE VAR PWR. J DATE1 USER .
    DATE1=SUBSTR(DATE1, 3, 10)
    USER=SUBSTR(USER, 4, 3)
    JCL. 16=USER||'| '||FORM
    JCL. 8=LEFT(JCL. 8, 7)||'| CRICE@CAROLINASTEEL.COM'
    IF USER='BCS' THEN JCL. 8=LEFT(JCL. 8, 7)||'| BSMITH@CAROLINASTEEL.COM'
    IF USER='BHJ' THEN JCL. 8=LEFT(JCL. 8, 7)||'| BJOHNSON@CAROLINASTEEL.COM'

    JCL. 9=LEFT(JCL. 9, 19)||JNAME
    JCL. 10=LEFT(JCL. 10, 8)||'| PDFSETUP'
    IF FORM='C1P8' THEN JCL. 10=LEFT(JCL. 10, 8)||'| PDFPORT'

    IF FORM='W1P8' THEN JCL. 10=LEFT(JCL. 10, 8)||'| PDFW1P8'
    IF FORM='WT1P' THEN JCL. 10=LEFT(JCL. 10, 8)||'| PDFW1P8'
    IF FORM='ST1P' THEN JCL. 13='SET NOEJECT=ON'
    JCL. 18=LEFT(JCL. 18, 13)||CLASS
    DATE2=SUBSTR(DATE1, 7, 4)||'| _ '||SUBSTR(DATE1, 1, 2)||'| _ '||SUBSTR(DATE1, 4, 2)
    SAY DATE2
    JNAME1 = JNAME||'| . '||JNUM
    JCL. 19=LEFT(JCL. 19, 5)||JNAME1||'| AS '||JNAME||'| - '||DATE2||'| . PDF'
    IF DISP='D' THEN DO
      IF PAGES > 0 THEN DO
        ADDRESS POWER PUTQE RDR STEM JCL. WAIT 0
      END
    END
  END
END
EXIT
/+
/*
/&
```



# Batch Services

- ◆ LINK/LINKPGM commands environment allows calls/loads VSE batch utilities/ programs:
  - LIBR
  - IDCAMS
  - DITTO
  - ASSEMBLER/LNKEDT
  - MSHP
  - your own VSE batch programs, etc.

# Batch Services

```
/* A REXX invocation of IDCAMS                                */
CALL OUTTRAP output.    /* Open output...          */
CALL REXXIPT sysipt.    /*input data stream    */
sysipt.0 = 1            /* number of commands */
sysipt.1 = 'LISTCAT CLUSTER' /* command            */
ADDRESS LINK 'IDCAMS MARGINS(1 80)'
IF RC > 4 Then
Do .....              /* analyze output.    */
EXIT4
```

# EXECIO Command

- ◆ Controls the input and output of information to and from a file.
- ◆ Supported operations are DISKR, DISKW, and DISKRU.
- ◆ Can read VSE library members, SYSIPT data, or SAM files.
- ◆ Can write VSE library members, SYSLST, or SAM files.

# TCP/IP Functions

- ◆ CGI interface to create dynamic HTML
- ◆ Socket interface
  - CSI socket interface is quite simple but not operating system independent
  - IBM socket interface is more complicated, but is the compatible with VM and OS/390

```

// JOB CATAL
// EXEC LIBR
A S=VENDOR.TCPI P
CATALOG VSECOM.PROC          EOD=/+          REPLACE=YES
/*                               */
/* GET THE PASSED PARAMETERS   */
/*                               */
/* ----- */
USERID=ARG(1)
PASSWORD=ARG(2)
DATA=ARG(3)
PARSE UPPER VAR DATA REQUEST 9 COMMAND
INLEN=LENGTH(COMMAND)
IF INLEN=0 THEN DO
  ADDRESS CONSOLE
  'ACTIVATE NAME CGI CONR PROFILE REXNORC'
  'CART USCHI'
  'REDI 30L'
  RC = GETMSG(MSG. , 'RESP' , 'USCHI' , , 5)
  X=HTML(' <HTML><HEAD><TITLE>' )
  X=HTML(' VSE CONSOLE COMMAND PROCESSOR' )
  X=HTML(' </TITLE></HEAD>' )
  X=HTML(' <BODY TEXT="#993300" BGCOLOR="#66FF99">' )
  X=HTML(' <CENTER><H2><B><I><FONT COLOR="#000000">' )
  X=HTML(' VSE CONSOLE COMMAND PROCESSOR' )
  X=HTML(' </FONT></I></B></H2></CENTER><P><HR>' )
  X=HTML(' <FORM METHOD=GET ACTION="VSECOM">' )
  X=HTML(' INPUT: ' )
  X=HTML(' <INPUT TYPE="TEXT" NAME="COMMAND" SIZE%>' )
  X=HTML(' <BR><HR>' )
  X=HTML(' <FONT COLOR="#000066"><PRE>' )
  I = MSG.0
  DO WHILE I > 0
    X=HTML(MSG. I )
    I=I-1
  END
  X=HTML(' </BODY></HTML>' )
  ADDRESS CONSOLE 'DEACTIVATE CGI CONR'
  EXIT
END
/*                               */
/* ACTIVATE THE CONSOLE INTERFACE */
/*                               */
ADDRESS CONSOLE
'ACTIVATE NAME CGI CONR PROFILE REXNORC'
'CART USCHI'
/*                               */
/* PASS THE COMMAND TO VSE        */
/*                               */
COMMAND
RC = GETMSG(MSG. , 'RESP' , 'USCHI' , , 5)
/*                               */
/* RETURN THE WEB PAGE HEADINGS   */
/*                               */
X=HTML(' <HTML><HEAD><TITLE>' )
X=HTML(' VSE CONSOLE COMMAND PROCESSOR' )
X=HTML(' </TITLE></HEAD>' )
X=HTML(' <BODY TEXT="#993300" BGCOLOR="#66FF99">' )
X=HTML(' <CENTER><H2><B><I><FONT COLOR="#000000">' )
X=HTML(' VSE CONSOLE COMMAND PROCESSOR' )
X=HTML(' </FONT></I></B></H2></CENTER><P><HR>' )
X=HTML(' <FORM METHOD=GET ACTION="VSECOM">' )
X=HTML(' INPUT: ' )

```

```

                                vsecom
X=HTML(' <INPUT TYPE="TEXT" NAME="COMMAND" SIZE%>' )
X=HTML(' <BR><HR>' )
X=HTML(' <FONT COLOR="#000066"><PRE>' )
/*                               */
/* INSERT THE CONSOLE RESPONSE */
/*                               */
I = 1
DO WHILE I <= MSG. 0
  X=HTML(MSG. I )
  I=I +1
END
/*                               */
/* INSERT THE WEB FOOTER       */
/*                               */
X=HTML(' </BODY></HTML>' )
/*                               */
/* DEACTIVATE THE CONSOLE AND EXIT */
/*                               */
ADDRESS CONSOLE ' DEACTIVATE CGI CONR'
EXIT
/+
/*
/&

```

```

DEFINE HTTPD, ID=WEB, ROOT=' PRD2. TCPI P'
DEFINE CGI , PUBLIC=VSECOM, TYPE=CGI -REXX

```

```

HTTP: //100. 100. 100. 10/VSECOM

```

vsepwr

```
// JOB VSEPWR
// EXEC LIBR
A S=VENDOR. TCPI P
CATALOG VSEPWR. PROC          EOD=/+          REPLACE=YES
/*                               */
/* GET THE PASSED PARAMETERS   */
/*                               */
/* ----- */
USERID=ARG(1)
PASSWORD=ARG(2)
DATA=ARG(3)
INLEN=LENGTH(DATA)
/*                               */
/* RETURN THE WEB PAGE HEADINGS */
/*                               */
X=HTML(' <HTML><HEAD><TITLE>' )
X=HTML(' VSE POWER LST QUEUE DISPLAY' )
X=HTML(' </TITLE></HEAD>' )
X=HTML(' <BODY TEXT="#993300" BGCOLOR="#66FF99">' )
X=HTML(' <CENTER><H2><B><I><FONT COLOR="#000000">' )
X=HTML(' VSE POWER LST QUEUE DISPLAY' )
X=HTML(' </FONT></I></B></H2></CENTER><P><HR>' )
X=HTML(' <BR><HR>' )
X=HTML(' <FONT COLOR="#000066"><PRE>' )
/*                               */
/* INSERT THE LST QUEUE        */
/*                               */
CALL OUTTRAP OUT.
ADDRESS POWER
' SETUID SYSTCPI P'
' DISPLAY LST, FULL=YES'
IF OUT.0 > 1 THEN DO
  DO I = 2 TO OUT.0 BY 2
    J = I + 1
    PARSE VAR OUT. I MSG REST1
    PARSE VAR OUT. I MSG JOB JOBNUM C1 C2 CLASS C3 C4 FORM C5
    PARSE VAR OUT. J REST2
    PDF = '?CSI_PDF=PDFSETUP'
    IF FORM = 'C1P8' THEN PDF = '?CSI_PDF=PDFPORT'
    IF FORM = 'W1P8' THEN PDF = '?CSI_PDF=PDFW1P8'
    LINK=' POWER/LST/' || CLASS || '/' || JOB || '/' || JOBNUM || PDF
    X=HTML(' <A HREF="' || LINK || '">' )
    X=HTML(REST1 || REST2 || ' </A>' )
  END
END
/*                               */
/* INSERT THE WEB FOOTER      */
/*                               */
X=HTML(' </BODY></HTML>' )
EXIT
/*
/
/
```

DEFINE HTTPD, ID=WEB, ROOT=' PRD2. TCPI P'

DEFINE CGI, PUBLIC=VSEPWR, TYPE=CGI -REXX

HTTP: //100. 100. 100. 10/VSEPWR

ftpjcl.txt

```
// JOB FTPJCL
// EXEC LIBR
A S=VENDOR.REXX
CATALOG FTPJCL.PROC          EOD=/+          REPLACE=YES
X=ASSGN(' STDOUT' , ' SYSLST' )
FIP=' 166. 82. 131. 204'
SYSID=' 00'
TIMEOUT=900
X=SOCKET(' FTP' , ' OPEN' , 21, FIP, 21, SYSID, TIMEOUT)
IF X \= 0 THEN DO
    SAY ERRMSG
    EXIT
END
CALL GETLINES
COMMAND=' CRI CE'
CALL SEND
CALL GETLINES
COMMAND=' DEEPSPRG'
CALL SEND
CALL GETLINES
COMMAND=' CRI CE'
CALL SEND
CALL GETLINES
COMMAND=' DEEPSPRG'
CALL SEND
CALL GETLINES
COMMAND=' LCD POWER. RDR'
CALL SEND
CALL GETLINES
COMMAND=' CD R: \UPLOAD'
CALL SEND
CALL GETLINES
FITAB. 0=0
COMMAND=' DIR *. JCL'
CALL SEND
CALL GETLINES
SAY FITAB. 0
IF FITAB. 0 > 0 THEN DO
    DO I = 1 TO FITAB. 0
        J = I - 1
        JOBS. J = FITAB. I
        JOBS. 0 = J
    END
    DO J = 1 TO JOBS. 0
        COMMAND = ' GET ' || JOBS. J
        CALL SEND
        CALL GETLINES
        COMMAND = ' DEL ' || JOBS. J
        CALL SEND
        CALL GETLINES
    END
END
COMMAND=' QUI T'
CALL SEND
CALL GETLINES
X=SOCKET(HANDLE, ' CLOSE' )
EXIT
SEND:
X=SOCKET(HANDLE, ' SEND' , COMMAND)
IF X \= 0 THEN DO
    SAY ERRMSG
    X=SOCKET(HANDLE, ' ABORT' )
    X=SOCKET(HANDLE, ' CLOSE' )
```



# ftpjcl.txt

```

      EXIT
END
RETURN
receive:                                /* Start of routine                */
x = SOCKET(handle, 'RECEIVE', timeout) /* If it failed...                */
if x \= 0 then do                       /* Tell us what went wrong        */
    say errmsg
    X=SOCKET(HANDLE, 'ABORT')
    X=SOCKET(HANDLE, 'CLOSE')
    exit
end                                     /* And terminate the program      */
return                                /* Otherwise...                   */
return                                /* Return to caller               */
getlines:                             /* Start of routine                */
do forever                            /* Do until we find a match       */
    call receive                       /* Read another datagram          */
    say buffer                         /* Display it on SYSOUT           */
    loc=POS('Error: ', buffer)         /* See if a connection failed     */
    if loc > 0 then exit 20            /* If the connection failed, exit */
    loc=POS('Ready: ', buffer)         /* Look for another keyword       */
    if loc > 0 then return             /* And if we found it, it's okay  */
    loc=POS('Enter Foreign User ID', buffer)
    if loc > 0 then return
    loc=POS('Enter Foreign Password', buffer)
    if loc > 0 then return
    loc=POS('Enter Foreign Account', buffer)
    if loc > 0 then return
    loc=POS('Enter Local User ID', buffer)
    if loc > 0 then return
    loc=POS('Enter Local Password', buffer)
    if loc > 0 then return
    loc=POS('Enter Local Account', buffer)
    if loc > 0 then return
    loc=POS('Service closing control connection', buffer)
    if loc > 0 then return
    LOC=POS('.JCL', BUFFER)
    IF LOC > 0 THEN DO
        PARSE VAR BUFFER DATE TIME CNT FILENAME
        I = FITAB.0 + 1
        FITAB.I = FILENAME
        FITAB.0 = I
    END
end
/+
/*
/&

```