

156

VM

August 1999

In this issue

- 3 Enhancing the XEDIT ALL macro
- 6 Alternative VM system file display
- 11 VM:Secure enhancement rules –
part 5
- 17 Viewing the VSE lock file from
CMS
- 30 A full screen console interface –
part 13
- 49 September 1994 – August 1999
index
- 52 VM news

© Xephon plc 1999

update

VM Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38030
From USA: 01144 1635 38030
E-mail: info@xephon.com

North American office

Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75077-2150
USA
Telephone: 940 455 7050

Editorial panel

Articles published in *VM Update* are reviewed by our panel of experts. Members of the panel include John Illingworth (UK), Reinhard Meyer (Germany), Philippe Taymans (Belgium), Romney White (USA), Martin Wicks (UK), and Jim Vincent (USA).

Subscriptions and back-issues

A year's subscription to *VM Update*, comprising twelve monthly issues, costs £180.00 in the UK; \$275.00 in the USA and Canada; £186.00 in Europe; £192.00 in Australasia and Japan; and £190.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1990 issue, are available separately to subscribers for £16.00 (\$23.00) each including postage.

Editor

Robert Burgess

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, EXECs, and other contents of this journal before making any use of it.

VM Update on-line

Code from *VM Update* can be downloaded from our Web site at <http://www.xephon.com/vmupdate.html>; you will need the user-id shown on your address label.

Contributions

Articles published in *VM Update* are paid for at the rate of £170 (\$250) per 1000 words for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

© Xephon plc 1999. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.

Enhancing the XEDIT ALL macro

The following two macros, SHOW and HIDE, were written to extend the capabilities of the standard XEDIT ALL macro.

ALL only allows one target to be specified, and does not work over a previous selection. SHOW is identical to ALL, but with multiple target capability, up to 20. Furthermore, you can add the keyword ALSO to act upon the current view of the file. HIDE is the logical complement of SHOW. It hides the lines that contain the specified targets.

Without arguments, both SHOW and HIDE restore the entire view of the file.

Here are some examples:

```
====> show/if/then/else/do/end
====> show also/select/when/otherwise
====> hide/move/compute/add/subtract
====> hide also/divide/multiply
```

SHOW

```
/*=====*/
/* SHOW macro for XEDIT */
/* This macro displays all lines containing the selected targets. */
/* Up to 20 targets can be specified simultaneously. */
/* Example: SHOW/Williams/Spielberg/Friedkin/Ford/ */
/* SHOW ALSO/Bergman/Fellini (does not destroy previous selection) */
/*=====*/

cmd = "command"
parse arg arg1
arg1 = strip(arg1)
if translate(left(arg1,4))="ALSO" then do
    also = 1
    arg1 = strip(substr(arg1,5))
end
sep = left(arg1,1)
if right(arg1,1) <> sep then arg1=arg1||sep
x = 2
do k = 1 to 20
    y = pos(sep,arg1,x)
    if y=0 | y=x then leave
```

```

    z = y-x
    target.k = substr(arg1,x,z)
    x = y+1
end
maxtarget = k-1
cmd "extract/line/msgmode/wrap/"
cmd "top"
cmd "set wrap off"
cmd "set msgmode off"
cmd "set scope all"
if also <> 1 then do
    cmd "set select 1 *"
end
do k = 1 to maxtarget
    cmd "top"
    do forever
        cmd "locate" sep||target.k
        if rc <> 0 then leave
        cmd "set select 0"
    end
end
if also <> 1 then do
    if maxtarget > 0 then do
        cmd "top"
        cmd "next"
    end
    else do
        cmd "set scope all "
        cmd "set select 0 *"
        ":" line.1
    end
end
end
else do
    ":" line.1
end
cmd "set display 0"
cmd "set scope display"
cmd "set wrap "wrap.1
cmd "set msgmode "msgmode.1
exit

```

HIDE

```

/*=====*/
/* HIDE macro for XEDIT */
/* This macro hides all lines containing the selected targets. */
/* Up to 20 targets can be specified simultaneously. */
/* Example: HIDE/Bartok/Schopenhauer/Rubinstein/Heidegger/Kant */
/* HIDE ALS0/Prokofiev (does not destroy previous selection) */

```

/*=====*/

```
cmd = "command"
parse arg arg1
arg1 = strip(arg1)
if translate(left(arg1,4))="ALSO" then do
    also = 1
    arg1 = strip(substr(arg1,5))
end
sep = left(arg1,1)
if right(arg1,1) <> sep then arg1=arg1||sep
x = 2
do k = 1 to 20
    y = pos(sep,arg1,x)
    if y=0 | y=x then leave
    z = y-x
    target.k = substr(arg1,x,z)
    x = y+1
end
maxtarget = k-1
cmd "extract/line/msgmode/wrap/"
first = line.1
cmd "top"
cmd "set wrap on"
cmd "set msgmode off"
if also <> 1 then do
    cmd "set scope all"
    cmd "set select 0 *"
end
cmd "set display 0"
cmd "set scope display"
do k = 1 to maxtarget
    cmd "top"
    do forever
        cmd "locate" sep||target.k
        if rc <> 0 then leave
        cmd "set select 1"
    end
end
do discover_line = first to 0 by -1 until rc=0
    ":" discover_line
end
cmd "set wrap "wrap.1
cmd "set msgmode "msgmode.1
exit
```

Luis Paulo Figueiredo Sousa Ribeiro
Systems Engineer
Edinfor (Portugal)

© Xephon 1999

Alternative VM system file display

The usage of VM system files can be obtained from the CP QALLOC command. This will present the system file usage of temporary disk, page, spool, and directory space used and allocated on each VM system disk.

The QALLOC EXEC given here essentially reformats this information, bringing together all the defined and allocated temporary disk, spool, page, and directory space from each VM system disk, and displays it under the respective system file type. This makes for a more useful display of system file usage.

Either summary, detailed, or help information will be displayed depending on the parameters passed to the EXEC. Examples of the output produced are shown for the detailed display in Figure 1, the summary display in Figure 2, and the help information in Figure 3.

The general format of the command is:

```
qalloc <parm>
```

where <parm> can be:

- 'summary' or ' ' – this will provide a summary usage display.
- 'detail' – this will provide a detailed usage display.
- '?' or 'help' – this will display help information.

QALLOC was written and tested under VM/ESA 2.2.0 service level 9704.

SOURCE CODE

```
/*-----*/
/*          QALLOC          */
/*          _____          */
/* Description: VM System File usage display          */
/*-----*/
/* Purpose  : To display VM system file usage in a user-friendly way          */
/* Input    : <parm> - ' ' will give summary display          */
/*          :          - 'SUMMARY' will give summary display          */
```

SYSTEM FILE ALLOCATION DETAIL ANALYSIS							
Volume	Real Dev	Start Page	End Page	Total Pages	Pages In-Use	High Page	% Used
Spool Area							
VMSRES	0310	708	1207	90000	18600	71610	20%
VM0001	0312	900	999	18000	8216	18000	45%
VM0002	0313	900	999	18000	6017	17726	33%
Summary				126000	32833		26%
Page Area							
VMSRES	0310	258	707	81000	14142	47265	17%
VM0001	0312	1000	1179	32400	10411	31680	32%
VM0002	0313	1000	1179	32400	13247	32398	40%
Summary				145800	37800		25%
Temporary Disk							
VMSRES	0310	25	223	199	0	0	0%
VM0001	0312	1180	1239	60	20	20	33%
VM0002	0313	1180	1239	60	0	0	0%
Summary				319	20		6% CKD
Directory							
VMSRES	0310	1	17	17	2	4	11% ACTIVE
Summary				17	2		11% CKD

Figure 1: Detailed system file usage

```

/*      :      - 'DETAIL' will give a detailed display      */
/*      :      - 'HELP' will display help information        */
/*      :      - '?' will display help information            */
/* Output : <screen display>                                  */
/* Routines: None                                             */
/*-----*/
/*                               Change History                */
/*-----*/
/* Date | Who | Description                                     */
/*-----*/

```

SYSTEM FILE ALLOCATION SUMMARY			
	Total Pages	Pages In-Use	% Used
Spool	126000	32833	26%
Page	145800	37800	25%
TDsk	319	20	6% CKD
Direct	17	2	11% CKD

Figure 2: Summary system file usage display

```

/* 22/3 | ICE | Initial version of program */
/*-----*/
arg parm .
upper parm

select
  when parm = '' then parm = 'SUMMARY'
  when parm = 'SUMMARY' then nop
  when parm = 'DETAIL' then nop
  when parm = '?' then call HELP
  when parm = 'HELP' then call HELP
  otherwise call HELP
end

'EXECIO * CP (STEM CP1. STRING Q ALLOC SPOOL'
'EXECIO * CP (STEM CP2. STRING Q ALLOC PAGE '
'EXECIO * CP (STEM CP3. STRING Q ALLOC TDISK'

```

System File Allocation and Usage Display.

Format of command is:

QALLOC <parm> where <parm> is blank will give a summary analysis
 is summary will give a summary analysis
 is detail will give a detail analysis
 is help or ? will display this screen
 anything else will display this screen

Figure 3: QALLOC help information display


```

'EXECIO * CP (STEM CP4. STRING Q ALLOC DRCT '

'VMFCLEAR'

do ix = 1 to cp1.0
  cp1.ix = 'SPOOL '||cp1.ix
end
do ix = 1 to cp2.0
  cp2.ix = 'PAGE '||cp2.ix
end
do ix = 1 to cp3.0
  cp3.ix = 'TDISK '||cp3.ix
end
do ix = 1 to cp4.0
  cp4.ix = 'DRCT '||cp4.ix
end
select
  when parm = 'SUMMARY' then call SUMMARY
  otherwise call DETAIL
end
exit
/* ----- */
SUMMARY:
/* ----- */
  say copies(' ',15) 'SYSTEM FILE ALLOCATION SUMMARY'
  say copies(' ',25) ' Total Pages          % '
  say copies(' ',25) ' Pages In-Use          Used'
  say copies('-',55)

  ix = cp1.0
  parse var cp1.ix . testit rest
  if testit = 'SUMMARY' then say 'Spool  'rest
  ix = cp2.0
  parse var cp2.ix . testit rest
  if testit = 'SUMMARY' then say 'Page   'rest
  ix = cp3.0
  parse var cp3.ix . testit rest
  if testit = 'SUMMARY' then say 'TDsk   'rest
  ix = cp4.0
  parse var cp4.ix . . nump usep rest
  say 'Direct          'right('  'nump,3) ' ',
      right('  'usep,3) rest
  say copies('-',55)

return

/* ----- */
DETAIL:
/* ----- */
  say '          SYSTEM FILE ALLOCATION DETAIL ANALYSIS'

```

```

say '          Real  Start   End  Total  Pages  High  % '
say 'Volume  Dev   Page   Page  Pages  In-Use  Page Used'
say copies('-',55)

say 'Spool Area'
do ix = 4 to cp1.0
  parse var cp1.ix . testit rest
  call SUMTEST
  say testit rest
end

say ''
say 'Page Area'
do ix = 4 to cp2.0
  parse var cp2.ix . testit rest
  call SUMTEST
  say testit rest
end

say ''
say 'Temporary Disk'
do ix = 4 to cp3.0
  parse var cp3.ix . testit rest
  call SUMTEST
  say testit rest
end

say ''
say 'Directory'
do ix = 4 to cp4.0
  parse var cp4.ix . testit rest
  call SUMTEST
  say testit rest
end

return

/* _____ */
SUMTEST:
/* _____ */
select
when testit = 'SUMMARY' then testit = 'Summary'
when testit = '—'   then do
    testit = copies('-',51)
    rest = ''
end

otherwise nop
end
return

```

```

/* _____ */
HELP:
/* _____ */
'VMFCLEAR'
say '      System File Allocation and Usage Display. '
say '      ===== '
say 'Format of command is: '
say 'QALLOC <parm>   where <parm> is blank will give a summary analysis'
say '                  is summary will give a summary
analysis'
say '                  is detail will give a detail analysis'
say '                  is help or ? will display this screen'
say '                  anything else will display this
screen'
say ''
exit 0
return

```

Ian Evans
Systems Programmer (UK)

© Xephon 1999

VM:Secure enhancement rules – part 5

This month we conclude the article providing special macros that enhance VM:Secure Rules to allow additional resource access control.

OBJLOG VMSECURE

```

/* Log a requested validation */
/* NW */
'TRANSFER OUTPUT SYSID USERID AUDT'
Pull output sysid user audt .
Call Trace output
auditfile = 'OBJECTS AUDIT' audt
Arg action tofn toft tofm . '(' keepopt .

If action = 'EXTRACT' Then Do
  'TEST PROCESS AUTHORIZ $OBJLOG EXTRACT'
  If rc = 0 Then Exit -1
  'TEST CMS FINIS' auditfile
  'TEST CMS STATE' auditfile
  If rc = 0 Then Do
    'TEST FORMAT MSG 8003E The Audit OBJECTS'

```

```

Exit 28
End
If tofn = '' Then
  Parse Value 'OBJECTS AUDIT A' With tofn toft tofm
Else If toft = '' Then
  Parse Value 'AUDIT A' With toft tofm
Else If tofm = '' Then tofm = 'A'
'TEST USER COPYTO' auditfile tofn toft tofm
crc = rc
If crc ≠ 0 Then Do
  'TEST FORMAT MSG 099I OBJLOG'
  Exit 100
End
If ¬Abbrev('NOERASE',keepopt,1) Then
  'TEST CMS ERASE' auditfile
'TEST USER MSG Objects Audit file extracted to' tofn toft tofm
End
Else Exit -1
Exit 0

```

OBJLOG VMSECURE

```

/* Log a requested validation */
/* NW */
'TRANSFER OUTPUT SYSID USERID AUDT'
Pull output sysid user audt .
Call Trace output
auditfile = 'OBJECTS AUDIT' audt
Arg action tofn toft tofm . '(' keepopt .

If action = 'EXTRACT' Then Do
  'TEST PROCESS AUTHORIZ $OBJLOG EXTRACT'
  If rc ≠ 0 Then Exit -1
  'TEST CMS FINIS' auditfile
  'TEST CMS STATE' auditfile
  If rc ≠ 0 Then Do
    'TEST FORMAT MSG 8003E The Audit OBJECTS'
    Exit 28
  End
If tofn = '' Then
  Parse Value 'OBJECTS AUDIT A' With tofn toft tofm
Else If toft = '' Then
  Parse Value 'AUDIT A' With toft tofm
Else If tofm = '' Then tofm = 'A'
'TEST USER COPYTO' auditfile tofn toft tofm
crc = rc
If crc ≠ 0 Then Do
  'TEST FORMAT MSG 099I OBJLOG'
  Exit 100

```

```

End
If ¬Abbrev('NOERASE',keepopt,1) Then
  'TEST CMS ERASE' auditfile
  'TEST USER MSG Objects Audit file extracted to' tofn toft tofm
End
Else Exit -1
Exit Ø

```

OBJMSG VMSECURE

```

/* get the msgs file - not for general use*/
/* NW */

'TRANSFER OUTPUT SYSID USERID'
Pull output sysid user
Call Trace output

'TEST USER COPYFROM VMSECURE NEWMSG A VMSECURE NEWMSG A'
If rc ¬= Ø Then Do
  Exit 1ØØ3
End
Exit

```

OBJQUERY VMSECURE

```

/* Map the usage of all USER OBJECT files */
/* NW */

'TRANSFER OUTPUT SYSID USERID'
Pull output sysid user
Call Trace output
Arg what . '(' opt . 1 orig . '('
If Length(what) > 1 & Right(what,1) = '*' Then Do
  wildsearch = 'FIND' Left(what,Length(what)-1)'|'
  what = '*'
End
Else wildsearch = ''
If what = '' Then what = '*'
If Abbrev('USED',opt,1) Then
  usedsearch = 'NLOCATE 28.2 / Ø/|'
Else usedsearch = ''

'TEST PROCESS AUTHORIZ $OBJQUERY' what
If rc ¬= Ø Then Exit -1

'TEST CMS PIPE COMMAND EXECMAP' what 'USEROBJ |',
'DROP 1 |',
wildsearch,

```

```

usedsearch,
'SPECS 1.8 1 /-/ N W 3 NW |',
'PAD 16 |',
'JOIN 3 / / |',
'STEM OUT.'

If rc = 0 Then Do
'TEST PROCESS SWITCH'
If out.0 > 0 Then Do i = 1 to out.0
'TEST USER MSG' out.i
If i//10 = 0 Then 'TEST PROCESS SWITCH'
End
Else If usedsearch = '' Then Do
'TEST FORMAT EMSG 8023E' orig
Exit 20
End
Else Do
'TEST FORMAT EMSG 8024E' orig
Exit 22
End
Exit 0
End
'TEST FORMAT EMSG 8003E The Objects' what
Exit 28

```

OBJSETUP VMSECURE

```

/* Load or Unload (erase) USER OBJECT files */
/* NW */

'TRANSFER OUTPUT SYSID USERID'
Pull output sysid user
Call Trace output

If user ≠ sysid Then Exit -1      /* Only SVM allowed to issue */

Arg objcuu objmode default .

'TEST CMS EXECDROP * USEROBJ'
'TEST CMS EXECDROP * RULEDEF'

'TEST CMS ACCESS' objcuu objmode
erc = rc
If rc ≠ 0 Then Do
'TEST FORMAT EMSG 113E' rc objcuu
'TEST USER LOGOP VMXOBJRULES VM:Secure OBJECT RULES did NOT activate.'
Exit erc
End
Else 'TEST FORMAT EMSG 8000I' objcuu objmode

```

```

'TEST EXEC OBJLOCK' objmode

settings = "/**/
;objcuu=""objcuu"";objmode=""objmode"";objdefault=""default""
workfile = 'OBJECT SETTINGS' objmode

'TEST CMS PIPE(name OBJDLOAD)|',
  'VAR SETTINGS |',
  '>' workfile

'TEST CMS EXECDROP OBJECT SETTINGS'
'TEST CMS EXECLOAD' workfile
erc = rc
If erc ≠ 0 Then Do
  'TEST FORMAT MSG 8005E' erc workfile
  Exit erc
End

'TEST CMS ERASE' workfile
'TEST FORMAT MSG 7001I'
Exit 0

```

OBJSTART VMSECURE

```

/* Check the access allowed for a particular user and OBJECT */
/* NW */

'TRANSFER OUTPUT SYSID USERID'
Pull output sysid user

Call Trace output
If user ≠ sysid Then Exit -1      /* Only SVM can execute */

Arg objcuu objmode default . '(' loadopt .

If WordPos(default,'ACCEPT REJECT') = 0 Then Do
  'TEST FORMAT MSG 8008E parameter' default
  'TEST EXEC OBJEND'
End

'TEST CMS VMFCLEAR'
'TEST USER LOGOP VMXOBJRULES Beginning initialization of OBJECT RULES.'
'TEST EXEC OBJSETUP' objcuu objmode default
If rc ≠ 0 Then 'TEST EXEC END'

'TEST EXEC OBJDLOAD *'
If rc ≠ 0 Then 'TEST EXEC END'
'TEST USER LOGOP VMXOBJRULES Object definitions have been loaded.'

```

```
'TEST USER LOGOP VMXOBJRULES Beginning load of USER objects.'
'TEST EXEC OBJLOAD * (' loadopt
Say rc
If rc  $\neq$  0 Then 'TEST EXEC END'

'TEST EXEC OBJLOCK (CLEAR'
'TEST USER LOGOP VMXOBJRULES Initialization of OBJECT RULES complete.'
Exit 0
```

SYSTEM VMSECURE MACRO

Update your local copy of the SYSTEM VMSECURE macro to start Object Rules. You can either copy the SYSTEM VMSECURE file to the 191 disk and update it, or use the standard VMRMANT procedures for applying a local modification to this macro.

You will need to add one line to the macro:

```
'TEST EXEC OBJSTART cuu mode default ( loadopt'
```

Where:

- 'cuu' is the address of the Object disk.
- 'mode' is any open mode the disk can be accessed as.
- 'default' is ACCEPT or REJECT.

The 'loadopt' can be FAST to bypass the processing done to validate the object rules content in each user object file. The FAST load option should be used only if you are sure that the content of the user object files are correctly coded.

For example:

```
'TEST EXEC OBJSTART 1B4 R ACCEPT ( FAST'
```

This line of code should be placed before the 'call housekeeping' line in the SYSTEM VMSECURE macro.

James S Vincent
Software Specialist
Nationwide Insurance (USA)

© Nationwide Insurance 1999

Viewing the VSE lock file from CMS

VM/ESA2.2 has enabled users with VSE virtual machines all running under the one VM/ESA system to share a lock file on a V disk (FBA architecture only). This gives vastly improved access time and should be the norm for such configurations.

One machine (not normally a VSE machine) has a directory entry similar to the following example:

```
MDISK 100 FB-512 V-DISK 1920 MWV R_PASS
```

Sharing machines will, of course, have a LINK defined. The owning machine must be AUTOLOGed before any VSE machines, initialize the volume with ICKDSF, and remain logged on but disconnected.

Several years ago I had an article, *Displaying the VSE lock file*, published in *VSE Update*, Issue 8, December 1992, showing how to view the VSE lock file from CMS, but it would only work for CKD devices. The following EXECs and Assembler program will read an FBA device and show you all resources under XEDIT.

The files included are:

- VSELOCK EXEC – front end command.
- XLOCK XEDIT – performs most of the logic.
- RESNAMES HELPCMS – tells you what the resources mean.
- FLOCK ASSEMBLE – reads the lock file.

VSELOCK EXEC

```
/*****  
* Read and view VSE lock file *  
*****/
```

```
address COMMAND 'XEDIT $$VSE$$ $$LOCK$$ A3 (PROFILE XLOCK) INIT'  
exit rc
```

XLOCK XEDIT

```
/******  
* Read and display VSE lock file          *  
* Called as XEDIT profile by VSELOCK EXEC *  
* Can also be called to refresh display   *  
* or flip-flop VSAM resource display     *  
*****/  
  
arg comm parm . ')' init .  
  
call xprofile  
if init='INIT' then  
do  
    'SET SIDCODE NOVSAM'    /* handy way to keep a setting */  
    call read_file         /* get all resource names      */  
    exit  
end  
  
select  
when comm='REFRESH' then  
do          /* re-read the lock file */  
    'SET MSGMODE OFF'  
    'ALL'  
    'TOP'  
    'DEL *'  
    'SET MSGMODE ON'  
    call read_file  
end  
when comm='VSAM' & parm='OFF' then  
do          /* don't show VSAM resources */  
    'TOP'  
    'ZONE 1 1'  
    'ALL -/V'                /* exclude resources starting "V" */  
    'ZONE 1 *'  
    'SET SIDCODE NOVSAM'    /* remember current setting */  
    'SET RESERVED -2 Y N' show_vsam  
    'SET PF9 XLOCK VSAM ON' /* PF9 will flip to VSAM   */  
    'SET LINEND ON'  
    'TOP'  
end  
when comm='VSAM' then  
do          /* show VSAM resources as well */  
    'TOP'  
    'ALL'  
    'SET SIDCODE VSAM'     /* remember current setting */  
    'SET RESERVED -2 Y N' show_novsam  
    'SET PF9 XLOCK VSAM OFF' /* PF9 will flip to NOVSAM */  
    'SET LINEND ON'  
end
```

```

        otherwise nop
end
exit

/*****
*           Main Processing           *
*           _____               *
* Read the Lock File and insert a line *
* for each locked resource found      *
*****/

read_file:
call xprofile      /* set XEDIT options for file */
                    /*_____ */
owner = 'VSELOCK' /* machine with lock file in directory entry */
cuu = '100'       /* virtual address in directory */
rpass = 'VSE'    /* read password */
                    /*_____ */

address COMMAND
call getlnk      /* link and access volume */
call extent 'DOS.LOCK.FILE' /* get extents */

firstrec = 1     /* first record not yet read */
lrec = 0        /* lock record number in a batch */
eof = 0        /* clear EOF indicator */
reclim = '99'   /* max. recs to read */
lrec.0 = '99'   /* counter for variables */

call 'FLOCK' 'LREC.',sblock,'1' /* read lock file descriptor */
rblock = sblock+1 /* block to read next */

rec1 = lrec.1    /* get file characteristics from descriptor record */
ncpus = c2d(substr(rec1,3,2)) /* sharing machines */
lent = c2d(substr(rec1,11,2)) /* entry length */

cpulist = substr(rec1,21,ncpus*8) /* all sharing cpu ids */

k = 0
do j=1 to ncpus*8 by 8 /* remember all ids */
    cpuid.k = substr(cpulist,j+2,6)
    cpuid.k = vmid(cpuid.k) /* convert to virtual machine id */
    k = k+1
end
                    /*_____
* Look at all blocks in lock file *
*****/

do forever
    lrec = readnext() /* get next block */
    if eof then leave

```

```

bents = c2d(substr(lrec,3,2)) /* entries in this block */
if bents>0 then

/* look at each resource entry */
do p=5 to 5+(bents-1)*lent by lent
  rname = substr(lrec,p,12) /* resource name */
  do k=0 to ncpus-1
    lflag = c2x(substr(lrec,p+12+k,1))

    if lflag='00' then /* some locking for this cpu */
                        /* add resource to file */
      address XEDIT 'I' ,
        left(rname,12) c2x(rname) left(cpuid.k,8) ' ' opt(lflag)
    end
  end
end
end

/*****
* Lock file has now been read *
* Detach links and complete XEDIT settings *
*****/
'RELEASE' lockfm
x = diag(8,'DETACH 100')
address XEDIT
'SET LINEND ON'
'TOP'
'SORT * A 1 12 39 46' /* sort by resource name */
'EXTRACT /SIDCODE/SIZE'
/* don't show VSAM resources if not wanted */
if sidcode.1='NOVSAM' then 'MACRO XLOCK VSAM OFF'
else 'MACRO XLOCK VSAM ON'
'SET RESERVED 1 H VSE lock file 'size.1 'entries'
return

/*****
* Return next record *
*****/

readnext:
lr = lr+1 /* next record */
if lr>lrec.0 | firstrec then
do /* next batch of blocks */
  if firstrec then
  do
    call 'FLOCK' 'LREC.',rblock,reclim
    lr = 1 /* reset counter */
  end
  else
  do

```

```

        if rblock>=eblock then eof = 1 /* last track has been passed */
        else
        do
            if rblock+reclim>eblock then reclim = eblock-rblock
            call 'FLOCK' 'LREC.',rblock,reclim
            lr = 1 /* reset counter */
        end
    end
    rblock = rblock+reclim /* next batch of records */
    firstrec = 0
end
return lrec.lr

/*****
* Return lock options separated into LOCKOPT and CONTROL *
*****/

opt:
arg lopt .
if left(lopt,1)='1' then control = 'E'

else control = 'S'
return right(lopt,1) 'control

/*****
* Work out virtual machine name from serial number *
* All virtual machines to be added here *
*****/

vmid:
parse arg id .
if id='FF1005639672'x then return 'VSEPROD'
if id='FF0305639672'x then return 'VSEICCF'
if id='FF2305639672'x then return 'VSETEST'
return c2x(left(id,4)) /* can't find this one, use cpuid */

/*****
* Make the link and access *
*****/

getlnk:
x = diag(8,'DET 100')
x = diag(8,'LINK' owner cuu '100 RR' rpass)
if rc=0 then exit rc
linkq = diag(8,'Q V 100') /* see what we've got */
if wordpos('BLK',linkq)=0 then
do
    say owner cuu 'is not an FBA device'
    exit 16
end

```

```

address COMMAND 'SET CMSTYPE HT'
address COMMAND 'GETFMADR'          /* find spare disk mode */
pull . lockfm . . .
address COMMAND 'AC 100' lockfm
address COMMAND 'SET CMSTYPE RT'
return

/*****
* Get lock file extent *
*****/

extent:
arg dsn
'PIPE CMS LISTDS' dsn lockfm '(EXTENT' ,
  '| STEM EXT.'
if ext.0=4 then
do
  say dsn 'not found on' owner cuu
  call detlink
  exit 16
end

  /* get start and end block numbers */
parse var ext.4 . . sblock eblock .
sblock = sblock+0          /* lose leading zeros */
eblock = eblock+0
return

/*****
* Set up reserved lines, PK keys etc. *
*****/

xprofile:
'SET SCALE OFF'
'SET PREFIX NULLS'
'SET TOFEOF OFF'
'SET RESERVED 3 P N' ,
'  <----- Resource name -----> Machine Lock Control'
'SET RESERVED 4 P N' ,
'  <- char -> <- hexadecimal -> name Option'
'SET CURLINE ON 5'
'SET CMDLINE B'
'SET SHADOW OFF'
'SET LINEND OFF'          /* suppress line editing characters */
'SET IMAGE OFF'
'SET PF1 HELP RESNAMES'
'SET PF2 XLOCK REFRESH'
'SET PF3 QQ'
'SET PF4 TOP#SORT * A 1 12 39 46'          /* sort by resource name */
'SET PF5 TOP#SORT * A 39 46 1 12'        /* sort by machine name */
'SET RESERVED -3 Y N' ,

```

```
'PF1:Help PF2:Refresh PF3:Quit PF4:Sort by resource PF5:- by machine
/* either of the following messages will be on line -2 */
show_vsam = ' PF9: Show VSAM resources'
show_novsam = ' PF9: Exclude VSAM resources'
return
```

RESNAMES HELPCMS

```
+-----+
| Common resource names |
| on VSE lock file      |
+-----+
```

Resource name	Meaning
---------------	---------

Avvvvvv...

The VTOC on volume "vvvvvv" is locked, probably during an OPEN. One machine waiting for a reply to an overlap message will lock out other OPENS to the same volume. This is a good reason not to have duplicate vol-ids on mini-disks used by VSE machines sharing a lock file if they are defined as SHR in the IPL procedure.

Cvvvvvv...

LIBR has locked a library on volume "vvvvvv" during an update.

IJQFL.vvvvvv

The POWER Q file on volume "vvvvvv" is locked, normally for only a couple of seconds, whilst one sharing system is updating or scanning it.

Vvvvvvv....

VSAM has locked something on volume "vvvvvv". These resource names will usually take up 99% of the lock file.

```
*****
* For full information on the rest of the resource *
* name see the appropriate logic manuals.          *
*****
```

```
+-----+
| Lock Options |
+-----+
```

LOCKOPT	CONTROL	Meaning
1	E	No other user is allowed concurrent use.
	S	Other 'S' users allowed concurrent use, but no concurrent 'E' user allowed.
2	E	No other 'E' user gets concurrent use, but other 'S' users allowed.
	S	Other 'S' users allowed and, in addition, one 'E' user is allowed.
4	E	No other 'E' user from another system is allowed. However, other 'S' users from other systems may have concurrent use.
	S	Other 'S' users and, in addition, one 'E' user from another system is allowed.

FLOCK ASSEMBLE

```

FLOC      TITLE 'Read from VSE lock file on 100 and return records'
*****
*
* FLOCK is called in the following way from REXX:
*
*      CALL FLOCK VAR,SBLOCK,NBLOCKS
*
*      returns a number of blocks starting at SBLOCK from
*      device address 100 and puts all records with lock
*      information into stem variable VAR.
*
*      NBLOCKS is how many blocks to read, but fewer will
*      be returned, since not all will contain information.
*
*      A maximum of 99 blocks may be read at once.
*
*      NB 'VAR' is a REXX variable name and must be in quotes.
*      To be more efficient 'FLOCK' should be in quotes.
*      These pairs of quotes must be separate:
*      eg 'FLOCK' 'LREC',startb,nrecs
*
* Macros used are DIAG, REGEQU, CMSSTOR (in DMSGPI)
*                  HCPSTGIOP           (in HCPGPI)
*
* Generate FLOCK MODULE by: GLOBAL MACLIB DMSGPI HCPGPI
*                           ASSEMBLE FLOCK
*                           LOAD FLOCK
*                           GENMOD FLOCK (ALL
*
*****

```



```

DEVADDR EQU X'0100'           device address
                                           SPACE
FLOCK   CSECT
        USING *,R12
        LR   R12,R15           set up base register
        B    PASTAMP
        DC   CL8'FLOCK'       eyecatcher
        DC   CL8'&SYSDATE'
        DC   CL8'&SYSTIME'
SAVRET  DC   F'0'             R14 on entry
PASTAMP DS   0H
        ST   R14,SAVRET
        LR   R13,R0           base for extended function PLIST
        USING EFPLIST,R13
        L    R1,EARGLIST
                                           SPACE
*           get variable name
        LM   R5,R6,0(R1)      variable name & length
        BCTR R6,R0            execute length
        EX   R6,MVVAR         move variable to our area
        LA   R6,VNAME+1(R6)   byte after variable name
        ST   R6,AVEND         save address
        MVC  SHVNAML,4(R1)    move length
        MVC  SAVNAML,4(R1)    save elsewhere
                                           SPACE
*           get block number
        LM   R5,R6,8(R1)      block no & length
        BCTR R6,R0            execute length
        EX   R6,PACKW         packed block
        CVB  R6,DWORK         hex block
        ST   R6,PHBLOCK      put into 'define extent' parm.
                                           SPACE
*           get no of records to read
        LM   R5,R6,16(R1)     record count & length
        BCTR R6,R0            execute length
        EX   R6,PACKW         packed count
        CVB  R6,DWORK         hex count
        STH  R6,LOGCOUNT+1   put into rec. counter for 'locate'
        ST   R6,RELEND        put into defined extent
        LR   R8,R6            remember for after read
                                           EJECT
*           *****
*           * read the requested number of records from DASD *
*           * the records will be returned in stem variables *
*           * if they contain any locking information *
*           *****
                                           SPACE
        LA   R2,DEVADDR       device address
        LA   R3,CCWS          read CCW chain
        SLA  R6,9             multiply by block size (512)
        STH  R6,READCCW+6     put into read CCW
        BAL  R10,GETSTOR      get some storage for I/O buffer

```

```

ST    R1,BUFFAD          save address of storage
STCM  R1,7,READCCW+1    and put into CCW
                                     SPACE
*
*   *****
*   * set up DIAG A8 parameter block *
*   * and read the disk             *
*   *****
LA    R7,DIAGBLK        parameter block
USING SGIOP,R7
STH   R2,SGIDEVNO      device number
ST    R3,SGICPA        CCW address
DIAG  R7,R0,X'A8'      do synchronous I/O
BNZ   ERROR            something wrong
                                     SPACE
*
*   *****
*   * Set a stem variable to every record *
*   * which contains locking information *
*   *****
RECOUT L    R7,BUFFAD      first record
EQU   *
NC    2(2,R7),2(R7)     if these bytes are zero then
BZ    NORET              there is nothing in the block
                                     SPACE
L     R9,CURSTEM        address of current stem no.
L     R3,AVEND           where to put it
MVC  0(R3,3),0(R9)     move ".nn" to variable name
L     R2,SAVNAML        get variable length
SR    R0,R0
IC    R0,2(R9)          length of n or nn
AR    R2,R0             add this to original length
LA    R1,512            record length
BAL  R10,RETVAR        return variable
                                     SPACE
NORET LA    R9,3(R9)      next stem no.
ST    R9,CURSTEM        remember it
EQU   *
LA    R7,512(R7)        next record
BCT  R8,RECOUT          look at all records
                                     SPACE
*
*   *****
*   * return number of variables in var.0 *
*   *****
L     R7,CURSTEM        next stem number address
BCTR  R7,R0             less 3 for
BCTR  R7,R0             highest one
BCTR  R7,R0             returned
AR    R2,R0             add this to original length
SR    R1,R1
IC    R1,2(R7)          length of n or nn
L     R2,SAVNAML        length of VAR.
LA    R2,1(R2)          length of VAR.0

```

```

L      R3,AVEND          where to put "0"
MVI   0(R3),C'0'        set up VAR.0
BAL   R10,RETVAR        how many variables returned
                                SPACE 2
*
*      *****
*      * Free storage and *
*      * exit to caller  *
*      *****
L      R1,BUFFAD         free I/O buffer
BAL   R10,FREESTOR      (length in R6)
L      R14,SAVRET        return
BR    R14
                                SPACE 5
*****
*      Error Routine      *
*      all errors will cause a null value to be returned *
*
*      R15 is negative for errors from EXECCOM *
*      R15 = 3 for wrong length record *
*      see manual for other i/o errors from DIAG X'A8' *
*****
SPACE
ERROR  DS      0H
SR    R1,R1          ensure null value returned
L     R2,SAVNAML     length of variable name
BAL   R10,RETVAR     to exec
L     R14,SAVRET     return to CMS
BR    R14
                                EJECT
*****
*      Get some storage   *
*
*      R6 = length in bytes to get *
*      R1 = address of storage obtained *
*      R10 = return address *
*****
SPACE
GETSTOR DS      0H
CMSSTOR OBTAIN,BYTES=(R6),BNDRY=PAGE
BR    R10
                                SPACE 5
*****
*      Release some storage *
*
*      R1 = address of storage to free *
*      R6 = length of storage *
*      R10 = return address *
*****
SPACE
FREESTOR DS      0H
LR    R2,R1

```

CMSSTOR RELEASE,BYTES=(R6),ADDR=(R2)
 BR R10

EJECT

```
*****
*          Return data into REXX variable          *
*                                                                 *
*          R7 = address of data                    *
*          R1 = length of data                    *
*          R10 = return address                   *
*          R2 = length of variable name in VNAME *
*****
```

SPACE

```
RETVAR  DS    0H
        ST    R7,SHVVALA      address of value
        ST    R1,SHVVALL     length of value
        ST    R2,SHVNAML     length of variable name
        LA    R0,RFPLIST     parameter list
        L     R1,ADNAME      set up to call 'EXECCOM'
        SVC   202
        DC    AL4(1)
        LTR   R15,R15        look at return code
        BM    ERROR         error if negative
        BR    R10           return OK
```

SPACE

```
AVEND   DC    A(0)          address of end of variable name
MVVAR   MVC   VNAME(0),0(R5)
VNAME   DC    CL16' '
COMMNAME DC   CL8'EXECCOMM'
ADNAME  DC    0A(0),X'02',AL3(COMMNAME)
```

SPACE

```
*          extended PLIST to call EXECCOMM
RFPLIST DS    0D
RCOMVERB DC   A(COMMNAME)
RBEGARGS DC   F'0'
RENDARGS DC   F'0'
RFBLOCK  DC   A(SHVBLOK)
RARGLIST DC   F'0'
RFUNRET  DC   F'0'
```

SPACE

```
SHVBLOK DS    0D          shared variable PLIST element
SHVNEXT  DC    A(0)        chain pointer
SHVUSER  DC    F'0'        private use
SHVCODE  DC    C'S'        function code (SET variable)
SHVRET   DC    X'00'       return code
        DC    H'0'
SHVBUFL  DC    F'0'        length of 'FETCH' value buffer
SHVNAMA  DC    A(VNAME)    address of variable name
SHVNAML  DC    F'0'        length of variable name
SHVVALA  DC    A(0)        address of value buffer
SHVVALL  DC    F'0'        length of value buffer
SHVBLEN  EQU   *-SHVBLOK  length of block
```

```

SAVNAML DC F'0' length of variable name as supplied
EJECT
*****
* A table of numbers is used rather than working *
* them out because the storage used is irrelevant *
* and fewer instructions are needed *
*-----*
* Increase STEMNOS table for more then 99 blocks *
* but beware that resulting length for read CCW *
* cannot be more than X'FFFF' *
*****
CURSTEM DC A(STEMNOS)
DC C'0 ',X'1'
STEMNOS DC C'1 ',X'1',C'2 ',X'1',C'3 ',X'1',C'4 ',X'1',C'5 ',X'1'
DC C'6 ',X'1',C'7 ',X'1',C'8 ',X'1',C'9 ',X'1',C'10',X'2'
DC C'11',X'2',C'12',X'2',C'13',X'2',C'14',X'2',C'15',X'2'
DC C'16',X'2',C'17',X'2',C'18',X'2',C'19',X'2',C'20',X'2'
DC C'21',X'2',C'22',X'2',C'23',X'2',C'24',X'2',C'25',X'2'
DC C'26',X'2',C'27',X'2',C'28',X'2',C'29',X'2',C'30',X'2'
DC C'31',X'2',C'32',X'2',C'33',X'2',C'34',X'2',C'35',X'2'
DC C'36',X'2',C'37',X'2',C'38',X'2',C'39',X'2',C'40',X'2'
DC C'41',X'2',C'42',X'2',C'43',X'2',C'44',X'2',C'45',X'2'
DC C'46',X'2',C'47',X'2',C'48',X'2',C'49',X'2',C'50',X'2'
DC C'51',X'2',C'52',X'2',C'53',X'2',C'54',X'2',C'55',X'2'
DC C'56',X'2',C'57',X'2',C'58',X'2',C'59',X'2',C'60',X'2'
DC C'61',X'2',C'62',X'2',C'63',X'2',C'64',X'2',C'65',X'2'
DC C'66',X'2',C'67',X'2',C'68',X'2',C'69',X'2',C'70',X'2'
DC C'71',X'2',C'72',X'2',C'73',X'2',C'74',X'2',C'75',X'2'
DC C'76',X'2',C'77',X'2',C'78',X'2',C'79',X'2',C'80',X'2'
DC C'81',X'2',C'82',X'2',C'83',X'2',C'84',X'2',C'85',X'2'
DC C'86',X'2',C'87',X'2',C'88',X'2',C'89',X'2',C'90',X'2'
DC C'91',X'2',C'92',X'2',C'93',X'2',C'94',X'2',C'95',X'2'
DC C'96',X'2',C'97',X'2',C'98',X'2',C'99',X'2'
EJECT
COPY HCPSGIOP DSECT of DIAG A8 parameter block
FLOCK CSECT
DWORK DS D
DIAGBLK DC (SGIDWSIZ)D'0' parameter list for DIAG A8
PACKW PACK DWORK,0(0,R5)
SPACE
PHEXT DS 0F physical extent for Define Extent
DC F'0'
PHBLOCK DC F'0' Block number on device
RELSTRT DC F'0' Relative block number of first block (zero)
RELEND DC F'0' Relative block number of last block
SPACE
LOGEXT DS 0F Locate parameters
LOGREAD DC X'06' Read
LOGCOUNT DC AL3(1) No. of blocks to read
LOGSTRT DC F'0' Relative no. of first block to be read
SPACE

```

```

*          *****
*          * Channel program to read up to 99 blocks *
*          *****
CCWS      CCW   X'63',PHEXT,X'40',16      define extent
          CCW   X'43',LOGEXT,X'40',8      locate record
READCCW   CCW   X'42',*,0,512             read records
*          (length will be set to 512*number of blocks to read)
          SPACE
BUFFAD    DC    A(0)                      address of I/O buffer (also in READCCW)
          REGEQU
          SPACE 3

          LTORG
EFPLIST   DSECT
ECOMVERB  DS    F
EBEGARGS  DS    F
EENDARGS  DS    F
EFBLOCK   DS    F
EARGLIST  DS    F
EFUNRET   DS    F
          END

```

*John Illingworth
Systems Engineer
Wm Morrison Supermarkets (UK)*

© Xephon 1999

A full screen console interface – part 13

Editor's note: the following article is an extensive piece of work which will be published over several issues of VM Update. It was felt that readers could benefit from the entire article and from the individual sections. Any comments or recommendations would be welcomed and should be addressed either to Xephon or directly to the author at fernando_duarte@vnet.ibm.com.

The modules CSCRNC, CSCRNL, and CSCUSA add support for APPC/VM.

CSCRNC ASSEMBLE

```

          TITLE 'CSCRNC - CSC Remote Node Commands'
CSCRNC   START X'01A9D0'
          PRINT NOGEN
          CSCHDR                               Remote Node Commands
*

```

```

* Process APPC/VM basic commands and routines
*
*
      USING IPARML,R9          IUCV Parameter List
      USING RNDSECT,R5        RND Table
      SPACE 3

*
* Initialize Remote Node Connections (APPC)
*   Connect to *IDENT CP System Service
*
*
      MSG   0400              Let's start APPC/VM
      MVI   RNCFLG01,X'00'    Clear our control flags
      L     R1,RNLFLGS
      MVI   0(R1),X'00'       Same for CSCRNL flags
      STCTL C0,C0,RNCCTL0     Save current C0
      LA    R0,133            *T* Create trace entry
      LINK  TRACE             *T*
      LA    R5,RNDPTR         Address RND table
INIT100  L     R5,RNDFWD
      TM    RNDOPT1,RNDOLCL   Local LOCAL entry
      BZ    INIT100
      MVC   RNCNAME,RNDRSRC   Copy Local resource name
      MVC   RNCUDATA,RNDRSRC  Copy also into IUCV user data
      L     R1,SBNODE
      MVC   0(L'RNDNODE,R1),RNDNODE Initialize <CSC>$ID buffer
      L     R1,SBRSRC
      MVC   0(L'RNDRSRC,R1),RNDRSRC
      HNDIUCV SET,NAME=RNCNAME,EXIT=RNLSTX Inform CMS about it
      LTR   R15,R15
      BZ    INIT200
      LR    R2,R15           We got a problem, forget it
      MSG   0401
      B     INIT900
      SPACE
INIT200  OI    RNCFLG01,RNCIUCVS Remember to clear it at the end
      MVI   RNCRTYPE,RNCRGLB Define Global resource
      TM    RNDOPT1,RNDOLRS  Should we define it as Local
      BZ    INIT210         No, keep it Global
      MVI   RNCRTYPE,RNCRLCL Yes, change it to Local resource
INIT210  IUCV  CONNECT,PRMLIST=(R9),USERID=RNCIDENT,USERDTA=RNCUDATA, *
          MF=L
          CMSIUCV CONNECT,NAME=RNCNAME,PRMLIST=(R9),EXIT=RNL CNX
          LTR   R15,R15
          BZ    INIT300     Check for errors
          BAS   R14,ACERROR Display message on the console
          B     INIT900
          SPACE
INIT300  OI    RNCFLG01,RNCIUCVC Remember to sever connection
INIT900  BACK
          SPACE 3

```

```

*
* Terminate Remote Node Connections (APPC)
*
*
CSCRNCTR RELOC          Terminate APPC/VM communications
      LCTL  C0,C0,RNCCTL0      Restore C0
      MSG   0423                Start with a message
      TM    RNCFLG01,RNCIUCVC  Are we connected to *IDENT
      BZ    TERM500             No, half done
      NI    RNCFLG01,X'FF'-RNCIUCVC Reset option
      LA    R5,RNDPTR           Address RND table
TERM100  L      R5,RNDFWD       Scan all table
      LTR   R5,R5
      BZ    TERM500             End of table found
      GO    CSCRNCSPP           Terminate Send and Receive links
      B     TERM100
      SPACE
TERM500  TM    RNCFLG01,RNCIUCVS  Is CMS IUCV initialized?
      BZ    TERM900             No, all done
      NI    RNCFLG01,X'FF'-RNCIUCVS Reset option
      HNDIUCV CLR,NAME=RNCNAME   Clear CMS IUCV trap
      LTR   R15,R15             Not much we can do...
      BZ    TERM900
      MSG   0425,RC             ... but let us know if...
TERM900  BACK
      SPACE 3
*
* Start link to remote node
*
*      Input R5 addresses RND entry of the link to start
*
*
CSCRNCST RELOC          Start APPC/VM link
      L     R9,@SCPARMA         Address APPC/VM Parameter List
      LA    R2,RNDNODE          Address node name for messages
      MSG   0503
*      MVI  IPFLAGS1,X'00'      Clear all flags
*      MVI  IPFLAGS2,X'00'
      APPCVM CONNECT,PRMLIST=(R9),RESID=RNDRSRC,BUFLN=ZERO,          *
              SYNCLVL=NONE,WAIT=NO,TYPE=BASIC,LOGDATA=NO,FMH5=NO,MF=L
      CMSIUCV CONNECT,PRMLIST=(R9),NAME=RNCNAME,COMDIR=NO,          *
              EXIT=RNLSDX
      L     R1,IPPATHID         Store IPPATHID+IPTYPE+IPRCODE
      ST    R1,RNDPIDS
      LTR   R15,R15             Any problems?
      BZ    START900
      BAS   R14,ACERROR         Yes, display error message
      SR    R0,R0
      ST    R0,RNDPIDS         Clear IPPATHID+IPTYPE+IPRCODE
START900 LA    R0,134           *T* Create trace entry
      LINK  TRACE               *T*

```



```

BACK
SPACE 3
*
* Stop link to remote node
*
*      Input R5 addresses RND entry of the link to stop
*
*
CSCRNCSP RELOC          Stop APPC/VM link
TM      RNDOPT1,RNDOSND+RNDORCV Is link up? (send and receive)
BNO     STOP100
LA      R2,RNDNODE       Yes, address node name for msg
MSG     0515             Display message
GO      CSCUSACL         Terminate affected sessions
STOP100 L      R1,RNDPIDS Start with the Send link
LTR     R1,R1
BZ      STOP600         Nothing, check receive link
ST      R1,IPPATHID     Store PATHID into Parameter List
SR      R0,R0
ST      R0,RNDPIDS      Clear Send PATHID
MVI     IPFLAGS1,X'00'  Clear all flags
CMSIUCV SEVER,NAME=RNCNAME,PRMLIST=(R9) Do the job
LTR     R15,R15         Check for errors
BZ      STOP600         So far so good
C       R15,TWO         Allow for return code 2
BE      STOP600
*      LA      R0,12     Is path gone?
*      CR      R0,R15
*      BE      STOP600   Yes, ignore it
LR      R2,R15         Copy return code for message
MSG     0533          We got a problem
SPACE
STOP600 TM      RNDOPT1,RNDOLCL  Is this entry for local node
BO      STOP800       Yes, so there is no receive link
L       R1,RNDPIDR    Load Receive PATHID
LTR     R1,R1
BZ      STOP800       Nothing, all done
ST      R1,IPPATHID   Store PATHID into Parameter List
SR      R0,R0
ST      R0,RNDPIDR    Clear Receive PATHID
MVI     IPFLAGS1,X'00' Clear all flags
CMSIUCV SEVER,NAME=RNCNAME,PRMLIST=(R9) Do the job
LTR     R15,R15
BZ      STOP800
C       R15,TWO       Allow return code 2
BE      STOP800
*      LA      R0,12     Is path gone?
*      CR      R0,R15
*      BE      STOP800   Yes, ignore it
LR      R2,R15         Copy return code for message
MSG     0533

```

```

SPACE
STOP800  TM    RNDOPT1,RNDOSND+RNDORCV Was link up?
        BNO    STOP900
        LA     R2,RNDNODE           Yes, address node name for msg
        MSG    0517                Display message
STOP900  NI    RNDOPT1,RNDOLCL+RNDOLRS+RNDOTMP Reset almost all flags
        MVI    RNDOPT2,X'00'       Clear option byte 2
        BACK
        SPACE 3

*
* Send data over APPC/VM link
*
*      Input R5 addresses RND entry
*
*
CSCRNCSD RELOC          Send data over APPC/VM link
        L     R9,@SCPARMA          Address APPC/VM IUCV Parm List
        L     R4,RNDSBUFF          Address Send buffer
        LH    R3,0(,R4)            Length of data to send
        OI    RNDOPT2,RNDOSPG      SENDDATA in progress
        APPCVM SENDDATA,PRMLIST=(R9),BUFFER=(R4),BUFLN=(R3),
        RECEIVE=NO,PATHID=RNDPIDS
        BZ    SEND900
        BC    X'F'-CC2,SEND800
        NI    RNDOPT2,X'FF'-RNDOSPG Reset Send in progress
        CLI   IPRCODE,IPCOMP       Did it complete normally?
        BE    SEND900
SEND800  MSG    0530                Display error message
SEND900  BACK
        SPACE 3

*
* Display appropriate error message from APPC/VM CONNECT
*
*
ACERROR  EQU    *                  Select error message
        ST    R14,APPCSV14
        LA   R2,RNDNODE            Address node name
        LA   R3,RNDRSRC            Address resource name
        LA   R0,APPCNOAU           Return code for not authorized
        CR   R0,R15
        BNE  ACE100                Try something else
        MSG  0440                  User not authorized
        B    ACE900
        SPACE
ACE100   LA   R0,APPCNOAV
        CR   R0,R15
        BNE  ACE200
        MSG  0441                  Resource not available
        B    ACE900
        SPACE
ACE200   LA   R0,APPCMAXC

```

	CR	R0,R15	
	BNE	ACE300	
	MSG	0442	Maximum connections (partner)
	B	ACE900	
	SPACE		
ACE300	LA	R0,APPCMAXP	
	CR	R0,R15	
	BNE	ACE800	
	MSG	0443	Maximum connections reached
	B	ACE900	
	SPACE		
ACE800	MSG	0448	Non-specific message
	MSG	0449	
ACE900	L	R14,APPCSV14	
	BR	R14	
	SPACE	3	
	ENTRY	RNCNAME	
	EXTRN	RNLSTX,RNLCNX,RNLSDX	
	SPACE		
RNLFLGS	DC	V(RNLFLG01)	RNL Flag byte
SBNODE	DC	V(SBININD)	
SBRSRC	DC	V(SBINIRS)	
RNCCTL0	DS	F	Save C0
APPCSV14	DS	F	Save R14 - ACERROR
	SPACE		
CC2	EQU	2	BC mask to check for CC 2
	SPACE		
RNCFLG01	DS	X	
RNCIUCVS	EQU	X'80'	HNDIUCVS executed
RNCIUCVC	EQU	X'40'	CMSIUCV CONNECT executed
	SPACE		
APPCNOAV	EQU	X'3E8'+X'0B'	Resource not available
APPCMAXC	EQU	X'3E8'+X'0D'	Maximum connections reached
APPCMAXP	EQU	X'3E8'+X'0E'	Maximum connections for partner
APPCNOAU	EQU	X'3E8'+X'0F'	Missing authorization for *IDENT
	SPACE		
	DS	0D	
RNCIDENT	DC	C'*IDENT '	CP System Service name
RNCNAME	DS	CL8	CMS IUCV name
RNCUDATA	DS	CL8	*1* CP *IDENT User data
	DC	X'01'	*2* Identify resource
RNCRTYPE	DC	X'00'	*3* Resource type (Local / Global)
RNCRLCL	EQU	X'40'	*4* Local resource
RNCRGLB	EQU	X'C0'	*5* Global resource
	DC	X'00000000000000'	*6*
	SPACE	3	
	CSCDATA		
	CSCDS	(RND)	
	PUSH	PRINT	
	PRINT	OFF	
	COPY	IPARML	

```

POP PRINT
REGEQU
END

```

CSCRNLASSEMBLE

```

TITLE 'CSCRNL - CSC Remote Node Links'
MACRO
&LABEL EXEC &DEST
&LABEL LA R6,&DEST Address internal routine
BASR R14,R6 Execute
MEND
SPACE 3
CSCRNL START X'01A278'
PRINT NOGEN
CSCHDR Remote Node Links
*
* Process APPC/VM requests
*
*
USING IPARML,R9 IUCV Parameter List
USING RNDSECT,R5 RND Table
SPACE
NI CSCFLG02,X'FF'-WORKRM Reset global flag
L R9,@SCPARMA Address IUCV Parameter List
TM RNLFLG01,APPCWKCN
BZ RNL100
BAS R14,CNPROC Interrupts from CONNECT *IDENT
RNL100 TM RNLFLG01,APPCWKST
BZ RNL200
BAS R14,STPROC Interrupts for new pending links
RNL200 TM RNLFLG01,APPCWKCH
BZ RNL300
BAS R14,CHKTEMP Check temporary RND entries
RNL300 TM RNLFLG01,APPCWKSD
BZ RNL400
BAS R14,SDPROC Interrupts from links (send)
RNL400 TM RNLFLG01,APPCWKRC
BZ RNL500
BAS R14,RCPROC Interrupts from links (receive)
RNL500 TM RNLFLG01,APPCWKTM
BZ RNL900
BAS R14,TMPROC Interrupts from Interval Timer
RNL900 BACK
SPACE 3
*
* Process CP *IDENT interrupts
*
*
CNPROC EQU * Process CP *IDENT interrupts

```

	ST	R14,CNPRSV14	
	NI	RNLFLG01,X'FF'-APPCWKC	Reset flag
	LA	R5,RNDPTR	Address RND table
CNPR100	L	R5,RNDFWD	
	TM	RNDOPT1,RNDOLCL	Locate LOCAL entry
	BZ	CNPR100	
	LA	R0,129	*T* Create trace entry
	LINK	TRACE	*T*
	LA	R2,RNDNODE	Just for messages
	LA	R3,RNDRSRC	
	TM	RNDOPT2,RNDOCNS	Severed connection?
	BZ	CNPR400	
	NI	RNDOPT2,X'FF'-RNDOCNS	Yes, reset option
	TM	RNDOPT1,RNDOSND	Was link up before
	BZ	CNPR200	No, not available
	MSG	0500	Local node terminated
	B	CNPR300	
	SPACE		
CNPR200	MSG	0501	Local node not available
CNPR300	GO	CSCRNCSP	
	B	CNPR900	
	SPACE		
CNPR400	TM	RNDOPT2,RNDOCNC	Connection complete?
	BZ	CNPR800	
	NI	RNDOPT2,X'FF'-RNDOCNC	Yes, reset option
	OI	RNDOPT1,RNDOSND	Link to *IDENT is up
	MSG	0502	Local node activated
	LA	R5,RNDPTR	Address RND table
CNPR500	L	R5,RNDFWD	Scan all entries
	LTR	R5,R5	Check for end of table
	BZ	CNPR600	Found it, all done
	TM	RNDOPT1,RNDOLCL	Process only remote entries
	BO	CNPR500	
	GO	CSCRNCST	
	B	CNPR500	
	SPACE		
CNPR600	BAS	R14,SETIME	Set Interval Timer Exit Routine
	B	CNPR900	
	SPACE		
CNPR800	MSG	0521	Call the Ghostbusters
	DC	H'0'	
	SPACE		
CNPR900	L	R14,CNPRSV14	
	BR	R14	
	SPACE	3	
	*		
	*	Process APPC/VM requests (pending connections)	
	*		
	*		
STPROC	EQU	*	APPC/VM interrupts (pending)

```

ST      R14,STPRSV14
NI      RNLFLG01,X'FF'-APPCWKST  Reset flag
LA      R5,RNDPTR                  Address RND table
STPR100 L      R5,RNDFWD              Scan all entries
LTR     R5,R5
BZ      STPR900                    End of table, all done
TM      RNDOPT1,RNDORQP            Any interrupt pending?
BZ      STPR100                    No, keep going
NI      RNDOPT1,X'FF'-RNDORQP      Reset option
LA      R0,130                     *T* Create trace entry
LINK    TRACE                       *T*
TM      RNDOPT2,RNDOCNP            Is this a pending connection?
BZ      STPR300
NI      RNDOPT2,X'FF'-RNDOCNP
MVI     IPFLAGS1,X'00'             Clear all flags
IUCV    ACCEPT,PRMLIST=(R9),PATHID=RNDPIDR,MF=L
CMSIUCV ACCEPT,PRMLIST=(R9),NAME=RNCNAME,EXIT=RNLACX
LTR     R15,R15
BZ      STPR100
C       R15,TWO                    Return code 2 may be valid
BNE     STPR200
CLI     IPRCODE,IPCOMP             Did it complete normally?
BE      STPR100
STPR200 MSG    0531                IUCV ACCEPT error
GO      CSCRNCSP                   Sever link
B       STPR100
SPACE
STPR300 TM      RNDOPT2,RNDOCNS      Is this a severed connection?
BZ      STPR800                    No, so what is it?
GO      CSCRNCSP
B       STPR100
SPACE
STPR800 MSG    0522                Call the Ghostbusters II
DC H'0'
STPR900 L      R14,STPRSV14
BR      R14
SPACE 3
*
* Process APPC/VM requests (send)
*
SDPROC  EQU    *                    APPC/VM interrupts (send)
ST      R14,SDPRSV14
NI      RNLFLG01,X'FF'-APPCWKSD    Reset flag
LA      R5,RNDPTR                  Address RND table
SDPR100 L      R5,RNDFWD              Scan all entries
LTR     R5,R5
BZ      SDPR900                    End of table, all done
TM      RNDOPT1,RNDORQS            Anything pending interrupt?
BZ      SDPR100                    No, try another
NI      RNDOPT1,X'FF'-RNDORQS      Reset option

```

```

LA      R0,131          *T* Create trace entry
LINK    TRACE          *T*
LA      R2,RNDNODE     Just for messages
LA      R3,RNDRSRC
TM      RNDOPT2,RNDOCNS Severed connection?
BZ      SDPR300
NI      RNDOPT2,X'FF'-RNDOCNS-RNDOCNC Yes, reset options
TM      RNDOPT1,RNDOSND+RNDORCV Was link up before?
BNO     SDPR200        No, just not available
MSG     0510           Remote node terminated
B       SDPR210
SPACE
SDPR200 MSG 0511       Remote node not available
SDPR210 GO  CSCRNCSP   Sever send and receive links
B       SDPR100       Check other RND entries
SPACE
SDPR300 TM  RNDOPT2,RNDOCNC Connection complete?
BZ      SDPR500
NI      RNDOPT2,X'FF'-RNDOCNC Yes, reset option
OI      RNDOPT1,RNDOSND Send link is up
OI      RNDOPT2,RNDOSPG SENDDATA in progress
APPCVM SENDDATA,PRMLIST=(R9),BUFFER=SBINI,BUFLEN=SBINIL, *
        RECEIVE=NO,PATHID=RNDPIDS
BZ      SDPR400       Interrupt never comes, why?
BC      CC2,SDPR400
MSG     0530         Display error message
GO      CSCRNCSP     Sever send and receive links
B       SDPR100     Next...
SPACE
SDPR400 NI  RNDOPT2,X'FF'-RNDOSPG SENDDATA complete
TM      RNDOPT1,RNDORCV Is receive link up?
BZ      SDPR100     Not yet, just wait
MSG     0512         Remote node activated
B       SDPR100
SPACE
SDPR500 TM  RNDOPT2,RNDOFNC Function complete?
BZ      SDPR800
NI      RNDOPT2,X'FF'-RNDOFNC Yes, reset option
NI      RNDOPT2,X'FF'-RNDOSPG Reset Send in Progress option
TM      RNDOPT2,RNDOPND Is anything waiting?
BZ      SDPR100
GO      CSCUSAPD     Yes, process pending request
B       SDPR100
SPACE
SDPR800 MSG 0523     Call the Ghostbusters III
DC H'0'
SDPR900 L   R14,SDPRSV14
BR      R14
SPACE 3

```

*

```

* Process APPC/VM requests (receive)
*
RCPROC EQU * APPC/VM interrupts (receive)
ST R14,RCPRSV14
NI RNLFLG01,X'FF'-APPCWKRC Reset flag
LA R5,RNDPTR Address RND table
RCPR100 L R5,RNDFWD Scan all table
LTR R5,R5
BZ RCPR900 End of table, all done
TM RNDOPT1,RNDORQR Any interrupt to process
BZ RCPR100
NI RNDOPT1,X'FF'-RNDORQR Yes, reset option
LA R0,132 *T* Create trace entry
LINK TRACE *T*
TM RNDOPT1,RNDORCV Is receive link up?
BO RCPR700 Yes, process as user data
TM RNDOPT2,RNDOMSG Incoming message?
BZ RCPR200
NI RNDOPT2,X'FF'-RNDOMSG Yes, reset option
LH R0,RNDMSG Load length of incoming message
C R0,RBINIL Is it the expected length?
BNE RCPR190 No, forget it
OI RNDOPT2,RNDORPG RECEIVE in progress
APPCVM RECEIVE,PRMLIST=(R9),PATHID=RNDPIDR, *
        BUFFER=RBINI,BUFLEN=RBINIL
BC CC2,RCPR210
BZ RCPR100 Wait for FCA interrupt
RCPR190 GO CSCRNCSP
B RCPR100 Check all RND entries
SPACE
RCPR200 TM RNDOPT2,RNDOFNC Function complete?
BZ RCPR600 No, so what is it?
NI RNDOPT2,X'FF'-RNDOFNC Reset option
RCPR210 NI RNDOPT2,X'FF'-RNDORPG RECEIVE complete
CLC RBINIID,SBINIID Validate <CSC>$ID data
BNE RCPR380 Wrong header, forget it
L R0,RNDPIDR Load IUCV PATHID
LR R4,R5 Save RND entry address
RCPR300 L R5,RNDFWD Scan the table
LTR R5,R5
BZ RCPR370 End of table, link is unknown
SR R1,R1 Start with zero matches
CLC RBININD,RNDNODE Compare node names
BNE RCPR310
A R1,ONE We got one right
RCPR310 CLC RBINIRS,RNDRSRC Compare resource names
BNE RCPR320
A R1,ONE We got another one right
RCPR320 C R1,ONE How many rights?
BH RCPR400 Two, we found our entry

```


	BL	RCPR300	Zero, try another RND entry
	LA	R2,RBININD	One, that's bad news
	LA	R3,RBINIRS	
	MSG	0535	Link not reversible
	B	RCPR380	
	SPACE		
RCPR370	LA	R2,RBININD	Address fields for message
	LA	R3,RBINIRS	
	MSG	0536	Link not defined
RCPR380	LR	R5,R4	
	GO	CSCRNCSP	Sever links
	TM	RNDOPT1,RNDOTMP	Is it temporary? (should be)
	BZ	RCPR100	No, bad news but that's OK
	LA	R1,RNDPTR	Address RND table
RCPR390	LR	R2,R1	Save address of previous entry
	L	R1,RNDFWD-RNDSECT(,R1)	Address entry to check
	CR	R1,R5	Is it our current entry?
	BNE	RCPR390	No, check all
	L	R0,RNDFWD-RNDSECT(,R1)	Address following entry
	ST	R0,RNDFWD-RNDSECT(,R2)	Store in previous entry
	LR	R5,R2	Address previous entry
	LA	R0,RNDSIZE	Entry size in double words
	LINK	RELEASE	Release temporary RND entry
	B	RCPR100	Keep going with other nodes
	SPACE		
RCPR400	ST	R0,RNDPIDR	Store Receive PATHID
	OI	RNDOPT1,RNDORCV	Receive link is up
	LA	R1,RNDPTR	Address RND table again
RCPR410	LR	R2,R1	Save address of previous entry
	L	R1,RNDFWD-RNDSECT(,R1)	Address next entry
	CR	R1,R4	Compare with current one
	BNE	RCPR410	Keep going until we found it
	L	R0,RNDFWD-RNDSECT(,R1)	Chain previous entry with next
	ST	R0,RNDFWD-RNDSECT(,R2)	
	LA	R0,RNDSIZE	Entry length in double words
	LINK	RELEASE	Release RND temporary entry
	TM	RNDOPT1,RNDOSND	Is Send link already up?
	BO	RCPR500	Yes, youppi, we did it
	L	R1,RNDPIDS	Load Send PATHID
	LTR	R1,R1	Are we connect pending
	BNZ	RCPR100	Yes, wait
	GO	CSCRNCST	No, start the link again
	B	RCPR100	
	SPACE		
RCPR500	LA	R2,RNDNODE	Just for messages
	MSG	0512	Remote node activated
	LA	R5,RNDPTR	Not nice, but it works
	B	RCPR100	
	SPACE		
RCPR600	TM	RNDOPT2,RNDOCNS	Severed connection?

```

        BZ      RCPR690
        GO      CSCRNCSP          Sever Send and Receive links
        B       RCPR100
        SPACE
RCPR690 MSG    0524              Call the Ghostbuster IV
        DC H'0'
        SPACE
RCPR700 TM     RNDOPT1,RNDOSND   Make sure Send link is up
        BO     RCPR750
        GO     CSCRNCSP          Cannot receive before handshake
        B      RCPR100
        SPACE
RCPR750 TM     RNDOPT2,RNDOCNS   Severed connection?
        BO     RCPR900          Don't worry, Send will handle it
        TM     RNDOPT2,RNDOMSG   Incoming message?
        BZ     RCPR800          No, so what is it?
        NI     RNDOPT2,X'FF'-RNDOMSG Yes, reset option
        L      R4,RNDRBUFF       Address Receive buffer
        LH     R3,RNDMSGL        Incoming message length
*       LA     R0,RNDBUFSZ       Load buffer size in dwords
*       SLL    R0,3              Convert to bytes
        OI     RNDOPT2,RNDORPG   RECEIVE in progress
        APPCVM RECEIVE,PRMLIST=(R9),PATHID=RNDPIDR,
        BUFFER=(R4),BUFLN=(R3)
        BC     CC2,RCPR780
        BZ     RCPR100           Wait for FCA interrupt
        NI     RNDOPT2,X'FF'-RNDORPG
        MSG    0532              APPC/VM Receive error
        B      RCPR100
        SPACE
RCPR780 NI     RNDOPT2,X'FF'-RNDORPG
        GO     CSCUSA
        B      RCPR100
        SPACE
RCPR800 MSG    0525              Call the Ghostbusters V
        B      RCPR100
        SPACE
RCPR900 L      R14,RCPRSV14
        BR     R14
        SPACE 3
*
* Set Interval Timer Exit Routine
*
SETIME  EQU    *                Set Interval Timer Exit Routine
        HNDEXT SET,CODE=1005,TIMEXIT Trap Interval Timer Interrupts
        L      R1,TIMEMIN        Load interval time in minutes
        SLL    R1,12             Shift to bit 51
        M      R0,TIMEMIC        Convert to micro seconds
        STM    R0,R1,TIMEPT      Save for later use
        SPT    TIMEPT            Set Interval Timer

```

	STCTL	C0,C0,TIMECTL0	Store current C0
	L	R0,TIMEINT	Load Interval Timer bit
	O	R0,TIMECTL0	Set Interval Timer bit
	ST	R0,TIMECTL0	Store new value
	LCTL	C0,C0,TIMECTL0	Load into C0
	BR	R14	
	SPACE	3	
TMPROC	EQU	*	
	ST	R14,TMPRSV14	
	NI	RNLFLG01,X'FF'-APPCWKTM	Reset Timer option
	LA	R5,RNDPTR	Address RND table
TMPR100	L	R5,RNDFWD	Scan table
	TM	RNDOPT1,RNDOLCL	Locate Local node
	BZ	TMPR100	Not this one
	TM	RNDOPT1,RNDOSND	Is Local node active
	BZ	TMPR900	No, all done
	L	R1,RNDPTR	Address RND table
TMPR400	LTR	R5,R1	Check for end of table
	BZ	TMPR900	Found it, all done
	L	R1,RNDFWD	Address following entry
	TM	RNDOPT1,RNDOLCL	Is it the Local node?
	BO	TMPR400	Yes, skip it
	TM	RNDOPT1,RNDOSND+RNDORCV	Is link already active?
	BO	TMPR400	Yes, skip it
	L	R0,RNDPIDS	Is activation in progress?
	LTR	R0,R0	
	BNZ	TMPR400	Yes, skip it
	GO	CSCRCST	Start link
	L	R1,RNDFWD	Address following entry
	B	TMPR400	Scan all RND table
	SPACE		
TMPR900	L	R14,TMPRSV14	
	BR	R14	
	SPACE	3	
	*		
	*	Process RND temporary entries on Send links	
	*		
CHKTEMP	EQU	*	Process RND temporary entries
	ST	R14,CHKTSV14	
	NI	RNLFLG01,X'FF'-APPCWKCH	Reset option
	LA	R5,RNDPTR	Address RND table
CHK100	LR	R2,R5	Save address of previous entry
	L	R5,RNDFWD	Address entry to process
	TM	RNDOPT1,RNDOTMP	Is it temporary?
	BZ	CHK900	No, all done
	L	R0,RNDPIDS	Load IPPATHID+IPTYPE+IPRCODE
	IC	R1,RNDOPT2	Load RND option byte 2
	LR	R4,R5	Save entry address
CHK200	L	R5,RNDFWD	Now scan the table
	LTR	R5,R5	End of table?

	BNZ	CHK300	
	LR	R5,R4	Yes, forget this entry
	B	CHK100	
	SPACE		
CHK300	CLM	R0,B'1100',RNDPIDS	Compare IUCV PATHID
	BNE	CHK200	
	OI	RNDOPT1,RNDORQS	Found it, set request option
	ST	R0,RNDPIDS	Save IPTYPE and IPRCODE
	IC	R0,RNDOPT2	Merge option byte 2
	OR	R0,R1	
	STC	R0,RNDOPT2	Save merged value
	L	R0,RNDFWD-RNDSECT(,R4)	Address following entry
	ST	R0,RNDFWD-RNDSECT(,R2)	Store in previous entry
	LR	R5,R2	Address it
	LA	R0,RNDSIZE	Entry size in double words
	LR	R1,R4	Copy address to R1
	LINK	RELEASE	Release storage
	B	CHK100	Check all table
	SPACE		
CHK900	L	R14,CHKTSV14	
	BR	R14	
	SPACE	3	
	*		
	*	HNDIUCV SET Exit routine	
	*		
	*		
RNLSTX	EQU	*	Process new connections
	USING	RNLSTX,R11	
	STM	R14,R12,12(R13)	
	LR	R11,R15	
	L	R12,ACSCCODE	Establish addressability for...
	L	R10,ACSCDATA	common code and data
	OI	CSCFLG02,WORKRM	Remember we had an interrupt
	OI	RNLFLG01,APPCWKST	
	LR	R9,R2	Address IUCV Parameter List
	LA	R0,135	*T* Create trace entry
	LINK	TRACE	*T*
	EXEC	ADDTEMP	Create temporary RND entry
	L	R1,IPPATHID	Load IUCV PATHID+IPTYPE+IPRCODE
	ST	R1,RNDPIDR	Store everything into RND entry
	OI	RNDOPT1,RNDORQP	Remember to process it
	CLI	IPTYPE,IPTYPPCA	Pending connection?
	BNE	STX100	
	OI	RNDOPT2,RNDOCNP	Yes, set option
	B	STX900	
	SPACE		
STX100	CLI	IPTYPE,IPTYPSVA	Severed connection?
	BNE	STX800	
	OI	RNDOPT2,RNDOCNS	Yes, set option
	B	STX900	

```

SPACE
STX800 LA R8,4(,R9) Second full words of PRMLIST
MSG 0526
SPACE
STX900 TM CSCFLG02,CSCWAIT
BZ STX910
NI CSCFLG02,X'FF'-CSCWAIT Post ECB once if required
OI CSCECB,CSCPOST
STX910 LM R14,R12,12(R13)
BR R14
SPACE
DROP R11
SPACE 3
*
* CMSIUCV Exit Routine (CP) IUCV Interrupts FROM CONNECT TO *IDENT
*
RNLCNX EQU * CMSIUCV Exit Routine (CP *IDENT)
USING RNLCNX,R11
STM R14,R12,12(R13)
LR R11,R15
L R12,ACSCCODE Establish addressability for...
L R10,ACSCDATA common code and data
OI CSCFLG02,WORKRM Remember we had an interrupt
OI RNLFLG01,APPCWKN Select routine to process it
LR R9,R2 Address IUCV Parameter List
LA R0,136 *T* Create trace entry
LINK TRACE *T*
LA R5,RNDPTR Address RND table
CNX100 L R5,RNDFWD
TM RNDOPT1,RNDOLCL Locate LOCAL entry
BZ CNX100 Not this one
L R1,IPPATHID Save IPPATHID+IPTYPE+IPRCODE
ST R1,RNDPIDS
CLI IPTYPE,IPTYPCC Connection complete?
BNE CNX200
OI RNDOPT2,RNDOCNC Yes, set option
B CNX900
SPACE
CNX200 CLI IPTYPE,IPTYPSC Severed connection?
BNE CNX800
OI RNDOPT2,RNDOCNS Yes...
B CNX900
SPACE
CNX800 LA R8,4(,R9) Second full words of PRMLIST
MSG 0526
SPACE
CNX900 TM CSCFLG02,CSCWAIT
BZ CNX910
NI CSCFLG02,X'FF'-CSCWAIT Post ECB once if required
OI CSCECB,CSCPOST

```

```

CNX910  LM    R14,R12,12(R13)
        BR    R14
        SPACE
        DROP  R11
        SPACE 3

*
* CMSIUCV Exit Routine (APPC) APPC/VM Remote links interrupts (send)
*
*
RNLSDX  EQU   *
        USING RNLSDX,R11
        STM   R14,R12,12(R13)
        LR    R11,R15
        L     R12,ACSCCODE          Establish addressability for...
        L     R10,ACSCDATA          common code and data
        OI    CSCFLG02,WORKRM       Remember we had an interrupt
        OI    RNLFLG01,APPCWKSD     Select routine to process it
        LR    R9,R2                 Address IUCV Parameter List
        LA    R0,137                *T* Create trace entry
        LINK  TRACE                  *T*
        L     R0,IPPATHID           Load PATHID (first two bytes)
        LA    R5,RNDPTR             Address RND table
SDX100  L     R5,RNDFWD             Scan it
        LTR   R5,R5                 Check for end of table
        BNZ   SDX200
        EXEC  ADDTEMP               Found it, create temporary RND
        ST    R0,RNDPIDS            Store IUCV PATHID
        OI    RNLFLG01,APPCWKCH     Set option to process it
SDX200  CLM   R0,B'1100',RNDPIDS    Compare IUCV PATHID
        BNE   SDX100                Not this one
        OI    RNDOPT1,RNDORQS       Set request option
        L     R1,IPPATHID
        ST    R1,RNDPIDS            Save IPTYPE and IPRCODE
        CLI   IPTYPE,IPTYPCCA       Connection complete?
        BNE   SDX300
        OI    RNDOPT2,RNDOCNC       Yes, set option
        B     SDX900
        SPACE
SDX300  CLI   IPTYPE,IPTYP SVA      Severed connection?
        BNE   SDX400
        OI    RNDOPT2,RNDOCNS       Yes...
        B     SDX900
        SPACE
SDX400  CLI   IPTYPE,IPTYP FCA      Function complete?
        BNE   SDX800
        OI    RNDOPT2,RNDOFNC       Set option
        B     SDX900
        SPACE
SDX800  LA    R8,4(,R9)             Second full words of PRMLIST
        MSG   0526

```

```

SPACE
SDX900  TM    CSCFLG02,CSCWAIT
        BZ    SDX910
        NI    CSCFLG02,X'FF'-CSCWAIT  Post ECB once if required
        OI    CSCECB,CSCPOST
SDX910  LM    R14,R12,12(R13)
        BR    R14
        SPACE
        DROP  R11
        SPACE 3
*
* CMSIUCV Exit Routine (APPC) APPC/VM Remote links interrupts (receive)
*
*
RNLACX  EQU   *
        USING RNLACX,R11
        STM   R14,R12,12(R13)
        LR    R11,R15
        L     R12,ACSCCODE           Establish addressability for...
        L     R10,ACSCDATA           common code and data
        OI    CSCFLG02,WORKRM       Remember we had an interrupt
        OI    RNLFLG01,APPCWKRC
        LR    R9,R2                  Address IUCV Parameter List
        LA    R0,138                 *T* Create trace entry
        LINK  TRACE                  *T*
        L     R0,IPPATHID           Load PATHID (first two bytes)
        LA    R5,RNDPTR             Address RND table
ACX100  L     R5,RNDFWD             Scan it
        LTR   R5,R5                 Check for end of table
        BZ    ACX700
        CLM   R0,B'1100',RNDPIDR    Compare IUCV PATHID
        BE    ACX300                 That's the one
        CLM   R0,B'1100',RNDPIDS    Compare also the Send PATHID
        BNE   ACX100                 Not this one
        OI    RNDOPT2,RNDOCNS       Illegal, sever link
        B     ACX900
        SPACE
ACX300  OI    RNDOPT1,RNDORQR       Set Receive request option
        L     R1,IPPATHID
        ST    R1,RNDPIDR            Save IPTYPE and IPRCODE
        CLI   IPTYPE,IPTYPMPA       Incoming message?
        BNE   ACX400
        OI    RNDOPT2,RNDOMSG       Yes, set option
        L     R0,IPBFLN1F           Load incoming message length
        STH   R0,RNDMSG             Store in RND entry
        B     ACX900
        SPACE
ACX400  CLI   IPTYPE,IPTYPSVA       Severed connection?
        BNE   ACX500
        OI    RNDOPT2,RNDOCNS       Yes, set option

```

```

      B      ACX900
      SPACE
ACX500 CLI  IPTYPE,IPTYPFCA      Function complete?
      BNE   ACX800
      NI    RNDOPT2,X'FF'-RNDORPG  Reset Receive in Progress option
      OI    RNDOPT2,RNDOFNC      Set Function Complete option
      B      ACX900
      SPACE
ACX700 LA   R8,4(,R9)          Second full words of PRMLIST
      MSG   0526
ACX800 LA   R8,4(,R9)          Second full words of PRMLIST
      MSG   0526
      SPACE
ACX900 TM   CSCFLG02,CSCWAIT
      BZ    ACX910
      NI    CSCFLG02,X'FF'-CSCWAIT  Post ECB once if required
      OI    CSCECB,CSCPOST
ACX910 LM   R14,R12,12(R13)
      BR    R14
      SPACE
      DROP  R11
      SPACE 3

```

*

* Interval Timer Exit Routine

*

*

```

TIMEXIT EQU  *
      USING TIMEXIT,R11
      STM   R14,R12,12(R13)
      LR    R11,R15
      L     R12,ACSCCODE          Establish addressability for...
      L     R10,ACSCDATA          common code and data
      OI    CSCFLG02,WORKRM      Remember we had an interrupt
      OI    RNFLG01,APPCWKTM
      SPT   TIMEPT
      SPACE
TMX900 TM   CSCFLG02,CSCWAIT
      BZ    TMX910
      NI    CSCFLG02,X'FF'-CSCWAIT  Post ECB once if required
      OI    CSCECB,CSCPOST
TMX910 LM   R14,R12,12(R13)
      BR    R14
      SPACE
      DROP  R11
      SPACE 3

```

Editor's note: this article will be continued next month.

*Fernando Duarte
Analyst (Canada)*

© F Duarte 1999

September 1994 – August 1999 index

Items below are references to articles that have appeared in *VM Update* since September 1994. References show the issue number followed by the page number(s). Individual copies of all issues from that date are available, as are a limited number of issues prior to September 1994.

3270	97.12-19	Data extract	118.12-37
ACCESS	113.37-38, 117.18-24, 121.3-8	Data-in-memory	155.8-23
Accounting	106.8-18, 123.42-50, 124.3-22	Date	113.26-36
ADSORT	137.3-9	Date handling	102.13-25
ALL macros	99.3-9, 104.16-26, 156.3-5	DATEFUNC	143.7-19
ALLSPOOL	144.11-17	DCSS	102.39-43, 110.10
Attributes	106.3-7	DDR	117.41-51
Back-up	99.21-38, 104.26-33, 110.34-35, 117.41-51, 127.10-19, 130.34-51, 131.33-51, 132.9-25, 133.42-51, 134.28-41, 142.44-49	Decimal	112.20-24, 113.47-49
BACKUP	135.6-28, 136.8-19	Diary	128.9-15
Bi-modal	123.18-19	Direct console I/O	149.3-12
BLOCKIO	137.40-51, 138.22-34	Directory	120.17-32
Bookmanager	98.16-20	Directory maintenance	98.40-51, 101.12-20
Broadcast	97.25-46	DIRMAINT	98.40, 110.8
Browse	109.3-4	DIRMAINT SAPI	150.48-51
Bulletin board	128.34-51, 129.28-51, 130.17-32, 131.9-25	Disk information	128.3-8
Cacheing	132.28-40	Display 'pseudo-graphics'	148.37-50, 149.41-51, 151.29
Calculator	106.39-43, 107.42-51	DISTINCT	135.28-34
CALENDAR	139.33-42	DOENDS	141.3-4
CHANGE	144.3-10	DOWNUP	133.3-9
CHPID	101.3-11	DRDA	106.18-39, 111.51
Chronological commands	151.21-29	Dump/restore	137.40-51, 138.22-34
CMS DISTINCT	135.28-34	EBCDIC to ASCII	136.3-7
CMS monitor	139.16-26	ECHANGE	144.3-10
CMSCALL	138.19-21	Edit	101.20-21
CNTLUNIT	101.3	Editing files in parallel	152.3-6
COBOL	112.24-36, 118.5-11	Encryption	116.7-18
COM	142.3-8	Erased files	130.3-13
Combining files	105.32-51	Error	107.13-14
Comments	103.8-15, 142.3-8	ES/9000	101.3-11
Comparing	121.41-51, 122.30-40, 123.3-17	EXECDROP	110.10-12
Console	127.20-36	EXECLOAD	110.10-12
COPY	108.33-35, 131.30-32	Executing	113.50-51
Copying files	141.29-41, 143.30-38	FCOPY	141.29-41
Cross reference	112.40-51	FCOPYPRO	143.30-38
Cut and paste	105.14-20	FDISK	121.3-8
DASD	125.18-44	File display	156.6-11
DASD space	112.3-11	File exchange	119.46-51, 120.44-47
		File list	124.32-47, 124.47-51
		File manager	98.21-31
		File transfer	133.3-9, 139.43-51, 140.10-19

FILESAVE	144.45-48	Message forwarding	110.24-33
FILETAPE	139.3-15	Message suppression	102.3-13
Find files	107.3-13	Migration	115.8-9
Finder	103.32-34	Mini-disks	99.10-12, 99.21-38, 102.26-39, 103.3-7, 104.47-51, 107.3-13, 114.17-19, 110.13-23, 111.19-24, 115.10-32, 120.17-32, 121.3-18, 126.33-51, 142.42-49
Format	100.31-33	Monitor	90.32-39, 115.3-7, 118.3-5
FORTH	100.34-51, 101.32-51, 102.44-51, 103.35-51, 104.34-47, 105.20-32	Monitoring virtual machines	154.3-4
FTP	114.12-16, 115.40-51, 116.38-51, 117.28-40, 125.3-15	Monitoring executed programs	155.3-8
FTTERM	133.3-9	Mouse on the mainframe	146.25-35, 150.34-48, 151.39-51, 152.7-17, 153.26-37, 154.17-29
Full screen console interface	144.18-44, 145.27-41, 146.35-51, 147.25-42, 148.8-34, 149.23-40, 150.12-33, 151.5-20, 152.17-34, 153.37-51, 154.38-51, 155.37-51, 156.30-48	MOVEFILE	139.3-15
Fullscreen	97.12-19, 117.3-17	Multicolumn files	101.22-26
GCS	123.19-22	Multiple machines	145.19-26, 146.17-24, 147.13-24
Hexadecimal	111.41-51, 112.20-24, 113.47-49	MVS	119.46-51, 120.44-47
Horizontal prefix	134.19-28	MXEDIT	138.3-18
HP XEDIT	134.19-28	Named substrings	133.9-16
IBM VM Download Library	151.30-38	New files	107.15-24
ICCF	139.43-51, 140.10-19	Newsgroups	143.39-51
INCLUDE	103.25-31	Non-shared resource	147.3-13
INDF	139.16-26	NOTE	108.36
Intranet	142.9-14	NSS	119.3-14
Inventory	109.28-38	Object-oriented	129.3-4
IOCP	101.3	OfficeVision	108.36-51, 109.43-51, 110.35-51, 111.25-35, 116.3-7
Job staging	130.14-17	ORDER	116.33-38
Keys	101.26-31	Ordering	97.48-51
LESS	133.29-41	Over-used directories	143.3-6
Lines of code	148.35-36	Packing files	148.50-1
LINK	104.47-51, 113.37-38, 117.18-24, 121.3-8	Panel manager	106.44-51, 107.24-41, 108.17-27
Linking spool files	144.11-17	Peek	108.3-6
LISTCAT ALL	150.9-12	PICTURE	135.35-39
LOADLIB	114.26-49	Pipelines	105.51, 107.13, 108.3-6, 109.3-4, 111.40, 121.28-38, 145.3-18
Locate	121.19-27, 122.16-17	PLUS	133.29-41
Locks	105.10	PostScript	97.19-24
Log-on	110.8-10	POWER datafile usage	153.3-8
LOGOFF	109.5-7, 112.37-40	PR/SM	101.3
LOOKVARS	141.5-8	PREPT	136.20-35
LPAR	101.3	PRFOLIE	138.35-45
MACLIB	114.26-49, 122.28-30	Printing	113.39-45
Macro	123.18-19	Printing transparencies	138.35-45
Mailing lists	143.39-51	PROP	113.3-26
Memos	119.27-45	PTF	118.38-39, 140.35-51, 141.8-27, 142.14-43, 143.20-29
Menu	125.46-51, 126.7-26, 127.37-39, 134.42-51, 135.40-51, 136.36-51, 137.29-39	PUNMEM	139.43-51, 140.10-19
Merging	122.3-15		

Purge	100.25-30, 151.3-5	System resources	114.3-12
Purging spool files	144.11-17, 151.3-5	Tabulated data	109.38-42
PUT	118.38-39	Tag files	128.29-33
QUERY	105.3-9, 106.3-7, 111.3-19	TCP/IP	114.12-16, 115.40-51, 116.38-51, 117.28-40, 118.39-51, 119.15-26, 120.33-43
RAMAC	126.30-32	TDSK	109.5-7
RDR	108.3-6, 132.26-27	Terminal	109.28-38
RECEIVE	108.3-6	Time	113.26-36
Recording change	144.45-48	Timing	110.3-7
Renaming	125.15-17	Toolkit	98.31-39
Restore	102.26-39, 130.34-51, 131.33-51, 132.9-25, 133.42-51, 134.28-41, 137.40-51, 138.22-34	Transfer	129.20-23
RETRIEVE	112.15-19, 114.50-51	Transferring code	141.28
REXX	129.5-19	Translation	129.5-19
REXX management	140.20-25	TTIME	110.3
REXX tracking	140.35-51, 141.8-27, 142.14-43, 143.20-29	TXTLIB	114.26-49
RXSQL	105.10	UCO	142.3-8
RXUSERFN	134.3-8	Unerase	115.33-39
SALT	133.29-41	Unix	129.20-23
SAVE	108.28-33	Utilities	112.11-14, 130.33-34
Scheduler	131.3-8, 132.40-48, 133.17-28, 134.9-18	VALDATE	142.50-51
Search	100.6-8, 117.24-27	Vaulting	99.43-51, 100.8-24
Security	120.3-16	VDISK	100.3-5
SELCOPY	97.46-48	VDUMP	37.40-51, 138.22-34
Server	98.21-31	Virtual disks	100.3-5
SET MORE	99.10-12, 99.21-38, 102.26-39, 103.15-24, 107.3-13, 108.6-16	Virtual rooms	104.3-15
SFS	116.19-32, 121.38-41, 131.26-30	VM/ESA	99.13-21, 115.8-9, 140.3-9
SFS directories	148.3-8	VM/ESA V2.3	146.3-17
SFSPACE	143.3-6	VM:Secure	152.35-51, 153.9-25, 154.5-16, 155.24-36, 156.11-16
Sorting	132.3-8, 137.3-9	VMDDR	142.44-49
Spanned records	100.31-33	VMFE2E	120.47-48, 125.44-45
Specified string	135.3-5	VMFPLC2	104.26
Split screen	97.3-6	VMSES/E	120.47-48
Spool	105.3-9, 109.7-16, 109.17-27, 111.36-40, 113.45-46	VSE lock file	156.20-33
SQL	106.8-18, 112.24-36	VREST	37.40-51, 138.22-34
SQL/DS	98.3-15, 105.10-13, 136.20-35, 137.10-28, 134.3-8	VSCS	114.20-25
SQLDBSU	98.3-15	VTIME	110.3
SSM	137.10-28	Web	127.3-9, 129.24-27
Statistics	122.40-51, 123.23-42, 124.23-32	Web site, IBM VM	139.27-32
String	99.39-43, 121.19-27, 122.16-27	Web site, L-Soft International	145.42-51
SUBVM	139.43-51, 140.10-19	Web site, Mainframe Program	149.13-22
SWAP	133.29-41	Web site, Melinda Varian	141.42-51
System	126.3-6	Web site, Mike Cowlshaw	140.26-34
		Web site, REXXLA	154.30-38
		Web site, Sterling Software	147.43-51
		Windows	113.3-26
		XEDIT	97.3-11, 98.31-39, 127.40-51, 128.16-29, 144.3-10
		XEDIT ring	126.27-30
		Year 2000	138.46-51, 140.3-9

VM news

IBM has announced VM/ESA Version 2 Release 4 with new support for IBM System/390 Integrated Server, features of the System/390 G5 and G6, dynamic CP exits, support for the euro, ADSM for VM/ESA Version 3.1, and NFS V3.

There are also TCP/IP enhancements including SMTP Service extensions support, improved usability and functionality of the FTP server, native ATM support, an upgraded Routed server, and enhanced user authorization.

Enhancements to the NFS feature of TCP/IP should provide broader access to files stored on a VM system as well as a number of other performance and administration improvements.

Specific new G5 and G6 features supported include FICON, System/390 Open System Adapter-Express (OSA-Express) Gigabit Ethernet, Cryptographic Coprocessor, IEEE Floating Point, Integrated Cluster Bus (ICB), and Enhanced Input/Output Configuration Program (IOCP).

For further information contact your local IBM representative.

Sterling Software has announced an extension to its VM:Webgateway Web-to-host software with a new CICS/VSE Interface providing Web access to VSE data and applications through a GUI. VSE developers can carry out Web enhancements using CICS/VSE COBOL and other CICS command-level programming languages.

With the new interface, application developers can write CGI scripts using either REXX or CICS/VSE COBOL.

For VSE developers, the product includes a VSE tutorial demonstrating how to create CGI scripts and how to Web-enable VSE applications. CGI scripts from the sample application can be copied and applied to real VSE applications to bring them to the Web.

For further information contact:
Sterling Software, 1800 Alexander Bell Drive, Reston, VA 22091, USA.
Tel: (703) 264 8000.
Sterling Software, Sterling Court, Eastworth Road, Chertsey, Surrey, KT16 8DF, UK.
Tel: (01932) 587000.
URL: <http://www.vm.sterling.com>.

* * *



xephon