

144

VM

August 1998

In this issue

- 3 Controlled replacement of strings in XEDIT
- 11 Easy listing and purging of spool files
- 17 Call for papers
- 18 A full screen console interface
- 45 Recording changes in CMS files
- 49 September 1994 – August 1998 index
- 52 VM news

© Xephon plc 1998

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: xephon@compuserve.com

North American office

Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75067
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 £175.00 in the UK; \$265.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1990 issue, are available separately to subscribers for £14.50 (\$22.50) 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>; 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 1998. 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.

Controlled replacement of strings in XEDIT

GENERAL DESCRIPTION

ECHANGE is an extension to an XEDIT CHANGE subcommand. It is useful when the control of changed strings is required, particularly if only a selected subset of target occurrences are to be modified, depending upon their context. In this case the updates may be performed interactively with ECHANGE.

ECHANGE is written in Assembler and REXX and runs under CMS with VM/SP Release 5.

ECHANGE USAGE

ECHANGE is an XEDIT macro and is invoked from the XEDIT command line. ECHANGE is called as shown:

```
ECHANGE /<old string>[/[<new string>[/]]]
```

where:

- 'old string' is the string to be replaced.
- 'new string' is the replacement for 'old string'.

Note: if 'new string' is not defined or has zero length, then 'old string' is deleted and not replaced.

For a successful search, 'old string' must have a non-zero length.

The search for 'old string' starts from the current line and goes forward to the end of the file. It is performed within the limits defined by the ZONE XEDIT option. The Assembler code is optimized for fixed record formats, so a search in a variable record file may be slightly slower.

When an occurrence of 'old string' is found, ECHANGE displays a prompt and waits until the user presses one of the following PF keys:

- 1 [Yes] – replace the first occurrence of the target; redo action

after PF4.

- 2 [Next] – search for the next occurrence of the target.
- 3 [Quit] – stop the ECHANGE MACRO.
- 4 [Undo] – restore the first occurrence of the target; undo action after PF1.
- 5 [All] – change all following occurrences of the target without prompting.

PF1 and PF4 keys do not change the column pointer, so PF2 must be used to find the subsequent occurrences of 'old string'. When the entire file has been searched, the line pointer is moved to the null End Of File line.

Examples of ECHANGE usage are:

- Selective change of L with LH:

```
ECHANGE /L /LH/
```

- Replacing the character '-' in variable names with the character '_':

```
ECHANGE /-/_
```

- Selective deletion of the group of characters ABC:

```
ECHANGE /ABC
```

INSTALL EXEC

```
/******  
/***  
/*** INSTALL          generate SEARCHER MODULE          *** DG'98 ***/  
/***  
/*****  
/***  SIZE 00043  VER 1.0 MOD 000          ***/  
/*****
```

```
CLRSCRN  
MESSAGE = 'user request'  
SAY ' – Start SEARCHER MODULE generation - reply Y or N'  
PULL REPLY  
IF REPLY = 'Y' THEN
```

```

SIGNAL ERROR
SET CMSTYPE HT
STATE SEARCHER MODULE A
SAVE_RC = RC
SET CMSTYPE RT
IF SAVE_RC = Ø THEN
DO
  SAY ' - SEARCHER MODULE found on disk A'
  SAY ' - Replace SEARCHER MODULE A - reply Y or N'
  PULL REPLY
  IF REPLY ≠ 'Y' THEN
    SIGNAL ERROR
  END
SET CMSTYPE HT
SIGNAL ON ERROR
MESSAGE = 'error when assemble' SEARCHER
ASSEMBLE SEARCHER
ERASE SEARCHER LISTING A
MESSAGE = 'error when load' SEARCHER
LOAD SEARCHER '(' NOMAP NOLIBE
MESSAGE = 'error when genmod' SEARCHER
GENMOD
ERASE SEARCHER TEXT A
SIGNAL OFF ERROR
SET CMSTYPE RT
SAY ' - SEARCHER MODULE generated successfully'
EXIT
ERROR:
  SET CMSTYPE RT
  SAY ' - SEARCHER MODULE not generated due to' MESSAGE

```

ECHANGE XEDIT

```

/*****/
/***                                     ***      ***/
/*** ECHANGE           enhanced XEDIT change          *** DG'98 ***/
/***                                     ***      ***/
/*****/
/***   SIZE 00078  VER 1.0 MOD 000                                     ***/
/*****/

```

```

PARSE ARG ARG_STRING

```

```

PARSE VALUE ARG_STRING WITH '/' OLD_V '/' NEW_V '/'
EXT'/CMD/CURL/FM/FN/FT/LI/LR/MSGL/RECF/Z'
IF LENGTH(OLD_V) + ZONE.1 - ZONE.2 < 2 THEN
DO
  START = MAX(1, LINE.1)

```

```

BGN_P = ZONE.1
END_P = ZONE.2
COL_N = BGN_P - 1
LRECL = LRECL.1
CMD OFF
MSGL OFF
RECF F
RESER CMDLINE.2 HI '>>>-> 1-Yes 2-Next 3-Quit 4-Undo 5-All'
DO FOREVER
  ADDRESS CMS SEARCHER FNAME.1 FTYPE.1 FMODE.1
  IF START = Ø THEN
    LEAVE
  ':'START
  ORIG = 'Y'
  DO FOREVER
    CURS F START COL_N
    READ N T
    PULL TAG PFK .
    IF TAG = 'PFK' THEN
      DO
        IF PFK = '1' | PFK = '4' THEN
          DO
            ':'START
            CL ':'COL_N
            IF PFK = '1' THEN
              IF ORIG = 'Y' THEN
                DO
                  CD LENGTH(OLD_V)
                  IF LENGTH(NEW_V) > Ø THEN
                    CI NEW_V
                    ORIG = 'N'
                  END
                ELSE
                  NOP
                ELSE
                  IF ORIG = 'N' THEN
                    DO
                      IF LENGTH(NEW_V) > Ø THEN
                        CD LENGTH(NEW_V)
                        CI OLD_V
                        ORIG = 'Y'
                      END
                    ELSE
                      NOP
                    ITERATE
                  END
                ELSE
                  IF PFK = '5' THEN
                    CHANGE '/'OLD_V'/'NEW_V'/* *'

```

```

        LEAVE
    END
END
IF PFK = '3' | PFK = '5' THEN
    LEAVE
END
CMD CMDLINE.1
OPERANDS = MSGLINE.1
DO I = 2 TO MSGLINE.0
    OPERANDS = OPERANDS  MSGLINE.I
END
MSGL OPERANDS
RECFM RECFM.1
END

```

SEARCHER ASSEMBLE

```

*****
****                                     ***          ****
**** SEARCHER          search substring in string          ***  DG'98  ****
****                                     ***          ****
*****
****  SIZE 00160  VER 1.0 MOD 000                                     ****
*****
*                                                                 *
SEARCHER CSECT
    USING *,12
    LR   11,14
    MVC  FN(8),8(1)
    MVC  FT(8),16(1)
    MVC  FM(2),24(1)
    MVI  REQUEST,C'F'
    MVC  CONTINUE(4),=X'41418001'
    MVC  BRFRST(2),=X'0700'
    MVC  REXXID(5),=CL5'BGN_P'
    BAL  2,GETORPUT
    LR   10,1
    MVC  REXXID(5),=CL5'END_P'
    BAL  2,GETORPUT
    LR   9,1
    MVC  REXXID(5),=CL5'COL_N'
    BAL  2,GETORPUT
    LR   8,1
    MVC  REXXID(5),=CL5'LRECL'
    BAL  2,GETORPUT
    LR   7,1
    MVC  REXXID(5),=CL5'START'
    BAL  2,GETORPUT

```

```

LR      6,1
ST      6,RECNO
LR      0,7
LR      1,7
SLL     1,3
ST      1,BSIZE
DMSFREE DWORDS=(0),ERR=RET,AREA=HIGH
ST      1,BUFFER
MVI     REQUEST,C'S'
SR      1,1
MVC     REXXID(5),=CL5'START'
BAL     2,GETORPUT
MVI     REQUEST,C'F'
MVC     BRFRST(2),BRSCND
MVC     REXXID(5),=CL5'OLD_V'
BAL     2,GETORPUT
L       15,VALUELEN
BCTR    15,0
STC     15,CLC+1
NEXTBLK EQU *
LA      0,EXTPLIST
LA      1,FSCB
ICM     1,8,=X'02'
SVC     202
DC      AL4(1)
LTR     15,15
BNZ     FREEMAIN
FULLBLK EQU *
L       0,PUTINBUF
SRDL    0,32
DR      0,7
LR      0,1
L       1,BUFFER
LA      2,1
NEXTREC EQU *
LA      3,0(1,9)
LA      4,0(1,10)
CONTINUE LA 4,1(1,8)
MVC     CONTINUE(10),=5X'0700'
BCTR    3,0
BCTR    4,0
NEXTPOS EQU *
CLC     0(0,4),REXXVAL
BE      FOUND
BXLE    4,2,NEXTPOS
LA      6,1(6)
AR      1,7
BCT     0,NEXTREC
L       15,RECNO

```



```

        LA      15,8(15)
        ST      15,RECNO
        B       NEXTBLK
FOUND   EQU    *
        MVI    REQUEST,C'S'
        MVI    VALUELEN+3,X'05'
        SR     4,1
        LA     1,1(4)
        MVC    REXXID(5),=CL5'COL_N'
        BAL    2,GETORPUT
        LR     1,6
        MVC    REXXID(5),=CL5'START'
        BAL    2,GETORPUT
FREEMAIN EQU *
        L      0,BSIZE
        SRL    0,3
        L      1,BUFFER
        DMSFRET DWORDS=(0),LOC=(1)
RET     EQU    *
        BR     11
GETORPUT EQU *
        CVD    1,WORKA
        UNPK   REXXVAL(5),WORKA+5(3)
        OI     REXXVAL+4,X'F0'
        LA     0,REXXPARM
        LA     1,COMMAND
        ICM    1,8,=X'02'
        SVC    202
        DC     AL4(1)
        LTR    15,15
        BM     RET
BRFRST  BR     0
        L      15,VALUELEN
        BCTR   15,0
        EX     15,PACK
        CVB    1,WORKA
BRSCND  BR     2
PACK    PACK   WORKA(8),REXXVAL(0)
WORKA   DS     D
COMMAND DC     CL8'EXECCOMM'
REXXPARM DC     A(COMMAND)
        DC     F'0'
        DC     F'0'
        DC     A(REQBLOK)
REQBLOK DC     A(0)
        DS     F
REQUEST DS     CL1
RETCODE DC     X'00'
        DC     H'0'

```

```

BUFSIZE  DC    F'64'
          DC    A(REXXID)
NAMELEN  DC    F'5'
          DC    A(REXXVAL)
VALUELEN DC    F'5'
EXTPLIST EQU   *
          DC    A(COMMVERB)
          DC    A(Ø)
          DC    A(Ø)
          DC    A(Ø)
COMMVERB DC    CL8'SUBCOM'
          DS    ØF
FSCB     EQU   *
          DC    CL8'DMSXFLRD'
FN       DC    CL8' '
FT       DC    CL8' '
FM       DC    CL2' '
          DC    H'Ø'
BUFFER   DC    A(Ø)
BSIZE    DC    A(Ø)
RECFM    DC    CL1'F'
          DC    X'2Ø'
          DC    H'ØØ'
PUTINBUF DC    AL4(Ø)
RECNO    DC    AL4(1)
NOREC    DC    AL4(8)
          DC    A(Ø)
          DC    A(Ø)
REXXID   DS    CL5
REXXVAL  DS    CL64
          END    SEARCHER

```

ECHANGE PREPARATION

INSTALL EXEC should be used to generate executable code. As a matter of convenience, a synonym of ECHANGE may be defined in the PROFILE XEDIT as shown:

```
SYN EC MACRO ECHANGE
```

Then, instead of typing 'ECHANGE /A/B/', the user would input 'EC /A/B/'.

Dobrin Goranov
Information Services Co (Bulgaria)

© Dobrin Goranov 1998

Easy listing and purging of spool files

INTRODUCTION

The following EXECs will show you a list of all the reader or printer files on your system, allowing you to sort by size, user, etc, and purge individual files or whole batches by prefix command.

Also included is QSPOOL EXEC, which will return the percentage of spool space used to another EXEC or extract a two-line summary of spool usage from the 'QUERY ALLOC SPOOL' command.

ALLSPOOL EXEC

The format is:

```
ALLSPOOL [type] [status]
```

This will give you a file containing all spool files of 'type' and 'status', where:

- 'type' can be RDR or PRT, or an abbreviation of READER or PRINTER, defaulting to RDR.
- 'status' can be specified as 'NOOPEN', otherwise open files will be shown as well as closed ones.

Unwanted files can then be purged by a prefix macro.

```
/* Look at all reader or print files on the system */
*****/

arg type open .
if abbrev('READER',type) then type = 'RDR'
if abbrev('PRINTER',type,1) then type = 'PRT'
if open='NOOPEN' then push type 'NOOPEN'
else push type 'ALL'
address command 'XEDIT LALLSPLL' type 'A3 (WIDTH 132 PROFILE ALLSPOOL'
if rc=16 then
do
  say 'Class D needed to manipulate other machines'' spool files'
  exit 16
```

```

end
if rc=0 then address CMS 'HELP ALLSPOOL'
exit

```

ALLSPOOL XEDIT

```

/*****
* Look at all reader or print files on the system      *
* Called initially by ALLSPOOL EXEC as an XEDIT profile *
*****/

address XEDIT
if queued()=0 then pull porr sel /* parm stacked on initial call */
else arg porr sel , /* parm will be passed on refresh call*/
/* do we want to see open files? */
if sel='NOOPEN' then open = 0
else open = 1
if porr=' ' then porr = 'RDR' /* default to reader files */
if porr='RDR' & porr='PRT' then
do
'QQ 4' /* ALLSPOOL EXEC should give help for this */
exit
end

'SET MSGLINE ON 2 2 OVERLAY'
'SET MSGM OFF'
'SET RECFM V'
'SET PREFIX NULLS'
'SET COLOR PREFIX RED UNDER'
/* set prefix macro for purging files */
'SET PREFIX SYNONYM ↪ PREFPUR'
'SET PREFIX SYNONYM ↻ PREFPUR'
'EXTRACT /LSCREEN'
swidth = lscreen.6
'SET V 1' swidth-7
'SET CURL ON 3'
'SET STAY ON'
'SET RESERVED 1 N' /* lose file heading */
'SET RESERVED 2 OFF'
'SET TOFEOF OFF'
'SET SHADOW OFF'
'SET SCALE OFF'

'SET PF3 QUIT'
'SET PF15 QUIT'

'SET LINEND OFF' /* for multi-command PF key settings */
/* refresh */
'SET PF2 ALL#TOP#SET MSGM OFF#DEL *#MACRO ALLSPOOL' porr sel

```

```

'SET PF14 ALL#TOP#SET MSGM OFF#DEL *#MACRO ALLSPOOL' porr sel
/* sort into type and then date */
'SET PF4 TOP#SORT * A 17 19 39 52'
'SET PF16 TOP#SORT * A 17 19 39 52'
/* sort into descending size */

'SET PF5 TOP#SORT * D 21 28'
'SET PF17 TOP#SORT * D 21 28'
/* sort into user then date */
'SET PF6 TOP#SORT * A 1 8 17 19 39 52'
'SET PF18 TOP#SORT * A 1 8 17 19 39 52'
'SET LINEND ON'

'SET RESERVED -4 H PF2=refresh PF3=quit PF4=type and date
PF5=size PF6=user'
'SET RESERVED -3 H Put ↵ or ↵ in prefix area and press Enter to purge
files'

/*****
* now get list of files from CP *
*****/
bsize = 128*1024 /* buffer size */
'SET MSGM ON'
lines = diagrc(8,'Q' porr 'ALL',bsize)
parse value lines with . reperr . 16 lines
/* reperr=1 means reply too big for buffer */
/* reperr>1 means we have a serious error */
/* and will cancel for bad syntax */

/******
* add lines to the file *
*****/
do until spline=''
  parse value lines with spline '15'x lines
  if spline=↵ then 'I' spline
end

/* alter value of "bsize" if the following message is seen */
if reperr then 'MSG Reply truncated after' bsize/1024'K'

'SET MSGM OFF' /* stop unnecessary messages */
':1'
'EXTRACT /CURLINE' /* keep CP's header on screen */
parse value curline.3 with head1 .
if head1='NO' then
do /* no files|? */
  say curline.3
  exit
end
if head1='ORIGINID' then 'QUIT 16' /* need class D */

```

```

if head1≠'OWNERID' then
do
  'CP Q' porr 'ALL'
  'QQUIT'
end

'SET RESERVED 1 H      ' left(curline.3,width-7)
'DEL'                  /* remove header from file,      */
                      /* leaving it as a reserved line */

                      /*****
* remove leading zeros from SIZE *
                      *****/

'ZONE 20 27'
do 7
  'C/ 0/ /*'
end
'ZONE 1 *'

                      /*****
* remove open files if not wanted *
                      *****/

if ¬open then
do
  'ALL / OPEN- /*'
  if rc=0 then
do
  'TOP'
  'DEL *'
  'ALL'
end
end
end

'EXTRACT /SIZE'
'SET MSGM ON'
pct = qspool()
'SET RESERVED 2 H      ' pct'% spool area used          ' size.1
porr
'files'
'TOP'
'SORT * A 17 19 39 52' /* type and then date */
exit

```

PREFPUR XEDIT

```

/*****
* PREFPUR - Prefix macro to put purge spool files - ALLSPOOL EXEC *
*****/

```

```

arg prefix operand pline parm .
if parm='?' | prefix='?' then call help
'EXTRACT /LINE'                /* get current line number */
current = line.1
parse source . . . . . name .    /* command name          */
if operand='CLEAR' then exit     /* leave if a clear only */
select
  when length(name)=1 then      /* one file only          */
  do
    ':' pline
    nlines = 1
    call spoolpur
  end
  when length(name)=2 then      /* block prefix           */
  do
    'EXTRACT /PENDING BLOCK' name ':0:'pline '/'
    if pending.0=0 then        /* start of block already found */
    do
      ':'pending.1
      'SET PENDING OFF'
      nlines = pline - pending.1 + 1 /* no. of lines          */
      ':'pending.1 /* locate start of block */
      call spoolpur
    end
    else ':'pline 'SET PENDING BLOCK' name
  end
  otherwise
  do
    'SET SCOPE ALL'
    ':'pline 'SET PENDING ERROR' left(name,5)
  end
end
':'current /* relocate to old line */
'N'
exit

/******
* Purge file(s) requested *
*****/

spoolpur:
'EXTRACT /FTYPE' /* filetype is RDR for reader file and PRT for print */
if ftype.1='PRT' then
do
  cpf = 'P'
  dispf = 'print'
end
if ftype.1='RDR' then
do
  cpf = 'R'
  dispf = 'reader'

```

```

end
do nlines          /* purge no. of files required */
  'EXTRACT /CURLINE'
  parse value curline.3 with vmid fno . . . . . fdate .
  if fdate='OPEN-'
  then 'REPL ***' left(vmid,8) dispf 'file' fno 'is still open ***'
  else
  do
    'EXECIO 1 CP (SKIP STRING PUR' vmid cpf fno
    if rc=0 then 'REPL ***' left(vmid,8) dispf 'file' fno 'purged ***'
  end
  'N'
end

return

/*****/
/* help routine */
/*****/

help:
address command 'HELP CMS PREFPUR'
exit

```

QSPPOOL EXEC

```

/*****/
*   Query spool slot usage                               *
*-----*
* Can be called as a REXX function   :   percent = qspool() *
*   to get a value in EXEC or EXEC2:   EXEC QSPPOOL RC   *
*   or to get info. at terminal      :   QSPPOOL         *
/*****/

arg parm .
parse source . func .          /* how we were called */

      /* only use the stuff we want from Q ALLOC */
parse value diag(8,'Q ALLOC SPOOL') with . 'SUMMARY' av us pct '%' .
pct = strip(pct)

if func='FUNCTION' then return pct      /* REXX used "qspool()"      */
if parm='RC' then exit pct             /* EXEC used "EXEC QSPPOOL RC" */
say us 'slots used out of' av
say pct'% spool area used'
exit

```


PREFPUR HELPCMS

↵ and ↵↵ prefix macros

Used in ALLSP00L EXEC to purge the spool file(s) indicated.

↵ will purge one file

↵↵ will purge a block of files up to the next ↵↵

"↵ ?" will display this help information.

NB the commands

```
SET PREFIX SYNONYM ↵ PREFPUR
```

```
SET PREFIX SYNONYM ↵↵ PREFPUR
```

must be issued before use of these macros.

They are already in ALLSP00L XEDIT.

If the filetype is RDR, reader files will be purged.

If the filetype is PRT, print files will be purged.

John Illingworth
Systems Engineer
Wm Morrison Supermarkets (UK)

© Xephon 1998

Call for papers

Why not share your expertise and earn money at the same time? *VM Update* is looking for REXX EXECs, macros, program code, etc, that experienced VMers have written to make their life, or the lives of their users, easier. We will publish it (after vetting by our expert panel) and send you a cheque when the article is published. Articles can be of any length and can be sent or e-mailed to Robert Burgess at any of the addresses shown on page 2. Why not call now for a free copy of our *Notes for contributors*?

A full screen console interface

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.

CSC is a full screen console interface for Disconnected Service Machines (DSM). It operates in two different modes – current and browse.

Current mode is entered (if authorized) when you invoke CSC, or after entering the CURRENT command. Any browse command will switch you into browse mode.

CSC was initially developed for VM/ESA 1.2.1 and is now running under VM/ESA 2.1.0. It is used mainly as a monitoring tool.

THE SCREEN

Figure 1 shows a sample CSC screen with the following lines:

- 1 The identification line shows the VM node on the left, the identification title, and the user-id on the right.
- 2 The header line shows the fields being displayed, the operating mode, some CSC options, and the current date and time, as of the last screen update.
- 3 From the third line down to the separator line (4), there is the output area that contains data supplied by the CSC service machine. The following fields are displayed:
 - a one-character prefix that identifies the originating DSM.
 - the date the message was received by CSC.
 - the time the message was received by CSC.
 - the user-id of the originating DSM.

```

(1) | VMNODE          Identification title          USERID  |
(2) |>___Time___Current___C_____98/05/09___14:23:02 |
(3) |R>14:00:01 DMTSPT071E VTAM error on link DDL1 LUNAME LUDDL1 REQ=22 |
    |R 14:00:01 DMTSPT972I Intervention required on printer link DDL1 |
    |R 14:00:02 DMTSPT058I File 2726 (2726) processing terminated on link |
    |> 14:22:53          *** ** * Restart of Data File * *** ** |
    |                   |
    |                   |
    |                   |
(4) |_____ |
(5) |===> _ |
(6) |          PF 1=Hlp 3=End 4=Top 5=Bot 6=Rep 7=Bwd 8=Fwd 9=Cur 10=Shf |
    |_____ |

```

Figure 1: Sample CSC screen

- the message text.

The date, time, and user-id fields are optional.

- 4 The third line from the bottom is a separator line.
- 5 The command line, where all local and CSC commands are entered.
- 6 The bottom line is the prompt/message line which displays the PF keys settings and all CSC messages.

HOW IT WORKS

CSC uses the Single Console Image Facility (SCIF) and its own service machine to collect the data. This means that the CSC service machine must be defined as the secondary user of all DSMs to be controlled.

When a message is received, it is checked against the rules defined in the configuration file, and written to the log file.

DEVELOPMENT STAGE

CSC is still being developed and there are some elements missing that might be nice to have; however, it is stable.

ASSEMBLING AND INSTALLATION

The service machine

Create a virtual machine with a CP directory entry similar to:

```
USER CONSOLES PSWD 4M 16M G
IPL CMS PARM AUTOCR
MACHINE ESA
CONSOLE 0009 3215
SPOOL 000C 3505 *
SPOOL 000D 3525 A
SPOOL 000E 1403 A
LINK MAINT 190 190 RR
LINK MAINT 19E 19E RR
LINK MAINT 19D 19D RR
LINK MAINT 31A 31A RR
IUCV ALLOW
MACHINE ESA 1
MDISK 0191 3390 1 20 VM0001
MDISK 0192 3390 1 5 VM0002
```

The IUCV statement is required to allow the CSC service machine to establish IUCV sessions with the CSC user program. A read/write mini-disk, accessed as A, is required to hold the log file, and a second mini-disk is recommended to load the CSC code.

The user program (CSCUSR)

Use the following statements to generate the user program:

```
GLOBAL MACLIB DMSGPI DMSOM HCPGPI
HLASM CSCUSR
LOAD CSCUSR (CLEAR RLDSAVE
GENMOD CSCUSR (ALL
```

To start a session enter **CSCUSR CONSOLES**, assuming that **CONSOLES** is the name of the CSC service machine, or rename the **CSCUSR** module to the name of the service machine and start it by entering its name with no parameters.

CSCUSR ASSEMBLE

```

        TITLE 'CSCUSR - CSC User Program'
        MACRO
&LABEL EXEC &WHERE
        LCLC &ADC
&ADC SETC '@&WHERE'(1,8)
&LABEL L R11,&ADC          Load routine address
        BASR R14,R11      Execute
        MEND
        MACRO
&LABEL ERROR &MSG
&LABEL LA R1,&MSG        Address error message
        ST R11,ERROSV11  Save our base register
        LA R11,ERROR     Address error routine
        BASR R14,R11     Execute
        MEND
        SPACE 3
*-----*
*
* CSCUSR Register usage
*
* R6 Length of data entered (includes 3270 prefix)
* R7 Pointer to screen Data Stream
* R8 Pointer to IUCV buffer
* R9 IUCV Parameter List
*
* R10 Base - Data area
* R11 Base - Independent routines
* R12 Base - Common code
*-----*
CSCUSR START X'014000'
CSCUSR RMODE ANY
        PRINT NOGEN
        USING CSCUSR,R12      Base for common code
        USING CSCDATA,R10    Base for Data area
        USING IPARML,R9      IUCV Parameter List
        STM R14,R12,12(R13)
        L R10,ACSCDATA      Address CSC Data area
        B INIT
ACSCDATA DC A(CSCDATA)
&DATE SETC '&SYSDATC'(1,4)'/'. '&SYSDATC'(5,2)'/'. '&SYSDATC'(7,2)
        DC C'&DATE &SYSTIME'
        DC C' Copyright CSC Inc, 1997'
        DS 0H
        SPACE 3
*
* Init
*
*

```

```

INIT      SR      R2,R2
          ST      R2,CSCRC          Clear return code
          ST      R2,CSCECB        Initialize ECB
          EXEC    CONFIG            Configuration and set-up
          OI      CSCFLG02,CSCWAIT  Get ready to accept work
          HNDIUCV SET,NAME=CSCNAME,EXIT=CSCSTX
          LTR     R15,R15
          BZ      INIT200          Good, IUCV enable
          ERROR  ERRIUCVS
          B       CLOSE
          SPACE
INIT200   OI      CSCFLG01,HNDIUCVS  Remember it
          LA      R9,CSCPARMC        Address IUCV Parameter List
          MVC     IPVMID,CSCTRGID    Move target-id (service machine)
          MVC     IPUSER(8),CSCNAME  Keep CMS happy
          CMSIUCV CONNECT,NAME=CSCNAME,PRMLIST=CSCPARMC,EXIT=CSCCNX
          LTR     R15,R15
          BZ      INIT300          IUCV Exit defined
          ERROR  ERRIUCVC
          B       CLOSE
          SPACE
INIT300   OI      CSCFLG01,CMSIUCVC  Remember it
          CONSOLE OPEN,PATH=CSCPATH,EXIT=CSCOPX,          *
            BUFFER=(CONSBUFF,CQYSIZE)
          LTR     R15,R15
          BZ      INIT400          Console Fullscreen ready
          ERROR  ERRCNLSLO
          B       CLOSE
          SPACE
INIT400   OI      CSCFLG01,CNSLOP    Remember it
          USING  CQYSECT,R1
          LA      R1,CONSBUFF        Address Console Query buffer
          TM      CQYDQRFL,CQYDQREC  Extended Data Stream supported?
          BZ      INIT410
          OI      CSCFLG01,CNSLEDS   Yes, set option
INIT410   LA      R0,80
          CH      R0,CQYDQRCL        Number of screen columns
          BNE     INIT500            Do not use Alt screen if not 80
          LA      R0,24              Number of lines for model 2
          CH      R0,CQYDQRRW        Compare with lines on screen
          BE      INIT500
          OI      CSCFLG01,CNSLALT    Different, use alternate screen
          TM      CSCFLG01,CNSLNALT   Was NOALT entered?
          BZ      INIT500
          STH     R0,CQYDQRRW        Yes, set size to 24 lines
          NI      CSCFLG01,X'FF'-CNSLALT  Reset option
INIT500   TM      CSCFLG01,CNSLNEDS  Was NOEDS entered?
          BZ      INIT600
          NI      CQYDQRFL,X'FF'-CQYDQREC-CQYDQREH  Reset CQY block options
          NI      CSCFLG01,X'FF'-CNSLEDS  Reset option

```

```

INIT600 DROP R1
        BAS R14, WAIT           Wait for CP to complete IUCV
        TM  CSCFLG02, WORKCV    Expected interrupt?
        BZ  PROC100            No, we are in trouble
        BAS R14, CVPROC         Check if everything is OK
        BAS R14, CHECK          Send initial request <CSC>INI
INIT700 BAS R14, WAIT           Wait for IUCV interrupt
        BAS R14, CVPROC
        TM  CSCFLG04, CVRCVE    Reply received?
        BZ  INIT700            No, wait again
        CONSOLE WRITE, PATH=CSCPATH, OPTIONS=(CLEAR)
        OI  CSCFLG04, SCRDISP   Force initial display
        EXEC PROFILE            Check and process Profile
PROCESS EXEC DISPLAY           Display screen
        BAS R14, WAIT           Wait...
        TM  CSCFLG02, WORKIO
        BZ  PROC100
        BAS R14, IOPROC         Process console input
PROC100 TM  CSCFLG02, WORKID
        BZ  PROC200
        BAS R14, IDPROC         Process IUCV connections
PROC200 TM  CSCFLG02, WORKCV
        BZ  PROC300
        BAS R14, CVPROC         Process IUCV message
PROC300 TM  CSCFLG02, WORKEND
        BZ  PROCESS
        SPACE 3
*
* Close the shop, turn off the lights
*
*
CLOSE  EQU  *
        EXEC RELEASE           Release all allocated storage
        TM  CSCFLG01, CNSLOP    Terminate Console Fullscreen
        BZ  CLOSE100
        NI  CSCFLG01, X'FF'-CNSLOP
        CONSOLE CLOSE, PATH=CSCPATH
        LTR R15, R15           Any problems?
        BZ  CLOSE100
        C   R15, THREE         Allow RC=3 from CLOSE
        BE  CLOSE100
        ERROR ERRCNLC          Display error message
        SPACE
CLOSE100 TM  CSCFLG01, CMSIUCVC  Terminate IUCV session
        BZ  CLOSE200
        NI  CSCFLG01, X'FF'-CMSIUCVC
        CMSIUCV SEVER, NAME=CSCNAME, PRMLIST=CSCPARMC
        LTR R15, R15
        BZ  CLOSE200
        ERROR ERRIUCVE

```

```

SPACE
CLOSE200 TM   CSCFLG01,HNDIUCVS      Make CMS happy
          BZ   RETURN
          NI   CSCFLG01,X'FF'-HNDIUCVS
          HNDIUCV CLR,NAME=CSCNAME
          LTR  R15,R15
          BZ   RETURN
          ERROR ERRIUCVT
          SPACE
RETURN   L    R15,CSCRC              Load our return code
          ST   R15,16(,R13)          Store it into save area
          LM   R14,R12,12(R13)      Restore everything
          BR   R14                  Go back...
          SPACE 3

*
* Wait for something to do
*
*
WAIT     EQU   *                    Wait...
          WAITECB ECB=CSCECB
          XC   CSCECB,CSCECB        Clear ECB
          OI   CSCFLG02,CSCWAIT     Get ready to do more work
          BR   R14                  Check what happened
          SPACE 3

*
* CONSOLE Exit Routine
*
*
CSCOPX   EQU   *                    CONSOLE Exit Routine
          STM  R14,R12,12(R13)
          LA   R0,CSCOPX-CSCUSR
          LR   R12,R15
          SR   R12,R0                Restore our base address (code)
          L    R10,ACSCDATA          Restore our base address (data)
          OI   CSCFLG02,WORKIO      Remember we had an interrupt
OPX900   TM   CSCFLG02,CSCWAIT
          BZ   OPX910
          NI   CSCFLG02,X'FF'-CSCWAIT Post ECB once if required
          OI   CSCECB,CSCPOST
OPX910   LM   R14,R12,12(R13)
          BR   R14                  Return
          SPACE 3

*
* HNDIUCV Exit Routine
*
*
CSCSTX   EQU   *                    HNDIUCV Exit Routine
          STM  R14,R12,12(R13)
          LA   R0,CSCSTX-CSCUSR
          LR   R12,R15

```


	SR	R12,R0	
	L	R10,ACSCDATA	Establish addressability
	OI	CSCFLG02,WORKID	Remember we had an interrupt
	LA	R0,1	*T* Create trace entry
	BAS	R14,TRACE	*T*
	TM	CSCFLG02,CSCWAIT	
	BZ	STX900	
	NI	CSCFLG02,X'FF'-CSCWAIT	Post ECB once if required
	OI	CSCECB,CSCPOST	
STX900	LM	R14,R12,12(R13)	
	BR	R14	
	SPACE	3	
	*		
	* CMSIUCV Exit Routine		
	*		
	*		
CSCCNX	EQU	*	CMSIUCV Exit Routine
	STM	R14,R12,12(R13)	
	LA	R0,CSCCNX-CSCUSR	
	LR	R12,R15	
	SR	R12,R0	
	L	R10,ACSCDATA	Establish addressability
	OI	CSCFLG02,WORKCV	Remember we had an interrupt
	LR	R9,R2	Address IUCV Parameter List
	LA	R0,2	*T* Create trace entry
	BAS	R14,TRACE	*T*
	TM	CSCFLG03,CVFIRST	
	B0	CNX200	
	OI	CSCFLG03,CVFIRST	First time routine was invoked
	CLI	IPTYPE,IPTYPCC	Must be connection complete
	BE	CNX100	
	OI	CSCFLG03,CVCCERR	No, set error option
	B	CNX900	
	SPACE		
CNX100	OI	CSCFLG03,CVCC	Target accepted the connection
	B	CNX900	
	SPACE		
CNX200	TM	CSCFLG03,CVCC	Was Connection Complete received
	BZ	CNX900	No, error already set, forget it
	CLI	IPTYPE,IPTYPMNP	Is this an incoming message?
	BNE	CNX400	
	OI	CSCFLG04,CVMSGIN	Yes, set option
	LA	R0,1	
	A	R0,CVMSGQ	
	ST	R0,CVMSGQ	Count queued messages
	B	CNX900	
	SPACE		
CNX400	CLI	IPTYPE,IPTYPRNP	Did last SEND complete?
	BNE	CNX500	
	NI	CSCFLG04,X'FF'-CVSEND	Yes, reset pending SEND

```

      B      CNX900
      SPACE
CNX500 CLI  IPTYPE,IPTYPSV      Was Connection Severed?
      BNE   CNX600
      OI    CSCFLG04,CVSEVER  Yes, set option, close the shop
      B      CNX900
      SPACE
CNX600 OI    CSCFLG04,CVERROR  One day we will code this
*      B      CNX900
      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 3

```

*

* Process Console I/O

*

*

```

IOPROC EQU  *      Process Console I/O
      ST    R14,IOPRSV14
      NI    CSCFLG02,X'FF'-WORKIO  Reset option
      OI    CSCFLG04,SCRDISP      Refresh screen, reset keyboard
      CONSOLE READ,PATH=CSCPATH,BUFFER=(READBUFF)
      LR    R6,R0      Length of data entered
      LA    R0,3      *T* Create trace entry
      BAS   R14,TRACE  *T*
      LTR   R15,R15
      BZ    IOPR500      No problems, process input
      C     R15,STOLEN  Stolen by another application
      BE    IOPR900      Yes, ignore, will be refreshed
      ERROR ERRIOPR      Display error message
      B     CLOSE      Close the shop
      SPACE
IOPR500 CLI  READATTN,NOAID      No-operation
      BE    IOPR900      Yes, ignore it
      BAS   R14,COMMAND      Process it
IOPR900 L    R14,IOPRSV14
      BR    R14      Return
      SPACE 3

```

*

* Process IUCV connections (should not happen)

*

*

```

IDPROC EQU  *      Process IUCV connections
      ST    R14,IDPRSV14
      NI    CSCFLG02,X'FF'-WORKID  Reset option
      TM    CSCFLG03,CVCC      Are we connected?

```

	B0	IDPR100	Yes, keep going
	ERROR	ERRIDPR	Display error message
	B	CLOSE	Close the shop
	SPACE		
IDPR100	CMSIUCV	SEVER,NAME=CSCNAME,PRMLIST=CSCPARMC	
IDPR900	L	R14,IDPRSV14	
	BR	R14	Return
	SPACE	3	
	*		
	*	Process IUCV interrupts	
	*		
	*		
CVPROC	EQU	*	Process IUCV interrupts
	ST	R14,CVPRSV14	
	NI	CSCFLG02,X'FF'-WORKCV	Reset option
	TM	CSCFLG03,CVCCERR	Any communication error?
	BZ	CVPR100	
	LA	R15,100	Yes, close the shop and bye bye
	ERROR	ERRCVPR	
	B	CLOSE	
	SPACE		
CVPR100	TM	CSCFLG04,CVSEVER	Was connection severed?
	BZ	CVPR200	
	LA	R15,101	Yes, nothing we can do
	ERROR	ERRCVPR	
	B	CLOSE	
	SPACE		
CVPR200	TM	CSCFLG04,CVERROR	Any unexpected interrupt?
	BZ	CVPR300	
	LA	R15,102	
	ERROR	ERRCVPR	
	B	CLOSE	
	SPACE		
CVPR300	TM	CSCFLG04,CVMSGIN	Any message queued?
	BZ	CVPR900	No, done
	NI	CSCFLG04,X'FF'-CVMSGIN	Process them, one at a time
CVPR400	L	R1,CVMSGQ	
	LTR	R1,R1	Anything left
	BZ	CVPR900	No, all done
	BCTR	R1,0	Decrement counter
	ST	R1,CVMSGQ	Store new value
	LA	R9,CSCPARMC	Address IUCV Parameter List
	LA	R0,4	*T* Create trace entry
	BAS	R14,TRACE	*T*
	L	R2,CVBUFBEG	Buffer address
	L	R3,CVBUFLEN	Buffer size in bytes
	MVI	IPFLAGS1,X'00'	Reset all IUCV flags
	IUCV	RECEIVE,PRMLIST=CSCPARMC,BUFFER=(R2),BUFLEN=(R3)	
	BZ	CVPR500	Check for errors
	SR	R15,R15	

	IC	R15,IPRCODE	Load IPRCODE
	ERROR	ERRCPRV	Display error message
	B	CVPR400	Life goes on
	SPACE		
CVPR500	OI	CSCFLG04,CVRCVE	We received a message
	BAS	R14,INPROC	Process received data
	B	CVPR400	
	SPACE		
CVPR900	L	R14,CVPRSV14	
	BR	R14	
	SPACE	3	
	*		
	*	Process Console command	
	*		
	*		
COMMAND	EQU	*	Process Console command
	USING	RTVSECT,R3	
	ST	R14,COMMSV14	
	OI	CSCFLG04,SCRTRL	Refresh all screen
	MVC	DSPMSG,DSPMSGD	Move default message
	XC	DSPCMD,DSPCMD	Clear command line
	MVI	DSPCMD,IC	Move cursor to command line
	MVI	DSPMSG,X'60'	Move default attributes
	MVI	EDSMSC,X'F1'	Move default colour
	MVC	EDSMSC,DSPMSGD	Same for EDS screen
	MVC	EDSCMD,DSPCMD	
COMM100	L	R1,RTRVECUR	Address current RTV entry
	LH	R3,RTVLEN-RTVSECT(,R1)	Get length of current entry
	LA	R3,RTVDATA-RTVSECT(R3,R1)	Possible address of next entry
	LA	R0,RTVDATA-RTVSECT(R3,R6)	Calculate its end address
	C	R0,RTRVEEND	Do we have space to add entry?
	BNH	COMM110	
	L	R3,RTRVEBEG	No, go back to the beginning
COMM110	LR	R0,R3	Copy address of new entry
	SR	R0,R1	Offset to previous one
	STH	R0,RTVPLUS-RTVSECT(,R1)	Store offsets
	STH	R0,RTVLESS	
	STH	R6,RTVLEN	Store length of current entry
	LR	R1,R6	Copy length to work register
	BCTR	R1,0	Prepare to EXecute
	EX	R1,COMMRTVM	Move input data to RTV buffer
COMM200	LA	R3,PFKTABLE	Address Local PF Key Table
COMM210	CLC	READATTN,3(R3)	Compare with key code
	BE	COMM220	Found it, process
	LA	R3,8(,R3)	Address next entry
	CLI	0(R3),X'FF'	Check for End-Of-Table
	BNE	COMM210	
	B	COMM300	
	SPACE		
COMM220	L	R15,4(,R3)	Key found, load routine address

	BASR	R14,R15	Execute routine
COMM300	LA	R0,READBUFF(R6)	End address of data entered
	ST	R0,CMSEPLE	Store into Extended Plist
	LA	R1,READDATA	Address data (after 3270 prefix)
COMM310	CR	R1,R0	End of input data?
	BNL	COMM700	Yes, not local, send it to CSC
	CLI	Ø(R1),C' '	Locate first non blank
	BNE	COMM400	
	LA	R1,1(,R1)	Advance pointer
	B	COMM310	Keep checking
	SPACE		
COMM400	CLI	Ø(R1),C'&&'	Command redisplay requested?
	BNE	COMM420	No, keep going
	OI	CSCFLG02,CMDREDSP	Remember to redisplay after
	MVI	Ø(R1),C' '	Erase the "&" from command
COMM410	LA	R1,1(,R1)	Get next non blank
	CR	R1,R0	Check for end of input
	BNL	COMM700	Found it, only blanks after &
	CLI	Ø(R1),C' '	Check for blank
	BE	COMM410	Skip them all
COMM420	LR	R3,R1	Save address of first non blank
COMM430	LA	R1,1(,R1)	Advance pointer
	CR	C1,R0	Still valid address
	BNL	COMM500	No, end of input data reached
	CLI	Ø(R1),C' '	Locate next blank (delimiter)
	BNE	COMM430	
COMM500	LA	R0,1(,R1)	After the space is the...
	ST	R0,CMSEPLA	... extended argument address
	SR	R1,R3	Length of command
	LA	R0,L'USRCOMM	Maximum length of command
	CR	R1,R0	
	BH	COMM700	Too long, send it to CSC
	BCTR	R1,Ø	Prepare to EXecute
	EX	R1,COMMMVC	Move to work area
	OC	USRCOMM,BLANKS	Quick convert to uppercase
	LA	R3,LCLTABLE	Address Local Commands Table
COMM600	CLC	USRCOMM(1),1(R3)	Check for one byte command
	BE	COMM690	Found it, process
	LA	R3,16(,R3)	Address next entry
	CLI	Ø(R3),X'FF'	Check for End-Of-Table
	BNE	COMM600	Not yet
COMM610	LA	R3,16(,R3)	Go to full commands table
	CLI	Ø(R3),X'FF'	Check for End-Of-Table
	BE	COMM700	Not local, send it to CSC
	SR	R0,R0	Required by next IC
	IC	R0,Ø(,R3)	Load minimum abbreviation
	CR	R1,R0	Is it enough
	BL	COMM610	No, keep searching table
	EX	R1,COMMCLC	Compare command name
	BNE	COMM610	Not this one, search table

COMM690	L	R15,12(,R3)	Found it, load routine address
	BASR	R14,R15	Execute routine
	B	COMM800	
	SPACE		
COMM700	BAS	R14,SEND	Send to CSC for processing
COMM800	SR	R0,R0	
	ST	R0,RTRVELST	Reset last referenced entry
	L	R3,RTRVECUR	Address "old" current RTV entry
	LH	R0,RTVLEN	Get length of data
	LR	R1,R3	Save entry address
	AH	R3,RTVPLUS	Address "new" current RTV entry
COMM810	CH	R0,RTVLEN	Check data length
	BNE	COMM820	Not the same, keep new entry
	LH	R2,RTVLEN	Get length of new entry
	LA	R0,7	Ignore 3270 DS prefix (6 bytes)
	SR	R2,R0	Prepare to EXecute
	BNM	COMM830	Found some input data, check it
	MVC	RTVDATA-RTVSECT(1,R1),RTVDATA	Only PA/PF key, copy it
	B	COMM900	But do not add new entry
	SPACE		
COMM820	LA	R0,6	Length of 3270 DS prefix
	CH	R0,RTVLEN-RTVSECT(,R1)	Any data in "old" current?
	BNE	COMM890	Yes, keep entry
	SH	R1,RTVLESS-RTVSECT(,R1)	No, address previous entry
	ST	R1,RTRVECUR	Possible new current
	LH	R0,RTVLEN-RTVSECT(,R1)	Get length of data
	B	COMM810	Check previous against "new"
	SPACE		
COMM830	EX	R2,COMMRTVC	Compare contents
	BE	COMM900	Same, do not add it to RTV
COMM890	ST	R3,RTRVECUR	Save current entry address
COMM900	TM	CSCFLG02,CMDREDSP	Command redisplay requested (&)?
	BZ	COMM990	No, done
	NI	CSCFLG02,X'FF'-CMDREDSP	Yes, reset option
	OI	CSCFLG04,SCRTRL	Refresh bottom of screen
	L	R1,RTRVELST	It is zero, move data from RTV
	LTR	R1,R1	Valid entry?
	BNZ	COMM910	Yes, use it
	L	R1,RTRVECUR	No, use current entry
COMM910	LH	R2,RTVLEN-RTVSECT(,R1)	Load length of data
	LA	R0,READDATA-READBUFF+1	Length of DS prefix plus one
	SR	R2,R0	Length of input data minus one
	EX	R2,COMMDSM	Move data to command line
	MVI	DSPCMD,IC	Insert cursor
	MVC	EDSCMD,DSPCMD	Copy also to EDS screen
COMM990	L	R14,COMMSV14	
	BR	R14	
	SPACE		
COMMRTVC	CLC	RTVDATA+6(*-*),RTVDATA-RTVSECT+6(R1)	Compare RTV entries
COMMRTVM	MVC	RTVDATA(*-*),READBUFF	Move data to the RTV buffer

```

COMMVC  MVC  USRCOMM(*-*),0(R3)      Move command to work area
COMMCLC CLC  USRCOMM(*-*),1(R3)      Compare command name
COMMDSPM MVC  DSPCMD+1(*-*),RTVDATA-RTVSECT+READDATA-READBUFF(R1)
        SPACE 3
*
* Local commands
*
*
ENDCMD  EQU  *                        END (Pf03 or PF15 keys only)
        OI  CSCFLG02,WORKEND          ... let's go home...
        B   COMM900                   Do not update Retrieve buffer
        SPACE 3
REPCMD  EQU  *                        REPEAT (PF06 or PF18 or =)
        L   R3,RTRVELST               Address last retrieved command
        LTR  R3,R3
        BNZ  REPC100                  None...
        L   R3,RTRVECUR               Repeat last entered command
REPC100 LH   R6,RTVLEN                Get length of data
        LTR  R1,R6                    Is retrieve buffer empty?
        BZ   REPC200                  Yes, display error message
        BCTR R1,0                      Prepare to EXecute
        EX   R1,REPMVC                Move command to console buffer
        TM   CSCFLG02,CMDREDSP        Command redisplay requested (&)
        BZ   COMM100                  No, re-process command
        NI   CSCFLG02,X'FF'-CMDREDSP  Yes, reset option
        OI   CSCFLG04,SCRTRL          Refresh bottom of screen
        L   R1,RTRVECUR               Address "old" current entry
        AH   R1,RTVPLUS-RTVSECT(,R1)  Address "new" current entry
        LH   R2,RTVLEN-RTVSECT(,R1)  Load length of data
        LA   R0,READDATA-READBUFF+1  Length of DS prefix plus one
        SR   R2,R0                    Length of input data minus one
        EX   R2,REPMVCC               Move data to command line
        MVI  DSPCMD,IC                 Insert cursor
        MVC  EDSCMD,DSPCMD             Copy also to EDS screen
        B   COMM100                   Re-process command
        SPACE
REPC200 LA   R1,MSGE0200              Retrieve Buffer is empty
        BAS  R14,MSGDISP              Display message
        NI   CSCFLG02,X'FF'-CMDREDSP  Clear redisplay option
        B   COMM990                   That's all
        SPACE
REPMVC  MVC  READBUFF(*-*),RTVDATA
REPMVCC MVC  DSPCMD+1(*-*),RTVDATA-RTVSECT+READDATA-READBUFF(R1)
        SPACE 3
RTVFPFK EQU  *                        RETRIEVE (PF11 or PF23) Forward
        BAS  R14,RETFWD               Locate next RTV entry
        BAS  R14,RETMOVE              Move it to command line
        B   COMM900                   Do not update Retrieve buffer
        SPACE 3
RETFWD  EQU  *                        Locate next RTV entry

```

	L	R3,RTRVECUR	Address current entry
	SR	R0,R0	
	C	R0,RTRVELST	Any entry already retrieved?
	BE	RETBWD	No, just get current entry
	LH	R0,RTVPLUS	Total length of current entry
	AR	R0,R3	End address of current entry
	L	R3,RTRVELST	Address last referenced entry
RETF100	C	R3,RTRVECUR	Is it the current entry?
	BE	RETF300	Yes, that's the last entry
	AH	R3,RTVPLUS	Address next entry
	C	R3,RTRVECUR	Is it the current entry?
	BE	RETF200	Yes, our last chance
	LR	R1,R3	Copy address
	AH	R1,RTVPLUS	End address of next entry
	CR	R3,R0	Any overlap with current entry?
	BNL	RETF200	No, check this entry
	C	R1,RTRVECUR	Is it a real overlap?
RETF200	BH	RETF300	Yes, not valid, no more data
	LH	R1,RTVLEN	Length of entry
	LTR	R1,R1	Is it zero?
	BZ	RETF300	Yes, Retrieve buffer is empty
	LA	R2,READDATA-READBUFF	Length of 3270 DS prefix
	CR	R1,R2	Compare
	BNH	RETF100	No data, just PA / PF key
	ST	R3,RTRVELST	We found it, save entry address
	BR	R14	
	SPACE		
RETF300	SR	R0,R0	Entry not found
	C	R0,RTRVELST	Any entry already retrieved?
	BNE	RETF400	
	LA	R1,MSGE0200	No, Retrieve buffer is empty
	BAS	R14,MSGDISP	
	B	COMM990	That's all
	SPACE		
RETF400	L	R3,RTRVELST	Yes, no more data to retrieve
	BAS	R14,RETMOVE	Move it to command line
	LA	R1,MSGE0202	Also display error message
RETF500	BAS	R14,MSGDISP	
	B	COMM900	Do not update Retrieve buffer
	SPACE	3	
RTVCMD	EQU	*	RETRIEVE Retrieve command
	SPACE		
RTVBPFK	EQU	*	RETRIEVE (PF12 or PF24) Backward
	BAS	R14,RETBWD	Locate previous RTV entry
	BAS	R14,RETMOVE	Move it to command line
	B	COMM900	Do not update Retrieve buffer
	SPACE	3	
RETBWD	EQU	*	Locate previous RTV entry
	L	R3,RTRVECUR	Address current entry
	SR	R0,R0	

	C	R0,RTRVELST	Any entry already retrieved?
	BE	RETB200	No, must be current or nothing
	LH	R0,RTVPLUS	Total length of current entry
	AR	R0,R3	End address of current entry
	L	R3,RTRVELST	Address last referenced entry
RETB100	LR	R1,R3	
	SH	R3,RTVLESS	Address previous entry
	CR	R3,R0	Any overlap with current entry?
	BNL	RETB200	No, check this entry
	C	R1,RTRVECUR	Is it a real overlap?
	BH	RETB300	Yes, not valid, no more data
RETB200	LH	R1,RTVLEN	Length of entry
	LTR	R1,R1	Is it zero?
	BZ	RETB300	Yes, end of buffer reached
	LA	R2,READDATA-READBUFF	Length of 3270 DS prefix
	CR	R1,R2	Compare
	BNH	RETB100	No data, just PA / PF key
	ST	R3,RTRVELST	We found it, save entry address
	BR	R14	
	SPACE		
RETB300	SR	R0,R0	Entry not found
	C	R0,RTRVELST	Any entry already retrieved?
	BNE	RETB400	
	LA	R1,MSGE0200	No, Retrieve buffer is empty
	BAS	R14,MSGDISP	
	B	COMM990	That's all
	SPACE		
RETB400	L	R3,RTRVELST	Yes, no more data to retrieve
	BAS	R14,RETMOVE	Move it to command line
	LA	R1,MSGE0202	Also display error message
RETB500	BAS	R14,MSGDISP	
	B	COMM900	Do not update Retrieve buffer
	SPACE	3	
RETMOVE	EQU	*	Move data from RTV to command
	LH	R1,RTVLEN	Load length
	LA	R0,READDATA-READBUFF+1	Length of DS prefix plus one
	SR	R1,R0	Length of input data minus one
	EX	R1,RETMVC	Move data to command line
	LA	R1,DSPCMD+1(R1)	Address end of data
	MVI	0(R1),IC	Insert cursor
	MVC	EDSCMD,DSPCMD	Copy also to EDS screen
	BR	R14	
	SPACE		
RETMVC	MVC	DSPCMD(*-*),RTVDATA+READDATA-READBUFF	
	SPACE	3	
CLEARCMD	EQU	*	CLEAR (CLEAR key only)
	B	COMM900	Don't worry, refresh screen
	SPACE	3	
CMSCMD	EQU	*	CMS Execute any CMS command
	USING	NUCON,R0	

	L	R1,CMSEPLA	Address parameters
CMS200	C	R1,CMSEPLE	Anything at all
	BNL	CMS400	No, enter CMS Subset mode
	CLI	Ø(R1),C' '	Is it just blanks?
	BNE	CMS500	No, process CMS command
	LA	R1,1(,R1)	
	B	CMS200	
	SPACE		
CMS400	CMSCALL	PLIST=CMSSS	Enter CMS Subset
	B	COMM800	
	SPACE		
CMS500	IPK		Insert PSW key into R2
	ST	R2,CSCWORK	Save PSW key temporarily
	SR	R2,R2	Zero register
	SPKA	Ø(R2)	Store PSW key of zero
	L	R1,CMSEPLA	Address parameters
	L	RØ,CMSEPLE	End address
	SR	RØ,R1	Length of entered parameters
	L	15,ASCANN	Build tokenized PL for CMS
	BASR	14,15	Call CMS routine
	L	R2,CSCWORK	Load previous PSW key
	SPKA	Ø(R2)	Store it into PSW
	CMSCALL	PLIST=CMSP, EPLIST=CMSEPL, CALLTYP=SUBCOM	
	LTR	R15,R15	Check return code
	BZ	CMS600	It is zero, forget it
	LA	R1,MSGEØ204	Tell user it was not zero
CMS600	BAS	R14,MSGDISP	
	B	COMM800	
	DROP	RØ	
	SPACE	3	
HELPCMD	EQU	*	HELP Invoke CSC Help
	L	R1,CMSEPLA	Address parameters
HELP100	C	R1,CMSEPLE	Anything at all
	BNL	HELPPFK	No, display CSC HELP menu
	CLI	Ø(R1),C' '	Is it just blanks?
	BNE	HELP500	No, process entered HELP command
	LA	R1,1(,R1)	
	B	HELP100	
	SPACE		
HELPPFK	CMSCALL	PLIST=HELPP, COPY=NO	Invoked by PF key or no operands
	B	COMM800	
	SPACE		
HELP500	LA	RØ,L'HELPTYPE	Our Help type
	SR	R1,RØ	Create space in command
	MVC	HELPSAVE(L'HELPTYPE),Ø(R1)	Save part of command
	MVC	Ø(L'HELPTYPE,R1),HELPTYPE	Move our Help type
	ST	R1,HELPEPLA	Build Extended Parameter List
	L	R1,CMSEPLE	
	ST	R1,HELPEPLE	
	CMSCALL	PLIST=HELPP, EPLIST=HELPEPL, COPY=NO	Invoke Help

```

L      R1,HELPEPLA          Restore part of entered command
MVC   Ø(L'HELPTYPE,R1),HELPSAVE ... for Retrieve Buffer
B      COMM8ØØ
SPACE
DROP  R3
SPACE 3

*
* Process data received from the Service machine
*
*
INPROC EQU *                Process IUCV received data
ST     R14,INPRSV14
OI     CSCFLGØ4,SCRDISP     Refresh the screen
L      R3,IPBFADR1         End of received data
ST     R3,CVBUFEND         Save it
L      R8,CVBUFBEG         Begin of received data
INPR1ØØ L R3,Ø(,R8)         Length of command
AR     R3,R8               End of command
LA     R8,4(,R8)           Skip length field
SPACE
CLC    COMMSCR,Ø(R8)       <CSC>SCR command?
BNE    INPR2ØØ            No, try something else
LA     R8,L'COMMSCR(,R8)   Skip command name
L      RØ,SCBUFBEG         Address Screen buffer
LR     R2,R8
SR     R3,R2               Length of screen Data Stream
LR     R1,R3               Make fields of equal length
MVCL  RØ,R2               Move Data Stream to buffer
ST     RØ,SCBUFEND         Store end address of Data Stream
LR     R3,R2
B      INPR9ØØ
SPACE
INPR2ØØ CLC COMMMSG,Ø(R8)   <CSC>MSG command?
BNE    INPR3ØØ
LA     R8,L'COMMMSG(,R8)
LR     R1,R3               End of command
SR     R1,R8               Length of message
BZ     INPR29Ø            If zero, forget it
LA     RØ,L'DSPMSG
CR     R1,RØ
BNH    INPR21Ø
LR     R1,RØ               If too big, truncate
INPR21Ø BCTR R1,Ø          Prepare to EXecute
XC     DSPMSG,DSPMSG       Clear Message field
EX     R1,INPR2MVC         Move new message
MVI    DSPMSG,X'F8'        Display message highlighted
MVI    EDSMSG,X'F2'        Display colour message in red
XC     EDSMSG,EDSMMSG      Clear Message field (EDS)
MVC    EDSMSG,DSPMSG       Copy message text
INPR29Ø OI CSCFLGØ4,SCRTRL Refresh bottom of screen

```

```

      B      INPR900
      SPACE
INPR300 CLC   COMMHDR,0(R8)      <CSC>HDR command?
      BNE   INPR400
      LA    R8,L'COMMHDR(,R8)
      LR    R1,R3
      SR    R1,R8              Header length
      BZ    INPR390           If zero forget it
      LA    R0,DSPHDR
      CR    R1,R0
      BNH   INPR310
      LR    R1,R0              If too big truncate
INPR310 BCTR  R1,0
      EX    R1,INPR3MVC       Move new header
      MVC   EDHDR(EDSHDR),DSPHDR Copy to EDS screen
INPR390 B      INPR900
      SPACE
INPR400 CLC   COMMTTL,0(R8)    <CSC>TTL command?
      BNE   INPR500
      LA    R8,L'COMMTTL(,R8)
      XC    DSPTITLE,DSPTITLE Clear old title
      XC    EDSTITLE,EDSTITLE Clear old title (EDS)
      LR    R1,R3              End of data
      SR    R1,R8              Title length
      BZ    INPR490           If zero, forget it
      LA    R0,L'DSPTITLE     Length of title field
      CR    R1,R0
      BNH   INPR410
      LR    R1,R0              Too big, truncate new title
INPR410 LR    R2,R0              Length of Title field
      SR    R2,R1              Subtract length of new title
      SRL   R2,1              Divide by 2
      LA    R0,DSPTITLE       Address of Title field
      AR    R2,R0              Centre new title
      BCTR  R1,0              Prepare to EXecute
      EX    R1,INPR4MVC       Move new title
      MVC   EDSTITLE,DSPTITLE Copy to EDS screen
INPR490 B      INPR900
      SPACE
INPR500 CLC   COMMALM,0(R8)    <CSC>ALM command?
      BNE   INPR550
      OI    CSCFLG04,SCRALM    Set Alarm option
      B      INPR900
      SPACE
INPR550 CLC   COMMACL,0(R8)    <CSC>ACL command?
      BNE   INPR600
      LA    R1,MSGE0206        Not implemented (yet)
      BAS   R14,MSGDISP
      B      INPR900
      SPACE

```

INPR600	CLC	COMMDCL,0(R8)	<CSC>DCL command?
	BNE	INPR650	
	LA	R1,MSGE0208	Not implemented (yet)
	BAS	R14,MSGDISP	
	B	INPR900	
	SPACE		
INPR650	CLC	COMMCL,0(R8)	<CSC>MCL command?
	BNE	INPR700	
	OI	CSCFLG04,SCRTRL	Refresh bottom of screen
	LA	R8,L'COMMCL(,R8)	
	LR	R1,R3	End of data
	SR	R1,R8	Data length
	BNZ	INPR670	Length not zero, process it
	TM	CSCFLG02,PROFRUN	Is Profile being processed?
	BO	INPR900	Yes, nothing to redisplay
	L	R1,RTRVELST	Length zero, move data from RTV
	LTR	R1,R1	Valid entry?
	BNZ	INPR660	Yes, use it
	L	R1,RTRVECUR	No, use current entry
INPR660	LH	R2,RTVLEN-RTVSECT(,R1)	Load length of data
	LA	R0,READDATA-READBUFF+1	Length of DS prefix plus one
	SR	R2,R0	Length of input data minus one
	EX	R2,INPR6MVC	Move data to command line
	MVI	DSPCMD,IC	Insert cursor
	MVC	EDSCMD,DSPCMD	Copy also to EDS screen
	B	INPR690	
	SPACE		
INPR670	LA	R0,L'DSPCMD-1	Length of Command Line (less IC)
	CR	R1,R0	
	BNH	INPR680	
	LR	R1,R0	Too big, truncate
INPR680	BCTR	R1,0	Prepare to EXecute
	EX	R1,INPR7MVC	Move data to Command Line
	MVI	DSPCMD,IC	Position cursor
	MVC	EDSCMD,DSPCMD	Copy to EDS screen
INPR690	B	INPR900	
	SPACE		
INPR700	CLC	COMMCNN,0(R8)	<CSC>CNN command?
	BNE	INPR800	
	OI	CSCFLG04,SCRTRL	Refresh bottom of screen
	LA	R8,L'COMMCNN(,R8)	
	MVC	DSPCNN,0(R8)	Move connected node name
	LA	R1,DSPCNN+L'DSPCNN	Address end of field
INPR710	MVI	0(R1),C'_'	
	BCTR	R1,0	Check for trailing blanks
	CLI	0(R1),C' '	Replace them all
	BE	INPR710	
	MVC	EDSCNNA(L'EDSCNNA+L'EDSCNN),BLANKS	Clear everything
	MVC	EDSCNNA(L'EDSCNN),0(R8)	Move connect node to EDS screen
	LA	R1,EDSCNN+L'EDSCNN	Address end of field plus reset

```

INPR720 MVI  0(R1),C'_'
        BCTR R1,0           Check for trailing blanks
        CLI  0(R1),C' '    Replace them all
        BE   INPR720
        MVC  1(L'EHIRESET,R1),EHIRESET Reset extended highlight
        B    INPR900
        SPACE
INPR800 CLC  COMMPRT,0(R8)   <CSC>PRT command?
        BNE  INPR820
        EXEC PRINT          Process Print command
        B    INPR900
        SPACE
INPR820 CLC  COMMWRT,0(R8)  <CSC>WRT command?
        BNE  INPR840
        EXEC WRITE          Process Write command
        B    INPR900
        SPACE
INPR840 CLC  COMMRSK,0(R8)  <CSC>RSK command?
        BE   INPR900
        SPACE
        LA   R1,MSGE0210    Unknown command, beep beep
        BAS  R14,MSGDISP
        B    INPR910        Life goes on
        SPACE
INPR900 LR   R8,R3          Address new command
        C    R8,CVBUFEND    Anything left?
        BL   INPR100        Yes, process it
INPR910 TM   CSCFLG02,PROFRUN Is Profile being processed?
        BZ   INPR990
        EXEC PROFILE        Yes, process all file
INPR990 L    R14,INPRSV14
        BR   R14            No, return
        SPACE 3
INPR2MVC MVC  DSPMSG(*-*),0(R8) Message
INPR3MVC MVC  DSPHDR(*-*),0(R8) Header
INPR4MVC MVC  0(*-*,R2),0(R8) Title (centred)
INPR6MVC MVC  DSPCMD+1(*-*),RTVDATA-RTVSECT+READDATA-READBUFF(R1)
INPR7MVC MVC  DSPCMD+1(*-*),0(R8) Command line
        SPACE 3
*
* Check user-id
*
*
CHECK   EQU   *            Check user-id
        ST   R14,CHCKSV14
        LA   R0,6           *T* Create trace entry
        BAS  R14,TRACE      *T*
        MVI  IPFLAGS1,X'00' Clear IUCV flags
        LA   R2,CHCKLEN     Send INI with terminal info
        IUCV SEND,PRMLIST=CSCPARMC,BUFFER=CHCKSEND,BUFLEN=(R2), *

```

```

                TYPE=1WAY
        BZ      CHCK900
        SR      R15,R15                IUCV SEND error
        IC      R15,IPRCODE           Load IPRCODE
        ERROR   ERRCOMM
        B       CLOSE
CHCK900      OI      CSCFLG04,CVSEND   SEND in progress
        L      R14,CHCKSV14
        BR     R14
        SPACE 3
*
* Send data to Service CSC
*
*
SEND        EQU     *                Send data to Service CSC
        ST      R14,SENDSV14
        LA      R0,5                  *T* Create trace entry
        BAS     R14,TRACE             *T*
        MVI     IPFLAGS1,X'00'       Clear all IUCV flags
        LA      R2,L'READSEND(,R6)
        IUCV    SEND,PRMLIST=CSCPARMC,BUFFER=READSEND,BUFLEN=(R2), *
                TYPE=1WAY
        BZ      SEND100
        SR      R15,R15                IUCV SEND error
        IC      R15,IPRCODE           Load IPRCODE
        ERROR   ERRCOMM
        B       CLOSE
        SPACE
SEND100     OI      CSCFLG04,CVSEND   SEND in progress
        NI      CSCFLG04,X'FF'-SCRDISP Do not refresh, wait for reply
        L      R14,SENDSV14
        BR     R14
        SPACE 3                      *T*
*
*
* Create trace entries
*
*
TRACE       EQU     *                *T* Create trace entries
        L      R1,TRACEPTR           *T* Only for testing
        SLL    R0,16                 *T* Destroys registers R0 and R1
        ST     R0,0(,R1)             *T* R14 is the return address
        LR     R0,R15                *T*
        SLL    R0,16                 *T*
        SRL    R0,16                 *T*
        O      R0,0(,R1)             *T*
        ST     R0,0(,R1)             *T*
        L      R0,IPPATHID           *T*
        ST     R0,4(,R1)             *T*
        LA     R1,8(,R1)             *T*
        C      R1,TRACEEND           *T*

```

```

        BL    TRACE100          *T*
        L     R1,TRACEBEG      *T*
TRACE100 ST    R1,TRACEPTR     *T*
        BR    R14              *T*
        SPACE 3

*
* Display screen messages
*
*
MSGDISP EQU    *              Display screen messages
        BCTR  R1,0             Address length byte
        SR    R2,R2           Required by next IC
        IC    R2,0(,R1)       Load length of message
        LA    R0,L'DSPMSG     Maximum message length
        CR    R0,R2           Just in case
        BNL   MSGD100
        LR    R2,R0           Too long, truncate message
MSGD100 BCTR  R2,0             Clear Message field
        XC    DSPMSG,DSPMSG   Copy message text
        EX    R2,MSGDMVC     Display message highlighted
        MVI   DSPMSG,X'F8'    Display colour message in red
        MVI   EDSMSG,X'F2'   Clear Message field (EDS)
        XC    EDSMSG,EDSMSG  Copy message text
        MVC   EDSMSG,DSPMSG
        OI    CSCFLG04,SCRTRL+SCRALM Refresh bottom of screen, beep
        BR    R14
        SPACE
MSGDMVC MVC   DSPMSG(*-*),1(R1)
        SPACE 3

*
* Display error messages
*
*
ERROR    EQU    *              Display error messages
        USING ERROR,R11
        LR    R2,R1
        ST    R15,CSCRC       Store return code
ERROR010 L    R1,0(,R2)
        CR    R1,R15          Locate return code
        BE    ERROR020
        C     R1,FFFFFFFF     End of search
        BE    ERROR020
        LA    R2,12(,R2)
        B     ERROR010
        SPACE
ERROR020 L    R1,4(,R2)       Load routine address
        L     R2,8(,R2)       Load message address
        BCTR  R2,0             Point to message length
        BR    R1              Execute routine
        SPACE

```



```

ERROR100 APPLMSG TEXTA=(R2),DISP=ERRMSG
        B      ERROR900
        SPACE
ERROR200 APPLMSG TEXTA=(R2),DISP=ERRMSG,SUB=(DECA,CSCRC)
        B      ERROR900
        SPACE
ERROR400 APPLMSG TEXTA=(R2),DISP=ERRMSG,SUB=(HEXA,(CSCRC,2))
        B      ERROR900
        SPACE
ERROR600 APPLMSG TEXTA=(R2),DISP=ERRMSG,SUB=(CHARA,(CSCNAME,8))
        B      ERROR900
        SPACE
ERROR800 APPLMSG TEXTA=(R2),DISP=ERRMSG,SUB=(CHARA,(CSCTRGID,8))
        B      ERROR900
        SPACE
ERROR900 L      R11,ERROSV11
        BR     R14
        SPACE
        DROP  R11
        SPACE 3
*
* Console display
*
*
DISPLAY  EQU    *
        USING DISPLAY,R11
        ST     R14,DISPSV14
        TM     CSCFLG04,SCRDISP      Refresh required?
        BZR    R14                    No, done
        SPACE
        NI     CSCFLG04,X'FF'-SCRDISP Reset option
        LA     R1,DIAG000C           Work area for DIAG
        DIAG   R1,R0,X'000C'
        MVC    DSPDATE(2),DIAG000C+6 Edit date to yy/mm/dd format
        MVI    DSPDATE+2,C'/'
        MVC    DSPDATE+3(5),DIAG000C
        MVC    DSPTIME,DIAG000C+8    Move time
        L      R7,DSBUFBEGBEG        Address buffer to build DS
        TM     CSCFLG01,CNSLEDS      Extended Data Stream supported?
        BO     DISP100                Yes, let's be fancy
        MVC    0(DSPLENT,R7),DSPBUFF Move top of screen
        LA     R7,DSPLENT(,R7)       Adjust pointer
        B      DISP110
        SPACE
DISP100  MVC    EDSDATE,DSPDATE      Copy date
        MVC    EDSTIME,DSPTIME       Copy time
        MVC    0(EDSLENT,R7),EDSBUFF Move top of screen in colours
        LA     R7,EDSLENT(,R7)       Adjust pointer
DISP110  LR     R0,R7                Copy for MVCL
        L      R2,SCBUFBEGBEG        Address screen buffer (from DVM)

```

	L	R3,SCBUFEND	
	SR	R3,R2	Length of screen Data Stream
	LR	R1,R3	
	MVCL	R0,R2	Copy to our buffer
	LR	R7,R0	Adjust pointer
	TM	CSCFLG01,CNSLEDS	Extended Data Stream supported?
	BO	DISP120	Yes, let's be fancy
	MVC	0(DSPLENC,R7),DSPTRL	Move middle screen, restore MDT
	LA	R7,DSPLENC(,R7)	Adjust pointer
	B	DISP130	
	SPACE		
DISP120	MVC	0(EDSLENC,R7),EDSTRL	Middle screen for EDS
	LA	R7,EDSLENC(,R7)	Adjust pointer
DISP130	LR	R0,R7	Address end of screen DS
	S	R0,DSBUFBEQ	Calculate length
	ST	R0,SCRLN1	Save length without trailer
	TM	CSCFLG01,CNSLEDS	Extended Data Stream supported?
	BO	DISP200	Yes, let's be fancy
	MVC	0(DSPLENB,R7),DSPCMD	Move bottom screen
	LA	R7,DSPLENB(,R7)	Adjust pointer
	B	DISP210	
	SPACE		
DISP200	MVC	0(EDSLENB,R7),EDSCMD	Move bottom screen
	LA	R7,EDSLENB(,R7)	
DISP210	ST	R7,DSBUFEND	Save it
	S	R7,DSBUFBEQ	
	ST	R7,SCRLN2	Length of total screen
	L	R2,DSBUFBEQ	Address buffer (again)
	L	R3,SCRLN1	Length of top and middle screens
	TM	CSCFLG04,SCRTRL	Do we need the bottom too
	BZ	DISP300	No, display it
	NI	CSCFLG04,X'FF'-SCRTRL	Yes, reset option
	L	R3,SCRLN2	Load total length
DISP300	TM	CSCFLG04,SCRALM	Alarm requested?
	BZ	DISP400	
	OI	0(R2),ALARM	Yes, activate alarm
DISP400	CONSOLE	WRITE,PATH=CSCPATH,BUFFER=((R2),(R3)),OPTIONS=(W)	
	NI	DSPBUFF,X'FF'-ALARM	
	LA	R0,7	*T* Create trace entry
	BAS	R14,TRACE	*T*
	LTR	R15,R15	
	BZ	DISP900	No problems, return
	C	R15,STOLEN	Console stolen (RC=32)?
	BNE	DISP800	No, we got a problem
DISP500	L	R3,SCRLN2	Display all screen (Erase/Write)
	TM	CSCFLG01,CNSLALT	Alternate screen size supported?
	BO	DISP600	Yes, so use it
	CONSOLE	WRITE,PATH=CSCPATH,BUFFER=((R2),(R3)),OPTIONS=(EW)	
	B	DISP700	
	SPACE		

```

DISP600  CONSOLE WRITE,PATH=CSCPATH,BUFFER=((R2),(R3)),OPTIONS=(EWA)
DISP700  LA      R0,8                *T* Create trace entry
        BAS    R14,TRACE           *T*
        LTR    R15,R15
        BZ     DISP900             All done, return
DISP800  ERROR  ERRDSP             Display error message
        B      CLOSE               Close the shop
        SPACE
DISP900  NI     CSCFLG04,X'FF'-SCRALM  Reset alarm option
        L      R14,DISPSV14
        BR     R14
        SPACE
        DROP   R11
        SPACE 3

*
* Process PRINT command
*
*
PRINT    EQU    *                    Print Log file
        USING PRINT,R11
        ST     R14,PRWRSV14
        ST     R8,PWBEG             Save command address
        LA     R0,PWFIRST           Load First bit
        N      R0,L'COMMWRT(,R8)    Check against Control word
        LA     R8,L'COMMWRT+4(,R8)  Address first record
        BZ     PRINT100             It is not first block, continue
        LA     R0,CPCLOSEB          Address CP command (Before)
        LA     R1,L'CPCLOSEB        Length of CP command
        DIAG   R0,R1,X'0008'        Close any previous Print file
PRINT100 L      R5,0(,R8)           Load length prefix
        LA     R4,4                 Length of length prefix!!!
        SR     R5,R4                Length of record (data)
        C      R5,PRINTMAX          Compare with maximum print line
        BNH   PRINT200
        L      R5,PRINTMAX          Too long, truncate
PRINT200 LA     R4,4(,R8)           Address of record
        PRINTL (R4),(R5),CC=NO      Print record
        LTR    R15,R15              Any problems?
        BNZ   PRINT800              Yes, check error
        A      R8,0(,R8)            No, address next record
        CR     R8,R3                Do we have one
        BL     PRINT100             Yes, process new record
        L      R1,PWBEG             No, address command again
        LA     R0,PWLAST            Is it the last block?
        N      R0,L'COMMWRT(,R1)
        BZ     PRINT300             No, request another block
        LA     R0,CPCLOSEA          Address CP command (After)
        LA     R1,L'CPCLOSEA        Length of CP command
        DIAG   R0,R1,X'0008'        Close and name Print file
        B      INPR900              Go back

```

```

SPACE
PRINT300 LA R6,6+L'PRINTCMD+4 Length of request
MVC READDATA(L'PRINTCMD),PRINTCMD Move command to buffer
ST R15,READDATA+L'PRINTCMD Zero Control word
BAS R14,SEND Send request to CSC
B PRINT900
SPACE
PRINT800 LA R6,6+L'PRINTCMD+4 Length of request
MVC READDATA(L'PRINTCMD),PRINTCMD
ST R15,READDATA+L'PRINTCMD Store return code
LA R0,CPCLOSEA Address CP command (After)
LA R1,L'CPCLOSEA Length of CP command
DIAG R0,R1,X'0008' Close and name Print file
L R1,PWBEG Address command
LA R0,PWLAST Is it the last block?
N R0,L'COMMWRT(,R1)
BZ PRINT820 No, cancel command
LA R1,MSGE0212
BAS R14,MSGDISP Display error message
B PRINT900
SPACE
PRINT820 BAS R14,SEND Send request to CSC
B PRINT900
SPACE
PRINT900 L R14,PRWRSV14
BR R14
SPACE
DROP R11
SPACE 3
*
* Process WRITE command
*
*
WRITE EQU * Write PrintLog file
USING WRITE,R11
ST R14,PRWRSV14
ST R8,PWBEG Save command address
LA R0,PWFIRST Load First bit
N R0,L'COMMWRT(,R8) Check against Control word
LA R8,L'COMMWRT+4(,R8) Address first record
BZ WRITE100 It is not first block, continue
FSERASE FSCB=PRINTLOG First block, erase old file
FSOPEN FSCB=PRINTLOG,FORM=E Open new file

```

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

Fernando Duarte
Analyst (Canada)

© F Duarte 1998

Recording changes in CMS files

GENERAL DESCRIPTION

When more than one person is allowed to change programs, and you don't want to buy one of the full function software change management systems available, this little XEDIT macro may be the right solution for you.

It only does one thing – recording in the program source which CMS user made the last changes to the file. To enable the function to be performed without the users having to learn new commands, it is achieved by defining synonyms for the usual commands for writing the files.

The information about the last change is stored as a comment in the source – so you need to make some assumptions about the type of the file that defines the programming language. In the following XEDIT macro it is assumed that CMS files with filetypes beginning ASS or ASM are Assembler files, those beginning with PL/1, PL1, PL/I, or PLI are PL/I code, those beginning with COB are COBOL sources, and those beginning with JOB are VSE job control. A filetype of COPY defines a COBOL copybook.

You can, of course, change the corresponding parts of the macro to adapt it to your needs or to other programming languages.

The placement of the comment line is also dependent on the language type:

- Assembler code is commented before the last END or MEND statement.
- PL/I gets its comment line at the end of the program.
- For COBOL the comment is placed after the PROGRAM-ID line or at the end of the code for a COBOL copybook.
- Jobs are marked after the '// JOB' card.

If the file is already marked as changed, then the comment line is only

updated with the new user.

If you want to enable the users to turn the process of marking the files on and off, you can provide the macros OFFSYN and ONSYN that are given below.

INSTALLATION

You have to define the following synonyms by setting them into the PROFILE XEDIT of all of your users:

```
SET SYNONYM FILE 4 MACRO FILESAVE PFILE
SET SYNONYM SSAVE 2 MACRO FILESAVE SAVE
SET SYNONYM FFILE 2 MACRO FILESAVE FILE
SET SYNONYM SAVE 4 MACRO FILESAVE PSAVE
```

These synonyms replace the standard ones that are defined for a standard CMS system:

```
SET SYNONYM FILE 4 Command PFILE
SET SYNONYM SSAVE 2 Command SAVE
SET SYNONYM FFILE 2 Command FILE
SET SYNONYM SAVE 4 Command PSAVE
```

FILESAVE XEDIT

```
/*
/* Writing information about the last user to change a CMS file */
/*
/* The comments that are inserted in the file are dependent on the */
/* filetype of the CMS file. */
/*
```

```
trace off
parse upper arg params
'EXTRACT/FTYPE/LINE'
actline = LINE.1
ft = FTYPE.1
ft3 = left(ft,3)
ft4 = left(ft,4)
```

```
txtasm = '* LAST CHANGE BY USER:'
txtp11 = ' /* LAST CHANGE BY USER:'
txtcob = ' * LAST CHANGE BY USER:'
txtjob = '/. LAST CHANGE BY USER:'
```

```

select;
  when ft3 = 'ASS' | ft3 = 'ASM' then do
    'BOT'
    if rc = 0 then do
      'STACK 1'
      pull zeile
      szeile = strip(zeile)
      if left(szeile,4) = 'MEND' | left(szeile,3) = 'END' then do
        'UP 1'
        'STACK 1'
        pull zeile
      end
      if left(zeile,41) = txtasm ,
      then 'REPLACE' txtads userid()
      else 'INPUT'  txtads userid()
    end
  end

  when ft3 = 'PL1' | ft3 = 'PLI' | ft4 = 'PL/1' | ft4 = 'PL/I' then do
    'BOT'
    if rc = 0 then do
      'STACK 1'
      pull zeile
      if left(zeile,43) = txtpl1 ,
      then 'REPLACE' txtpl1 userid() '*'
      else 'INPUT'  txtpl1 userid() '*'
    end
  end

  when ft3 = 'COB' | ft4 = 'COPY' then do
    'TOP'
    if ft4 = 'COPY' then 'BOT'
    else '/PROGRAM-ID'
    if rc = 0 then do
      if ft4 = 'COPY' then 'N 1'
      'STACK 1'
      pull zeile
      if left(zeile,47) = txtcob ,
      then 'REPLACE' txtcob userid()
      else do
        if ft4 = 'COPY' then 'UP 1'
        'INPUT' txtcob userid()
      end
    end
  end

  when ft3 = 'JOB' then do
    'TOP'

```

```

'LOCATE %// JOB %'
if rc = 0 then do
  'N 1'
  'STACK 1'
  pull zeile
  if left(zeile,44) = txtjob ,
    then 'REPLACE' txtjob userid()
    else do
      'UP 1'
      'INPUT' txtjob userid()
    end
  end
end
end
otherwise nop
end

':actline

'SET SYNONYM OFF'
command params
'SET SYNONYM ON'

exit

```

OFFSYN XEDIT

/* Reset synonyms of FILESAVE XEDIT to standard */

```

'SET SYNONYM FILE 4 Command PFILE'
'SET SYNONYM SSAVE 2 Command SAVE'
'SET SYNONYM FFILE 2 Command FILE'
'SET SYNONYM SAVE 4 Command PSAVE'

```

ONSYN XEDIT

/* Set synonyms for FILESAVE XEDIT */

```

'SET SYNONYM FILE 4 MACRO FILESAVE PFILE'
'SET SYNONYM SSAVE 2 MACRO FILESAVE SAVE'
'SET SYNONYM FFILE 2 MACRO FILESAVE FILE'
'SET SYNONYM SAVE 4 MACRO FILESAVE PSAVE'

```

Dr Reinhard Meyer (Germany)

© Xephon 1998

September 1994 – August 1998 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	Cross reference	112.40-51
ACCESS	113.37-38, 117.18-24, 121.3-8	Cut and paste	105.14-20
Accounting	106.8-18, 123-42-50, 124.3-22	DASD	125.18-44
ADSORT	137.3-9	DASD space	112.3-11
ALL macros	99.3-9, 104.16-26	Data extract	118.12-37
ALLSPOOL	144.11-17	Date	113.26-36
Attributes	106.3-7	Date handling	102.13-25
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	DATEFUNC	143.7-19
BACKUP	135.6-28, 136.8-19	DCSS	102.39-43, 110.10
Bi-modal	123.18-19	DDR	117.41-51
BLOCKIO	137.40-51, 138.22-34	Decimal	112.20-24, 113.47-49
Bookmanager	98.16-20	Diary	128.9-15
Broadcast	97.25-46	Directory	120.17-32
Browse	109.3-4	Directory maintenance	98.40-51, 101.12-20
Bulletin board	128.34-51, 129.28-51, 130.17-32, 131.9-25	DIRMAINT	98.40, 110.8
Cacheing	132.28-40	Disk information	128.3-8
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
CMS DISTINCT	135.28-34	Dump/restore	137.40-51, 138.22-34
CMS monitor	139.16-26	EBCDIC to ASCII	136.3-7
CMSCALL	138.19-21	ECHANGE	144.3-10
CNTLUNIT	101.3	Edit	101.20-21
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
		FCOPYPRO	143.30-38
		FDISK	121.3-8
		File copying	141.29-41, 143.30-38
		File exchange	119.46-51, 120.44-47
		File list	124.32-47, 124.47-51
		File manager	98.21-31

FILESAVE	144.45-48	Message suppression	102.3-13
File transfer	133.3-9, 139.43-51, 140.10-19	Migration	115.8-9
FILETAPE	139.3-15	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
Find files	107.3-13	Monitor	90.32-39, 115.3-7, 118.3-5
Finder	103.32-34	MOVEFILE	139.3-15
Format	100.31-33	Multicolumn files	101.22-26
FORTH	100.34-51, 101.32-51, 102.44-51, 103.35-51, 104.34-47, 105.20-32	MVS	119.46-51, 120.44-47
FTP	114.12-16, 115.40-51, 116.38-51, 117.28-40, 125.3-15	MXEDIT	138.3-18
FTTERM	133.3-9	Named substrings	133.9-16
Fullscreen	97.12-19, 117.3-17	New files	107.15-24
Full screen console interface	144.18-44	Newsgroups	143.39-51
GCS	123.19-22	NOTE	108.36
Hexadecimal	111.41-51, 112.20-24, 113.47-49	NSS	119.3-14
Horizontal prefix	134.19-28	Object-oriented	129.3-4
HP XEDIT	134.19-28	OfficeVision	108.36-51, 109.43-51, 110.35-51, 111.25-35, 116.3-7
ICCF	139.43-51, 140.10-19	ORDER	116.33-38
INCLUDE	103.25-31	Ordering	97.48-51
INDF	139.16-26	Over-used directories	143.3-6
Intranet	142.9-14	Panel manager	106.44-51, 107.24-41, 108.17-27
Inventory	109.28-38	Peek	108.3-6
IOCP	101.3	PICTURE	135.35-39
Job staging	130.14-17	Pipelines	105.51, 107.13, 108.3-6, 109.3-4, 111.40, 121.28-38
Keys	101.26-31	PLUS	133.29-41
LESS	133.29-41	PostScript	97.19-24
LINK	104.47-51, 113.37-38, 117.18-24, 121.3-8	PR/SM	101.3
Linking spool files	144.11-17	PREPT	136.20-35
LOADLIB	114.26-49	PRFOLIE	138.35-45
Locate	121.19-27, 122.16-17	Printing	113.39-45
Locks	105.10	Printing transparencies	138.35-45
Log-on	110.8-10	PROP	113.3-26
LOGOFF	109.5-7, 112.37-40	PTF	118.38-39, 140.35-51, 141.8-27, 142.14-43, 143.20-29
LOOKVARS	141.5-8	PUNMEM	139.43-51, 140.10-19
LPAR	101.3	Purge	100.25-30
MACLIB	114.26-49, 122.28-30	Purging spool files	144.11-17
Macro	123.18-19	PUT	118.38-39
Mailing lists	143.39-51	QUERY	105.3-9, 106.3-7, 111.3-19
Memos	119.27-45	RAMAC	126.30-32
Menu	125.46-51, 126.7-26, 127.37-39, 134.42-51, 135.40-51, 136.36-51, 137.29-39	RDR	108.3-6, 132.26-27
Merging	122.3-15	RECEIVE	108.3-6
Message forwarding	110.24-33	Recording change	144.45-48

Renaming	125.15-17	System resources	114.3-12
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	Tabulated data	109.38-42
RETRIEVE	112.15-19, 114.50-51	Tag files	128.29-33
REXX	129.5-19	TCP/IP	114.12-16, 115.40-51, 116.38-51, 117.28-40, 118.39-51, 119.15-26, 120.33-43
REXX management	140.20-25	TDSK	109.5-7
REXX tracking	140.35-51, 141.8-27, 142.14-43, 143.20-29	Terminal	109.28-38
RXSQL	105.10	Time	113.26-36
RXUSERFN	134.3-8	Timing	110.3-7
SALT	133.29-41	Toolkit	98.31-39
SAVE	108.28-33	Transfer	129.20-23
Scheduler	131.3-8, 132.40-48, 133.17-28, 134.9-18	Transferring code	141.28
Search	100.6-8, 117.24-27	Translation	129.5-19
Security	120.3-16	TTIME	110.3
SELCOPY	97.46-48	TXTLIB	114.26-49
Server	98.21-31	UCO	142.3-8
SET MORE	99.10-12, 99.21-38, 102.26-39, 103.15-24, 107.3-13, 108.6-16	Unerase	115.33-39
SFS	116.19-32, 121.38-41, 131.26-30	Unix	129.20-23
SFSPACE	143.3-6	Utilities	112.11-14, 130.33-34
Sorting	132.3-8, 137.3-9	VALDATE	142.50-51
Spanned records	100.31-33	Vaulting	99.43-51, 100.8-24
Specified string	135.3-5	VDISK	100.3-5
Split screen	97.3-6	VDUMP	37.40-51, 138.22-34
Spool	105.3-9, 109.7-16, 109.17-27, 111.36-40, 113.45-46	Virtual disks	100.3-5
SQL	106.8-18, 112.24-36	Virtual rooms	104.3-15
SQL/DS	98.3-15, 105.10-13, 136.20-35, 137.10-28, 134.3-8	VM/ESA	99.13-21, 115.8-9, 140.3-9
SQLDBSU	98.3-15	VMDDR	142.44-49
SSM	137.10-28	VMFE2E	120.47-48, 125.44-45
Statistics	122.40-51, 123.23-42, 124.23-32	VMFPLC2	104.26
String	99.39-43, 121.19-27, 122.16-27	VMSES/E	120.47-48
SUBVM	139.43-51, 140.10-19	VREST	37.40-51, 138.22-34
SWAP	133.29-41	VSCS	114.20-25
System	126.3-6	VTIME	110.3
		Web	127.3-9, 129.24-27
		Web site, Melinda Varian	141.42-51
		Web site, Mike Cowlshaw	140.26-34
		Web site, IBM VM	139.27-32
		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 Version 3.0 of its Network Station software for VM/ESA, AS/400, and OS/390. Version 3.0 has integrated 40-bit NC Navigator, converged 3270/5250 emulators and VT Telnet capability, ICA Client Communications protocol, JVM 1.1.4 across the IBM Network Station family, group support via centralized Network Station management, and LPR/LPD streaming support for network printing. The VT supports VT320, VT300, VT220, VT200, VT102, and VT100.

The subset of Navigator 3.0 is an upgrade of the existing Navio NC Navigator browser currently available, which won't run on Release 3.

For further information contact your local IBM representative.

* * *

Available for VM, OS/390, or MVS, Bull has announced the integration of Telecoms Data Systems' TDSLink agent technology into its OpenMaster secure enterprise management software suite. This is to improve OpenMaster's ability to manage MVS applications and SNA networks.

The TDSLink software suite provides agents with standard interfaces for management integration.

TDSLink for SNA provides a graphic representation of the different resource states, and the command interface is secured via user profile definitions and their

associated rights. The views are configurable according to user profiles.

For further information contact:

Bull Information Systems, Computer House, Great West Road, Brentford, Middlesex, TW8 9DH, UK.

Tel: (0181) 568 9191.

Bull HN Information, 2 Wall Street, Technology Park, Billerica, MA 01821-4199, USA.

Tel: (978) 294 6000.

URL: <http://www.bull.com>.

* * *

Software Diversified Services has announced Vital Signs VisionNet, a VM performance monitor providing real-time and batch reporting facilities.

VisionNet provides a single view of the network's performance and retains both current and historical performance information in a central data repository. VisionNet's Exception Monitor monitors performance statistics and sends filtered alerts to the network management system or automated operations system when thresholds are exceeded. Vital Signs also supports VM/XA.

For further information contact:

Software Diversified Services, 5155 East River Road, Minneapolis, MN 55421-1025, USA.

Tel:(612) 571 9000.

URL: <http://www.sdsusa.com>.

* * *



xephon