148

VM

December 1998

In this issue

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 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 £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*.

Printed in England.

How large are your SFS directories?

Sometimes it would be nice to know how large an SFS directory is, but, unless you have a special program, there is no way to find out.

The DS EXEC will show you how large a specific SFS directory is. Just enter DS in front of a directory from DIRLIST and the result will be displayed at the bottom of your screen.

A sample display is shown in Figure 1.

SOURCE CODE

```
/*%Copyright (c) ── Copyright Andreas Brunner ── */
/*====================================================================*
 * Proc.Name  : DS                                                    *
 * Title      : Get the size of an SFS directory                      *
 * Author     : ANDI                                                  *
 * Updater    : ANDI                                                  *
 * Residence  : A                      (P=Public/S=Service/O=Office) *
 * Purpose    : Display the size of SFS Directories.                  *
 * Input      : directory       Name of a Directory                   *
 *──────                                                             *
 *====================================================================*
 *==================* Modification - Statements *====================*
 *====================================================================*
 *========================* End of Header *==========================*/
Address 'COMMAND'                       /* Normally to commands       */
Parse Source penv padr pgn ptyp pmod    /* Who am I?                  */
Parse Upper Arg directory '(' parm .    /* Get the input              */

Main:
Call Process_Input
Call Check_Directory
Call Display_Result
Call Release_Directory
Call Exit

/*====================================================================*
 * Exit                                          S U B R O U T I N E *
 *====================================================================*/
Exit:
Exit

/*====================================================================*
 * Process the Input given                       S U B R O U T I N E *
```

```
ANDI     DIRLIST  AØ  V 319  Trunc=319 Size=58 Line=1 Col=1 Alt=4

Cmd   Fm Directory Name
      A  SFSUSER:ANDI.
      -  SFSUSER:ANDI.ACELOG
      -  SFSUSER:ANDI.APPS
      -  SFSUSER:ANDI.CACHE1997
      F  SFSUSER:ANDI.CEXEC
      G  SFSUSER:ANDI.CREXX
      H  SFSUSER:ANDI.CXEDIT
      -  SFSUSER:ANDI.C37Ø
      -  SFSUSER:ANDI.DELEGATE
      -  SFSUSER:ANDI.DOCU
      -  SFSUSER:ANDI.ESAMIGR
      -  SFSUSER:ANDI.EXEC
      -  SFSUSER:ANDI.EXEC.DIRM
      -  SFSUSER:ANDI.EXEC.SERV
      -  SFSUSER:ANDI.GML
      -  SFSUSER:ANDI.INTERNET
      -  SFSUSER:ANDI.INTERNET.CMSPIP_L
      -  SFSUSER:ANDI.INTERNET.PROFS_L
      -  SFSUSER:ANDI.INTERNET.SFS_L
      -  SFSUSER:ANDI.INTERNET.VM_UTIL
*     -  SFSUSER:ANDI.INTERNET.VMESA_L
      -  SFSUSER:ANDI.INTERNET.VMY2K_L
      -  SFSUSER:ANDI.JUNK
      -  SFSUSER:ANDI.LOGFILE
      -  SFSUSER:ANDI.MISC
      -  SFSUSER:ANDI.MSG
      -  SFSUSER:ANDI.OVVM
      -  SFSUSER:ANDI.OVVM4
      -  SFSUSER:ANDI.PANEL
      -  SFSUSER:ANDI.PIPES
      -  SFSUSER:ANDI.PROFS
      -  SFSUSER:ANDI.REXX
      -  SFSUSER:ANDI.REXXC
      -  SFSUSER:ANDI.REXXCOMPILER
      -  SFSUSER:ANDI.SRDIRECT_TEMP
      -  SFSUSER:ANDI.TABLE

1= Help  2= Refresh   3= Quit   4= Sort(fm)   5= Sort(dir)   6= Auth
DS 15:49:35 Files Blocks Records Megabytes
DS 15:49:35    64   3ØØ8  184218 11.75
====>
                                                        X E D I T  1
```

Figure 1: Example of display

```
 *====================================================================*/
Process_Input:
'MAKEBUF'                              /* Create new Buffer          */
'LISTDIR' directory '(STACK'           /* Try to list the directory  */
If rc ¬= '0' Then Do                   /* It's not a directory       */
  Address XEDIT 'MSG' pgn Time() directory 'is not a Directory'
  If rc ¬= '0' Then Say directory 'is not a SFS Directory'
  Call Exit
End                                    /* End If Then Do             */
'DROPBUF'                              /* Drop the Buffer            */
If parm = 'ALL' Then Call Compute_All_Directories
'PIPE CMS QUERY FILEPOOL CURRENT | Var fp'
Parse Var fp fp filepool
filepool = Strip(filepool)
Select                                 /* Start Select               */
  When directory = Userid()'.' Then    /*                            */
    directory = filepool||directory    /*                            */
  When Pos('.',directory) = '1' Then   /*                            */
    directory = filepool||Userid()||directory
  Otherwise                            /* Called from DIRLIST        */
    directory = directory
End                                    /* End Select                 */
Return
                                       /*                            */
/*====================================================================*
 * ....                                          S U B R O U T I N E *
 *====================================================================*/
Check_Directory:
If Directory_Accessed(directory) Then Do /* Directory is accessed    */
  Call Compute_Directory_Size
End                                    /* End If Then Do             */
Else Do                                /* Directory is not accessed  */
  Call Access_Directory(directory)
  Call Compute_Directory_Size
End                                    /* End Else Do                */
Return
                                       /*                            */
/*====================================================================*
 * ....                                              F U N C T I O N *
 *====================================================================*/
Directory_Accessed:
Arg directory                          /* Read the Input             */
'PIPE CMS QUERY ACCESSED',             /* Issue Query Command        */
 '| Drop First',                      /* Dont use the Header        */
 '| Locate /'directory'/',            /* Watch out for directory    */
 '| Var rubbish'                      /* Store Result in Variable   */
Parse Var rubbish mode . . . rubbish   /* Extract the Filemode       */
If rubbish = directory Then Do         /* Directory is accessed      */
  accessed = '1'                       /* Set the correct Flag       */
End                                    /* End If Then Do             */
```

```
Else accessed = 'Ø'                    /* Directory not accessed     */
Return accessed
                                       /*                            */
/*====================================================================*
 * Access Directory                          S U B R O U T I N E *
 *====================================================================*/
Access_Directory:
Parse Value Freemode() With mode       /* Get a free Filemode        */
'ACCESS' directory mode                /* Access the Directory       */
accessed = '2'                         /* Set accessed Flag          */
Return
                                       /*                            */
/*====================================================================*
 * Release the Directory if neccessary       S U B R O U T I N E *
 *====================================================================*/
Release_Directory:
If accessed = '2' Then 'RELEASE' mode  /* If accessed with this EXEC */
Return
                                       /*                            */
/*====================================================================*
 * Do the real work now                      S U B R O U T I N E *
 *====================================================================*/
Compute_Directory_Size:
'PIPE (ENDCHAR ?)',                    /* Start 1st Pipeline         */
  '  CMS LISTFILE * *' mode '(ALL',    /* List all Files             */
  '| Drop First',                      /* Drop the Header            */
  '| a: Fanout',                       /* Connect 2nd Output Stream  */
  '| Specs 35-44 1',                   /* Take only those columns    */
  '| Stem recs.',                      /* Store Result into Stem     */
  '?',                                 /* Start 2nd Pipeline         */
    'a:',                              /* Connect to 2nd Output Str. */
    '| Specs 45-55 1',                 /* Take only those columns    */
    '| Stem blocks.'                   /* Store Result into Stem     */
recs = 'Ø'                             /* No Records so far          */
blocks = 'Ø'                           /* No Blocks so far           */
Do a = '1' To recs.Ø                   /* Compute number of Records  */
  recs = recs.a + recs                 /* Add all together           */
  blocks = blocks.a + blocks           /* Add all together           */
End                                    /* End Do to                  */
files = recs.Ø                         /* Number of Files found      */
bytes = blocks * '4Ø96'                /* Multiply with Blocksize    */
kilobytes = bytes / '1Ø24'             /* Divide into Kilobytes      */
megabytes = kilobytes / '1Ø24'         /* Divide into Megabytes      */
Call Justify_Variables
Return
                                       /*                            */
/*====================================================================*
 * Display the Result                        S U B R O U T I N E *
 *====================================================================*/
Display_Result:
```

```
header = 'Files Blocks Records Megabytes' /* Define Message Header   */
result = files blocks recs megabytes   /* Define the Result          */
Address XEDIT 'MSG' pgn Time() header
Address XEDIT 'MSG' pgn Time() result
If rc ¬= 'Ø' Then Do                   /* XEDIT not invoked          */
  Say pgn Time() header
  Say pgn Time() result
End                                    /* End If Then Do             */
Return
                                       /*                            */
/*====================================================================*
 * Find out Size of all Subdirectories          S U B R O U T I N E *
 *====================================================================*/
Compute_All_Directories:
'PIPE (ENDCHAR ?)',                    /* Start 1st Pipeline         */
  'CMS LISTDIR' directory ,            /* List all Subdirectories    */
  '| Drop First',                     /* Drop the Message Header    */
  '| Specs 4-* 1',                    /* Extract the Directory only */
  '| Stem subdirs.'                   /* Store Result into Stem     */
rec = 'Ø'                              /* Init Variable              */
blk = 'Ø'                              /* Init Variable              */
meg = 'Ø'                              /* Init Variable              */
fil = 'Ø'                              /* Init Variable              */
Do b = '1' To subdirs.Ø                /* Process every found SubDir */
  directory = subdirs.b                /* This is the Dir. Name      */
  Call Check_Directory
  fil = fil + files                    /* Add all together           */
  rec = rec + recs                     /* Add all together           */
  blk = blk + blocks                   /* Add all together           */
  meg = meg + megabytes                /* Add all together           */
  Call Release_Directory
End                                    /* End Do To                  */
recs = rec                             /*                            */
files = fil                            /*                            */
blocks = blk                           /*                            */
megabytes = meg                        /*                            */
Call Justify_Variables
Call Display_Result
Call Exit
Return
                                       /*                            */
/*====================================================================*
 * ....                                         S U B R O U T I N E *
 *====================================================================*/
Justify_Variables:
recs = Right(recs,7)                   /* Total Number of Records    */
files = Right(files,5)                 /* Number of Files found      */
blocks = Right(blocks,6)               /* Total Number of Blocks     */
megabytes = Left(megabytes,9)          /* Number of Megabytes        */
Return
```

```
                                        /*                              */
/*======================================================================*
 * Get a free Filemode                                  F U N C T I O N *
 *======================================================================*/
Freemode:
Call 'CSL' 'DMSGETFM rc reasc fm'       /* Get a free Filemode          */
Return fm
                                        /*                              */
```

Andreas Brunner
Atraxis AG (Switzerland)

A full screen console interface – part 5

Editor's note: this month we continue the code for the full screen console interface for Disconnected Service Machines (DSM). This 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.*

```
RELEASE  EQU   *
         CMSSTOR RELEASE,ADDR=(1),DWORDS=(Ø),MSG=YES
         A     RØ,FSRELDW              Total of double words released
         ST    RØ,FSRELDW
         L     RØ,FSREL
         A     RØ,ONE
         ST    RØ,FSREL                Total of storage releases
         BR    R14
         SPACE 3                    *T*
*                                   *T*
* Create trace entries for IUCV     *T* (testing)
*                                   *T*
*                                   *T*
TRACE    EQU   *                    *T* Create trace entries (testing)
         L     R1,TRACEPTR          *T*
         ST    RØ,Ø(,R1)            *T*
         L     RØ,IPPATHID          *T*
         ST    RØ,4(,R1)            *T*
         LA    R1,8(,R1)            *T*
         C     R1,TRACEEND          *T*
         BL    TRACE1ØØ             *T*
```

```
         L     R1,TRACEBEG         *T*
TRACE100 ST    R1,TRACEPTR         *T*
         BR    R14                 *T*
         SPACE 3
         ENTRY CSCSVPD
         SPACE 3
         CSCDATA PRINT
         SPACE
CSC00980
CSCPARMC DS    0D                      IUCV Parmlist for CP
CSC00990
         ORG   *+IPSIZE*8
CSC01000
CSCPARMU DS    0D                      IUCV Parmlist for Users
CSC01010
         ORG   *+IPSIZE*8
CSC01020
CSCPARMA DS    0D                      IUCV Parmlist for APPC/VM
CSC01010
         ORG   *+IPSIZE*8
CSC01020
         SPACE
CSC01060
CSCSSV13 DS    F                       Save R13   CSCSVP
IOPRSV14 DS    F                       Save R14   IOPROC
CPPRSV14 DS    F                                  CPPROC
IDPRSV14 DS    F                                  IDPROC
MGPRSV14 DS    F                                  MGPROC
ADDTSV14 DS    F                                  ADDTEB
CLEASV14 DS    F                                  CLEAR
SENDSV14 DS    F                                  SEND
MATCSV14 DS    F                                  MATCH
MATCSAVE DS    2F                      MATCH save area for R4 and R5
         SPACE 3
         CSCDS (UID,CCH,CMD,USR,PFX,MSG),PRINT
         SPACE 3
         PUSH  PRINT
         PRINT OFF
         COPY  IPARML
         POP   PRINT
*        INTBLOK
         REGEQU
         END
```

CSCMSG ASSEMBLE

```
         TITLE 'CSCMSG - CSC Error Messages Processor'
CSCMSG   START X'0161C8'
         PRINT NOGEN
```

```
         CSCHDR                          Display error messages
*
*   Input:  R1 points to a 6 bytes parameter
*
*              From To Len Type Description
*
*               Ø   1   2    H          Number
*               2   4   3    C          Identifier
*               5   5   1    X          Output options
*
*   The message is prefixed by
*
*       hh:mm:ss - Time the message is issued (See SET TIME)
*
*       ppprrrnnnnt
*        |  |  |  |
*        |  |  |  '- Message type, I, W, E, S, C, T
*        |  |  '- Message number
*        |  '- Module identifier
*        '- Application prefix
*
*
*      Application prefix is CSC.
*
*      Module identifier contains the last three characters of the
*         module name
*
*      Message number is a decimal from 1 to 9999. Zero and 10000 are
*         reserved and must be the first and last entries in the table.
*
*      Message type is one alphabetic character. The following types
*         are defined:
*
*            I - Information messages
*            W - Warning
*            E - Error
*            S - Severe
*            C - Critical
*            T - Termination
*
*
*   The message output is controled by:
*
*         1. Flags in the option byte
*
*            MSGOPTUS - Message to be sent to the user
*            MSGOPTNC - Do not redisplay the command entered
*            MSGOPTNA - Do not sound the alarm
*            MSGOPTSP - Do not compress spaces
*
```

```
*            Note: By default the message is sent to the console.
*                  User implies Redisplay and Alarm.
*
*         2. Message level (See SET MSGLVL)
*
*            Ø - Debug general messages
*            1 -          I messages
*            2 -          W messages
*            3 -          E messages
*            4 - Normal   I messages
*            5 -          W messages
*            6 -          E messages
*            7 - Critical messages
*            8 -
*            9 - Mandatory
*
*
*   Called via MSG macro
*
*
*   Register usage
*       R2/R3 - Begin and end of output message
*       R4/R5 - Begin and end of input message
*       R6    - Input parameters
*       R7    - Message scanning
*
*
*   Replaceable parameters
*
*       Key   Field      Description
*
*       @AP   CSCID      Application-id
*       @CD   CSCCOMM    User command name
*       @CO   CSCCOPT    User command option
*       @DT   (diag ØC)  Today's date, format yy/mm/dd
*       @IP   IPRCODE    IUCV IPRCODE
*       @IU   UIDVMID    IUCV target user
*       @RC   CSCRC      Return code, stored by MSG macro
*       @PG   CSCNAME    Program name (IUCV)
*       @Bn   (Rn)       Character (1 byte)
*       @Cn   (Rn)       Character string (8 bytes maximum)
*       @Ln   (Rn)       Character string (15 bytes maximum)
*       @Rn   (Rn)       Decimal value
*       @Xn   (Rn)       Hexadecimal string (4 bytes)
*
*             Note: "n" must be between Ø and 9.
*
*
         SPACE 3
*
```

```
* Display error messages
*
*
         USING UIDSECT,R8
         USING MSGISECT,R4
         SPACE
         STM   R14,R9,ERRSAVE          Save work registers
         LH    RØ,Ø(,R1)               Load message number
         L     R4,MSGTABLE             Address message table
MSG1ØØ   AH    R4,MSGILEN              Skip first entry
         CH    RØ,MSGINUM              Check message number
         BH    MSG1ØØ                  Not this one, try next
         BE    MSG2ØØ                  Found it
         L     R4,MSGTABLE             Not in table, use first one
MSG2ØØ   SR    RØ,RØ                   Clear register
         IC    RØ,MSGILVL              Load message level
         C     RØ,MSGLEVEL             Compare with display level
         BL    MSGRET                  Too low, do not display
         LR    R6,R1                   Address input parameters
         MVC   MSGOPREF,CSCID          Move application prefix
         MVC   MSGOIDEN,2(R6)          Move module identification
         LH    RØ,Ø(,R6)               Load message number
         CVD   RØ,MSGOCONV             Convert to decimal
         OI    MSGOCONV+7,X'ØF'
         UNPK  MSGONUM,MSGOCONV        Move into message code
         MVC   MSGOTYPE,MSGITYPE       Message type
*        MVI   MSGOBLNK,C' '
         XC    MSGOTEXT,MSGOTEXT       Clear message text
         LA    R2,MSGOTEXT             Begin of output message
         LA    R3,MSGOTEXT+L'MSGOTEXT  End of message
         LH    R5,MSGILEN              Message length
         AR    R5,R4                   Message end
         LA    R4,MSGITEXT             Begin of message text
         LR    R7,R4                   Scan message
MSG3ØØ   LA    R1,@CDLIST              Table of processing routines
MSG31Ø   CLI   Ø(R7),C'@'              Look for replaceable symbol
         BE    MSG4ØØ                  Found one, test
         LA    R7,1(,R7)
         CR    R7,R5                   All done?
         BL    MSG31Ø                  No, keep looping
         B     MSG6ØØ                  Done, copy end of message
         SPACE
MSG4ØØ   CLC   Ø(3,R1),Ø(R7)           Check @xx code
         BE    MSG5ØØ                  Found it, process
         LA    R1,8(,R1)               Search all table
         CLI   Ø(R1),X'FF'             End of part one
         BNE   MSG4ØØ
MSG41Ø   LA    R1,8(,R1)               Now for part two
         CLC   Ø(2,R1),Ø(R7)           Check @x code
         BE    MSG5ØØ
```

```
         CLI   Ø(R1),X'FF'
         BNE   MSG41Ø                  If not found, use last entry
MSG5ØØ   L     R15,4(,R1)             Load routine address
         BASR  R14,R15
MSG6ØØ   SR    R7,R4                  Bytes to copy
         BZ    MSG62Ø
         LA    RØ,Ø(R7,R2)            Can we do it?
         CR    RØ,R3
         BNH   MSG61Ø                 No, too much
         LR    R7,R3
         SR    R7,R2                  Truncate message
MSG61Ø   BCTR  R7,Ø                   Adjust to EXecute
         EX    R7,MSGMVC              Move data
         LA    R2,1(R7,R2)            First available byte
         CR    R2,R3
         BNL   MSG7ØØ                 No space...
         LA    R4,1(R7,R4)            Point to '@'
MSG62Ø   LA    R4,3(,R4)              Skip parameter @xx
         CR    R4,R5                  Valid @xx
         BH    MSG66Ø
         MVC   Ø(L'MSGOWORK,R2),MSGOWORK Yes, move replacement
         LR    R7,R2                  Address replacement
MSG63Ø   LA    R2,1(,R2)              Check for first blank/null
         CLI   Ø(R2),C' '
         BH    MSG63Ø                 Not yet, loop
         TM    5(R6),MSGOPTSP         Compress spaces?
         BZ    MSG65Ø                 Yes, keep going
         LA    R7,3(,R7)              No, compensate for @xx
         CR    R7,R2                  Minimum length is three
         BNH   MSG64Ø                 Long enough
         LR    R2,R7                  Expand replacement to three byte
MSG64Ø   SR    R7,R2                  Complement length of replacement
         SR    R4,R7                  Adjust input pointer
MSG65Ø   LR    R7,R4                  New scan byte
         CR    R4,R5                  All done?
         BL    MSG3ØØ
         CR    R2,R3                  Space for end '.'?
         BNL   MSG7ØØ
MSG66Ø   MVI   Ø(R2),C'.'              Yes...
         LA    R2,1(,R2)
MSG7ØØ   TM    5(R6),MSGOPTUS         Message to be sent to the user?
         BO    MSG8ØØ
         B     MSG71Ø                 ******************************
         LA    RØ,MSGOLENN+1          Calculate message length
         SR    R2,RØ                  Message length (prefix excluded)
         STC   R2,MSGOLENN            Store message length into prefix
         LA    R2,MSGOLENN
         B     MSG72Ø
         SPACE
MSG71Ø   LA    R1,DIAGØØØC            Work area for DIAG
```

```
         DIAG  R1,RØ,X'ØØØC'           Get date and time
         MVC   MSGOTIME,DIAGØØØC+8     Move time
         MVI   MSGOLENN,C' '           Clear message length (no time)
         LA    RØ,MSGOLENT+1           Calculate message length (time)
         SR    R2,RØ                   Message length (prefix excluded)
         STC   R2,MSGOLENT
         LA    R2,MSGOLENT
         SPACE
MSG72Ø   TM    CSCFLGØ1,HNDIOS         Check for Console trap
         BZ    MSG73Ø
         HNDIO CLR,DEVNAME=CONS        Disable trap
MSG73Ø   APPLMSG TEXTA=(R2),COMP=NO,DISP=ERRMSG
         TM    CSCFLGØ1,HNDIOS
         BZ    MSG74Ø
         WAITT                         Wait for I/O to complete
         L     R2,ADDRCONS
         L     R3,@CSCIOX
         LA    R4,IOXBK
         HNDIO SET,DEVNAME=CONS,DEVICE=(R2),EXIT=(R3),                 *
               INTBLOK=((R4),L'IOXBK)
MSG74Ø   TM    CSCFLGØ1,MSGPRINT       Copy message to Printer?
         BZ    MSGRET                  No, done
         SR    R2,R2                   Required by next IC
         IC    R2,MSGOLENT             Load message length
         LA    R3,MSGOTIME             Address message text
         PRINTL (R3),(R2)              Print message
         B     MSGRET
         SPACE
MSG8ØØ   LA    RØ,MSGOPREF             Calculate message length
         SR    R2,RØ                   Message length
         ST    RØ,SCRMSG               Store message address
         ST    R2,SCRMSGL              Store message length
         OI    UIDOPT4,UIDBMSG         Set MSG option
         TM    5(R6),MSGOPTNC          Redisplay input command?
         BO    MSG81Ø                  No, check alarm
         SR    RØ,RØ                   Store zero to address and length
         ST    RØ,SCRMCL               CSCUSR will display it from...
         ST    RØ,SCRMCLL              ... the Retrieve Buffer
         OI    UIDOPT4,UIDBMCL         Set MCL option
MSG81Ø   TM    5(R6),MSGOPTNA          Sound the alarm?
         BO    MSGRET                  No, done...
         OI    UIDOPT4,UIDBALM         Set ALM option
MSGRET   LM    R14,R9,ERRSAVE
         BACK
         SPACE 3
@BN      EQU   *                       Rn (Character - 1 byte)
         CLI   2(R7),C'Ø'              "n" must be between Ø and 9
         BL    COPY
         CLI   2(R7),C'9'
         BH    COPY
```

```
         SR    R1,R1                  Clear register
         IC    R1,2(,R7)              Insert "n"
         N     R1,NOZONE              Remove zone
         SLL   R1,2                   Convert to fullwords
         L     R1,ERRSAVE+8(R1)       Load Rn
         MVC   MSGOWORK(1),Ø(R1)      Move single byte
         MVI   MSGOWORK+1,C' '        Terminate after single byte
         BR    R14
         SPACE 3
@CN      EQU   *                      Rn (Character string - 8 bytes)
         CLI   2(R7),C'Ø'             "n" must be between Ø and 9
         BL    COPY
         CLI   2(R7),C'9'
         BH    COPY
         SR    R1,R1                  Clear register
         IC    R1,2(,R7)              Insert "n"
         N     R1,NOZONE              Remove zone
         SLL   R1,2                   Convert to fullwords
         L     R1,ERRSAVE+8(R1)       Load Rn
         MVC   MSGOWORK(8),Ø(R1)      Move string
         MVI   MSGOWORK+8,C' '        Terminate after 8 bytes maximum
         BR    R14
         SPACE
@LN      EQU   *                      Rn (Character string - 15 bytes)
         CLI   2(R7),C'Ø'             "n" must be between Ø and 9
         BL    COPY
         CLI   2(R7),C'9'
         BH    COPY
         SR    R1,R1                  Clear register
         IC    R1,2(,R7)              Insert "n"
         N     R1,NOZONE              Remove zone
         SLL   R1,2                   Convert to fullwords
         L     R1,ERRSAVE+8(R1)       Load Rn
         B     MSGCHAR                Move and return to caller
         SPACE
@RN      EQU   *                      Rn (Decimal value)
         CLI   2(R7),C'Ø'             "n" must be between Ø and 9
         BL    COPY
         CLI   2(R7),C'9'
         BH    COPY
         SR    R1,R1                  Clear register
         IC    R1,2(,R7)              Insert "n"
         N     R1,NOZONE              Remove zone
         SLL   R1,2                   Convert to fullwords
         L     R1,ERRSAVE+8(R1)       Load Rn
         B     MSGDEC                 Convert and return to caller
         SPACE
@XN      EQU   *                      Rn (Hexadecimal value - 4 bytes)
         CLI   2(R7),C'Ø'             "n" must be between Ø and 9
         BL    COPY
```

```
         CLI   2(R7),C'9'
         BH    COPY
         SR    R1,R1                    Clear register
         IC    R1,2(,R7)                Insert "n"
         N     R1,NOZONE                Remove zone
         SLL   R1,2                     Convert to fullwords
         L     R1,ERRSAVE+8(R1)         Load Rn
         B     MSGHEXA4                 Convert and return to caller
         SPACE 3
COPY     MVC   MSGOWORK(3),Ø(R7)        Unknown, copy
         MVI   MSGOWORK+3,C' '          Terminate string
         BR    R14
         SPACE
@AP      MVC   MSGOWORK(L'CSCID),CSCID Application-id
         BR    R14
         SPACE
@CD      MVC   MSGOWORK(L'CSCCOMM),CSCCOMM Move command name
         MVI   MSGOWORK+L'CSCCOMM,C' '
         BR    R14
         SPACE
@CO      MVC   MSGOWORK(L'CSCCOPT),CSCCOPT Move command option
         MVI   MSGOWORK+L'CSCCOPT,C' '
         BR    R14
         SPACE
@DT      LA    R1,DIAGØØØC              Work area for DIAG
         DIAG  R1,RØ,X'ØØØC'            Get date and time
         MVC   MSGOWORK(2),DIAGØØØC+6   Move today's year
         MVI   MSGOWORK+2,C'/'          Give VM a hand
         MVC   MSGOWORK+3(5),DIAGØØØC   Move today's month/day
         MVI   MSGOWORK+8,C' '          Terminate date
         BR    R14
         SPACE
@IP      L     R1,Ø(,R9)                Load IUCV IPRCODE
         B     MSGHEXA                  Convert and return to caller
         SPACE
@IU      MVC   MSGOWORK(L'UIDVMID),UIDVMID Move IUCV target user
         MVI   MSGOWORK+L'UIDVMID,C' '
         BR    R14
         SPACE
@PG      MVC   MSGOWORK,CSCNAME         Move program name
         BR    R14
         SPACE
@RC      L     R1,CSCRC                 Load return code
         B     MSGDEC                   Convert and return to caller
         SPACE 3
MSGCHAR  EQU   *                        Character - 1 to 15 bytes
         MVC   MSGOWORK,Ø(R1)           Move string
         BR    R14
         SPACE
MSGDEC   EQU   *                        Decimal value - fullword
```

```
         LR    RØ,R1                    Save value to convert
         CVD   R1,MSGOWORK              Convert to packed decimal
         OI    MSGOWORK+7,X'ØF'         Remove signal
         UNPK  MSGOCONV(7),MSGOWORK(8) Unpack
         MVI   MSGOCONV+7,C' '          Terminate field
         LA    R1,MSGOCONV-1            Remove leading zeros
MSGDEC1Ø LA    R1,1(,R1)
         CLI   Ø(R1),C'Ø'
         BE    MSGDEC1Ø
         LTR   RØ,RØ                    Check for negative number
         BNM   MSGDEC2Ø                 No, it is positive or zero
         BCTR  R1,Ø                     Back-up one byte
         MVI   Ø(R1),C'-'               Copy signal
MSGDEC2Ø MVC   MSGOWORK,Ø(R1)           Move value into work space
         CLI   MSGOWORK,C' '            Is it blank?
         BNER  R14
         MVI   MSGOWORK,C'Ø'            Yes, make it zero
         BR    R14
         SPACE
MSGHEXA  EQU   *                        Hexadecimal - 1 byte
         STC   R1,MSGOWORK+4            Store byte to decode
         UNPK  MSGOWORK(3),MSGOWORK+4(2) Unpack
         NC    MSGOWORK(2),NOZONE
         TR    MSGOWORK(2),HEXTABLE     Convert to hexadecimal
         MVI   MSGOWORK+2,C' '          This is a 2 bytes field
         BR    R14
         SPACE
MSGHEXA4 EQU   *                        Hexadecimal - 4 bytes
         UNPK  MSGOWORK(9),Ø(5,R1)      Unpack
         NC    MSGOWORK(8),NOZONE
         TR    MSGOWORK(8),HEXTABLE     Convert to hexadecimal
         MVI   MSGOWORK+8,C' '          This is an 8 bytes field
         BR    R14
         SPACE
MSGMVC   MVC   Ø(Ø,R2),Ø(R4)
         SPACE 3
@CDLIST  DS    ØD
         DC    C'@AP ',A(@AP)           Application-id
         DC    C'@CD ',A(@CD)           Command name
         DC    C'@CO ',A(@CO)           Command option name
         DC    C'@DT ',A(@DT)           Today's date
         DC    C'@IP ',A(@IP)           IUCV IPRCODE
         DC    C'@IU ',A(@IU)           IUCV target user
         DC    C'@PG ',A(@PG)           Program name
         DC    C'@RC ',A(@RC)           Return code
         DC    X'FFFF',A(Ø)             End of part one
         DC    C'@Bn ',A(@BN)           Rn Character (1 byte)
         DC    C'@Cn ',A(@CN)           Rn Character string (8 bytes)
         DC    C'@Ln ',A(@LN)           Rn Character string (15 bytes)
         DC    C'@Rn ',A(@RN)           Rn Decimal value
```

```
         DC    C'@Xn ',A(@XN)          Rn Hexadecimal value
         DC    X'FFFF',A(COPY)         Not found, copy it
         SPACE
MSGOLENT DS    CL1                     Message length (time option)
MSGOTIME DS    CL8                     Time prefix
MSGOLENN DS    CL1                     Message length (no time option)
MSGOPREF DS    CL3                     Message code
MSGOIDEN DS    CL3
MSGONUM  DS    CL4
MSGOTYPE DS    CL1
MSGOBLNK DS    CL1                     Message separator
MSGOTEXT DS    CL66                    Message text
         SPACE
MSGOCONV DS    D                   *** Message work areas
MSGOWORK DS    CL16                *** Must follow MSGOTEXT
         SPACE
MSGTABLE DC    V(CSCMSL)               Message table
ERRSAVE  DS    12F                     Save are for R14-R9
NOZONE   DC    X'0F0F0F0F0F0F0F0F'     Mask to remove zones
HEXTABLE DC    C'0123456789ABCDEF'     Translate table for HEXA data
         SPACE 3
         CSCDATA
         CSCDS UID
         SPACE
MSGISECT DSECT                         MSGWRITE Input message
MSGILEN  DS    H                       Table entry length
MSGINUM  DS    H                       Message number
MSGILVL  DS    X                       Message level
MSGITYPE DS    X                       Message type
MSGITEXT DS    C                       Message text (variable length)
         SPACE
         REGEQU
         END
```

CSCMSL ASSEMBLE

```
         TITLE 'CSCMSL - CSC Message file'
         MACRO
         MSG   &P1,&P2,&P3,&P4
         LCLA  &I,&LENGTH
         GBLA  &NUMBER
&I       SETA  1
&LENGTH  SETA  K'&P4
.LOOP    ANOP
&I       SETA  &I+1
         AIF   (&I EQ K'&P4).OUT
         AIF   ('&P4'(&I,1) NE '''').LOOP
&LENGTH  SETA  &LENGTH-1
&I       SETA  &I+1
```

```
         AGO   .LOOP
.OUT     ANOP
         AIF   (&NUMBER LT &P1).SKIP
         AIF   (&NUMBER EQ Ø).SKIP
         MNOTE 4,'Message out of sequence. Ignored.'
         MEXIT
.SKIP    AIF   (&LENGTH LE 66).GEN
         MNOTE 4,'Message too long. Truncation may occur.'
.GEN     ANOP
&NUMBER  SETA  &P1
&LENGTH  SETA  &LENGTH+4
         DC    AL2(&LENGTH)           Message length (Includes prefix)
         DC    AL2(&P1)               Message number
         DC    AL1(&P2)               Message level
         DC    CL1'&P3'               Message type
         DC    C&P4                   Message text
         MEND
         EJECT
CSCMSL   START X'Ø1BCØØ'
CSCMSL   RMODE ANY
         PRINT NOGEN
*
*   Message definition
*
*
*      Message number is a decimal from 1 to 9999. Zero and 1ØØØØ are
*         reserved and must be the first and last entries in the table.
*
*
*      Message type is one alphabetic character. The following types
*         are defined:
*
*            I - Information messages
*            W - Warning
*            E - Error
*            S - Severe
*            C - Critical
*            T - Termination
*
*
*      Message level.
*
*            Ø - Debug general messages
*            1 -          I messages
*            2 -          W messages
*            3 -          E messages
*            4 - Normal   I messages (default)
*            5 -          W messages
*            6 -          E messages
*            7 - Critical messages
```

```
*            8 -
*            9 - Mandatory
*
*
*   Message expansion
*
*         From  To  Length  Type    Contents
*          ØØ   Ø1    Ø2   binary   Entry length. Message plus prefix
*          Ø2   Ø3    Ø2   binary   Message number
*          Ø4   Ø4    Ø1   binary   Message level
*          Ø5   Ø5    Ø1    char    Message type
*          Ø6   ..   var    char    Message text. Maximum is 64 bytes
*
*
*   Used by CSCMSG. No executable code.
*
*
         EJECT
*
*   Message groups
*
*      ØØØØ      - Begin of table. Undefined messages
*
*
*      ØØ1-ØØ39 - CSCSVP
*
*      Ø040-Ø139 - CSCCFG   Configuration
*
*      Ø150-Ø159 - CSCSEV   Sever IUCV connection
*
*      Ø160-Ø169 - CSCCPW   Write Data File
*
*      Ø170-Ø179 - CSCRDF   Read Data File
*
*      Ø180-Ø189 - CSCRLS   Release allocated storage
*
*      Ø190-Ø199 - CSCCLS   Terminate IUCV and Console I/O processing
*
*      Ø200-Ø299 - CSCUSR   Reserved for CSCUSR usage
*
*      Ø300-Ø339 - CSCUSC   User commands
*
*      Ø340-Ø349 - CSCUPR   Print/Write commands
*
*      Ø350-Ø359 - CSCULC   Locate/Match/Go commands
*
*      Ø360-Ø369 - CSCURL   Release command
*
*      Ø370-Ø379 - CSCUOP   OP command
*
```

```
*      0380-0389 - CSCUEX   Exclude/Include commands
*
*      0600-0639 - CSCOPC   Operator commands
*
*      0640-0649 - CSCOPQ   Query command
*
*      0800-0809 - CSCTMR   Time-Based Events
*
*  *   0400-0419 - CSCRNL   Remote Node Links
*
*  *   0870-0879 - CSCUSA   APPC/VM related commands
*
*  *** 0900-0919 - CSCOPA   APPC/VM Start/Stop commands
*
*
*      10000     - End of table
*
*
         EJECT
*
*    Begin of table
*
*
MSGLIST  DS    0D           Error messages
*
*
 MSG 0000,9,E,'Message not defined'
*
*    CSCSVP
*
*
 MSG 0001,4,I,'Initialization of @AP started'
 MSG 0002,9,E,'Error executing HNDIO SET. Return code is @RC'
 MSG 0003,9,E,'Program @PG already active'
 MSG 0004,9,E,'Error executing HNDIUCV SET. Return code is @RC'
 MSG 0005,9,E,'Error executing CMSIUCV CONNECT. Return code is @RC'
 MSG 0006,4,I,'Initialization of @AP ended on @DT'
 MSG 0007,4,I,'@AP is terminating'
 MSG 0008,4,I,'Termination of @AP ended on @DT'
 MSG 0011,9,E,'Unable to connect to *MSG CP System Service'
 MSG 0012,9,S,'Unexpected message type received from *MSG System Servic*
               e'
 MSG 0013,9,E,'IUCV Receive buffer too small'
 MSG 0014,9,E,'IUCV RECEIVE error. IPRCODE is @IP'
 MSG 0015,6,E,'Unable to establish IUCV session with @IU'
 MSG 0016,4,I,'IUCV Session with @IU started'
 MSG 0017,6,E,'IUCV Session with @IU not authorized. Rejected'
 MSG 0018,6,E,'IUCV SEVER error. IPRCODE is @IP'
 MSG 0019,9,S,'Call the Ghost Busters..'
 MSG 0020,9,S,'Call the Ghost Busters again..'
```

```
 MSG 0024,6,E,'IUCV SEND error. IPRCODE is @IP'
*
*    CSCCFG   Configuration
*
*
 MSG 0040,9,E,'Unable to Open Data File. Return code from FSOPEN is @RC*
               '
 MSG 0041,4,I,'Creating new Data File'
 MSG 0042,4,I,'Data File restarted'
 MSG 0043,4,I,'Configuration completed successfully'
 MSG 0044,6,E,'Invalid Configuration record: "@C6". Ignored'
 MSG 0045,5,W,'Configuration File not found'
 MSG 0046,6,E,'Error reading Configuration File. Return code from FSREA*
               D is @RC'
 MSG 0047,6,E,'Invalid LOCAL/REMOTE combination found'
 MSG 0048,6,E,'Remote links will not be activated'
 MSG 0050,6,E,'Missing @CD operand(s). Statement discarded'
 MSG 0051,6,E,'Invalid @CD operand: @L6. Statement discarded'
 MSG 0052,6,E,'Unexpected @CD operand: @L6. Statement discarded'
 MSG 0053,6,E,'@CD operand "@C6..." is too long. Statement discarded'
 MSG 0054,6,E,'Invalid @CD operand: @L6. Ignored'
 MSG 0060,5,W,'Value @R4 for DFRECS is too small. Default of @R2 used'
 MSG 0061,5,W,'DFRECS value adjusted from @R2 to @R4'
 MSG 0070,6,E,'Missing @CO value for EVENT. Statement discarded'
 MSG 0071,6,E,'Invalid @CO value "@L6" for EVENT. Statement discarded'
 MSG 0072,5,W,'Duplicate @CO option for EVENT'
 MSG 0080,6,E,'Only one LOCAL statement allowed. Statement discarded'
 MSG 0081,6,E,'Node name @C2 is not unique. Statement discarded'
 MSG 0082,6,E,'Resource name @C2 is not unique. Statement discarded'
 MSG 0090,6,E,'Missing USER option for MSG. Statement discarded'
 MSG 0091,6,E,'Missing LOCATE option for MSG. Statement discarded'
 MSG 0092,6,E,'Missing @CD value for MSG. Statement discarded'
 MSG 0093,6,E,'@CD value @C6... for MSG too long. Statement discarded'
 MSG 0094,6,E,'@CD mask @C6... for MSG too long. Statement discarded'
 MSG 0100,6,E,'@C6 is an invalid Prefix. Must be one character long'
 MSG 0101,6,E,'Missing user-id for Prefix "@C6". Statement discarded'
 MSG 0102,6,E,'Missing class value in PREFIX statement. Discarded'
 MSG 0103,6,E,'@CD class @R2 not in the range 25-32. Statement discarde*
               d'
 MSG 0110,6,E,'Node without user-id found on ROUTE statement.Discarded'
 MSG 0120,5,W,'Title @C6... too long. Truncated'
 MSG 0130,6,E,'Non numeric @CD class: @L6. Ignored'
 MSG 0131,6,E,'@CD class @R2 not in the range 01-32. Ignored'
*
*    CSCSEV   Sever IUCV connection
*
*
 MSG 0150,6,E,'IUCV Session with @IU cancelled'
 MSG 0151,6,E,'Active IUCV Session with @IU not found'
 MSG 0152,4,I,'IUCV Session with @IU ended'
```

```
*
*    CSCCPW   Write Data File
*
*
 MSG Ø16Ø,4,I,'Data File expanded'
 MSG Ø161,4,I,'Beginning Data File expansion'
 MSG Ø162,4,I,'Data File truncated'
 MSG Ø163,9,S,'Error writing Data file. Return code from FSWRITE is @RC*
               '
 MSG Ø164,6,E,'Exit Exec @C2 not found'
*
*    CSCRDF   Read Data File
*
*
 MSG Ø17Ø,9,S,'Error reading Data file. Return code from FSREAD is @RC'
*
*    CSCRLS   Release allocated storage
*
*
 MSG Ø18Ø,5,W,'@R2 bytes (@R3 allocations) were not released'
*
*    CSCCLS   Terminate IUCV and Console I/O processing
*
*
 MSG Ø19Ø,9,E,'Unable to terminate connection with CP System Service *M*
               SG'
 MSG Ø191,9,E,'Unable to terminate IUCV session'
 MSG Ø192,9,E,'Unable to restore Console I/O processing'
*
*    CSCUSC   User commands
*
*             Ø2ØØ-Ø299 Reserved for CSCUSR
*
*
 MSG Ø3ØØ,9,E,'Invalid data received from @IU'
 MSG Ø3Ø1,9,E,'Invalid PA/PF key. Please try something different..'
 MSG Ø3Ø2,9,E,'Invalid command. Please keep trying, you will succeed..'
 MSG Ø3Ø9,9,I,'Please be patient. This function is not available yet..'
 MSG Ø31Ø,9,E,'Missing operand(s)'
 MSG Ø311,9,E,'Invalid @CD operand: @L6'
 MSG Ø312,9,E,'Unexpected @CD operand: @L6'
 MSG Ø32Ø,9,W,'SHIFT has no effect while Message WRAP is active'
 MSG Ø321,9,E,'SHIFT value @R5 is too big'
 MSG Ø33Ø,9,E,'@CD is valid only in Refresh mode with CMS scroll ON'
 MSG Ø331,9,E,'@CD only works if CMS scroll is active'
*
*    CSCUPR   User Print/Write commands
*
*
 MSG Ø34Ø,9,W,'Data file is empty'
```

```
 MSG Ø341,9,E,'Command interrupted. Reason code is @RØ'
*
*    CSCULC   User Locate/Match/Go commands
*
*
 MSG Ø35Ø,9,E,'String too long. Must be from 1 to 36 characters'
 MSG Ø351,9,I,'String not found'
 MSG Ø352,9,E,'Value @C6 is too long for Date or Time values'
 MSG Ø353,9,E,'Invalid Date entered'
 MSG Ø354,9,E,'Invalid Time entered'
 MSG Ø355,9,E,'You cannot locate a future Date/Time'
 MSG Ø356,9,I,'No record found after specified Date/Time'
*
*    CSCURL   User Release command
*
*
 MSG Ø36Ø,9,W,'You are not authorized to release @C2 messages'
 MSG Ø361,9,W,'Message number @R2 not found'
 MSG Ø362,9,E,'Invalid message number: @R2. Must be between @R4 and @R3*
               '
*
*    CSCUOP   User OP command
*
*
 MSG Ø37Ø,9,E,'Missing value for userid'
 MSG Ø371,9,E,'Value @C6... is too long. Must be from 1 to 8 characters*
               '
 MSG Ø372,9,E,'Userid @C6 is not defined'
 MSG Ø373,9,E,'@C6 is not a valid prefix. Must be one character long'
 MSG Ø374,9,E,'Prefix @C6 is not defined'
 MSG Ø375,9,E,'You are not authorized to operate the @C6 machine'
*
*    CSCUEX   User Exclude/Include commands
*
*
 MSG Ø38Ø,9,E,'@CD did not process prefix "@B6" successfully'
 MSG Ø381,9,W,'Prefix "@B6" not defined. Ignored'

*
*    CSCRNL   Remote Node Links
*
*
 MSG Ø4ØØ,4,I,'Starting APPC/VM communications'
 MSG Ø4Ø1,6,E,'Error executing HDNIUCV SET. Return code is @R2'
 MSG Ø423,4,I,'Terminating APPC/VM communications'
 MSG Ø424,6,E,'Unable to terminate session with CP System Service *IDEN*
               T'
 MSG Ø44Ø,6,E,'Missing authorization to define Global resource @C3'
 MSG Ø441,6,E,'Resource @C3 not available for connections'
 MSG Ø442,6,E,'Maximum IUCV connections reached'
```

```
 MSG Ø443,6,E,'Communication partner @C2 is at maximum connections'
 MSG Ø448,6,E,'Error executing CMSIUCV CONNECT to resource @C3'
 MSG Ø449,6,E,'Return code is @R3. IPRCODE is @IP'
 MSG Ø5ØØ,6,E,'Local node @C2 terminated'
 MSG Ø5Ø1,6,E,'Local node @C2 not available. Resource name is @C3'
 MSG Ø5Ø2,4,I,'Local node @C2 activated. Resource name is @C3'
 MSG Ø5Ø3,4,I,'Activating link to @C2'
 MSG Ø51Ø,4,I,'Remote link from node @C2 lost'
 MSG Ø511,4,I,'Remote node @C2 not available. Resource name is @C3'
 MSG Ø512,4,I,'Remote link to node @C2 successfully activated'
 MSG Ø515,4,I,'Terminating link to node @C2'
 MSG Ø517,4,I,'Remote link to node @C2 terminated'
 MSG Ø521,9,T,'Call the APPC/VM Ghostbusters'
 MSG Ø522,9,T,'Call the APPC/VM Ghostbusters II'
 MSG Ø523,9,T,'Call the APPC/VM Ghostbusters III'
 MSG Ø524,9,T,'Call the APPC/VM Ghostbusters IV'
 MSG Ø525,9,T,'Call the APPC/VM Ghostbusters V'
 MSG Ø526,9,S,'Unexpected IUCV interrupt. PRMLIST is @X9 @X8'
 MSG Ø53Ø,6,E,'APPC/VM SENDDATA error. IPRCODE is @IP'
 MSG Ø531,6,E,'IUCV ACCEPT error. IPRCODE is @IP'
 MSG Ø532,6,E,'APPC/VM RECEIVE error. IPRCODE is @IP'
 MSG Ø533,6,E,'IUCV SEVER error. Return code from CMSIUCV is @R2'
 MSG Ø535,6,E,'Link to @C2 is not reversible. Resource name is @C3'
 MSG Ø536,6,E,'Link @C2 with resource name @C3 is not defined'
 MSG Ø55Ø,4,I,'Remote session started for @C2 on node @C3'
 MSG Ø551,4,I,'User @C2 connected to node @C3'
 MSG Ø552,4,I,'You are already in session with node @C4'
 MSG Ø553,4,I,'You are not authorized to connect to @C4'
 MSG Ø554,4,I,'Remote session ended for @C2 on node @C3'
 MSG Ø559,9,S,'Invalid data received from @C2 on node @C3'
 MSG Ø56Ø,9,S,'Unable to locate session with @C2 from node @C3'
 MSG Ø562,4,I,'User @C2 disconnected from node @C3'
 MSG Ø569,9,E,'Connect to node @C3 is pending. Request ignored'
 MSG Ø57Ø,9,S,'Unable to locate APPC/VM pathid @R2'
 MSG Ø574,4,I,'Session with user @C2 from node @C3 ended'
 MSG Ø58Ø,4,I,'@C2 session with node @C3 cancelled'
 MSG Ø581,4,I,'Connection with node @C3 lost'
 MSG Ø582,4,I,'Remote session with @C2 on node @C3 cancelled'
*
*    CSCOPC   Operator commands
*
*
 MSG Ø6ØØ,9,E,'"@L6" is not a valid CSC command'
 MSG Ø6Ø1,9,I,'Please enter a CSC command or END to terminate'
 MSG Ø6Ø2,9,I,'CSC command @CD ended with return code @R2'
 MSG Ø6Ø3,9,E,'Missing @CD operand(s)'
 MSG Ø6Ø4,9,E,'Invalid @CD operand: @L6'
 MSG Ø6Ø5,9,E,'Unexpected @CD operand: @L6'
 MSG Ø6Ø6,9,E,'Missing @CD subcommand'
 MSG Ø61Ø,9,E,'CMS command is missing'
```

```
 MSG Ø611,9,I,'CMS command "@L6" ended with return code @R2'
 MSG Ø619,9,I,'CSC @CD command entered..'
*
*    CSCOPQ   Operator Query subcommands
*
*
 MSG Ø64Ø,9,I,'@R2 bytes allocated in @R3 allocations'
 MSG Ø641,9,I,'Balance is @R2 bytes in @R3 allocations'
 MSG Ø642,9,E,'APPC/VM support is not enabled'
 MSG Ø643,9,I,'APPC/VM links defined'
 MSG Ø644,9,I,'   Local    @C2        @C3        @C4        @C5'
 MSG Ø645,9,I,'   Remote   @C2        @C3        @C4'
 MSG Ø646,9,I,'@C2      @C3        @C4'
 MSG Ø647,9,I,'No sessions found'
*
*    CSCTMR   Time Based Events
*
*
 MSG Ø8ØØ,6,E,'Missing NAME option for EVENT. Statement discarded'
 MSG Ø8Ø1,6,E,'Missing COMMAND option for EVENT. Statement discarded'
 MSG Ø8Ø2,6,E,'Invalid DATE "@L2" for EVENT. Statement discarded'
 MSG Ø8Ø3,6,E,'Invalid TIME "@L2" for EVENT. Statement discarded'
*
*    CSCUSA   Remote Node User commands
*
*
 MSG Ø87Ø,6,E,'Destination node "@C6..." too long'
 MSG Ø871,6,E,'Destination node @C2 is not defined'
 MSG Ø872,6,E,'Cannot connect to local node'
 MSG Ø873,6,E,'Link to node @C2 is not active'
 MSG Ø874,6,E,'You are already connected'
*
*    CSCOPA   APPC/VM Start/Stop commands
*
*
 MSG Ø9ØØ,9,E,'APPC/VM not enabled'
 MSG Ø9Ø1,9,E,'Local node @C2 is not active'
 MSG Ø9Ø2,9,E,'Missing operand(s)'
 MSG Ø9Ø3,9,E,'Operand @C6... is too long'
 MSG Ø9Ø4,9,E,'Unexpected operand: @L6'
 MSG Ø9Ø5,9,E,'Node @C6 is not defined'
 MSG Ø9Ø6,9,E,'Command is not valid for local node (@C2)'
 MSG Ø9Ø7,9,W,'Link to node @C2 already active'
 MSG Ø9Ø8,9,W,'Activation pending for link to node @C2'
 MSG Ø9Ø9,9,E,'Link to node @C2 not active'
 MSG Ø91Ø,9,I,'@R2 nodes started'
 MSG Ø911,9,I,'All defined nodes are already active'
 MSG Ø916,9,I,'@R2 nodes stopped'
 MSG Ø917,9,I,'All defined nodes are already inactive'
```

```
 MSG 2222,6,E,'@C2 @L3 @L4 @R5'
 MSG 3333,6,E,'@RØ @X2 @X3 @X4'
*
*    End of table
*
*
 MSG 1ØØØØ,9,T,'End of message table'
         END
```

CSCCFG ASSEMBLE

```
         TITLE 'CSCCFG - CSC Set-up and configuration'
CSCCFG   START X'Ø167C8'
         PRINT NOGEN
         CSCHDR                          Set-up and Configuration
*
* Set-up and Configuration
*
*
         USING CCHSECT,R7                CCH (cache) Block
         LA    RØ,CPQUSER                Address CP QUERY command
         LA    R1,USERID                 Address CP response buffer
         LA    R2,L'CPQUSER              Length of CP command
         O     R2,CPRESPB                Request response in buffer
         LA    R3,USERIDL                Length of buffer
         DIAG  RØ,R2,X'ØØØ8'
         LA    R1,USERID(R3)             Address end of CP response
         BCTR  R1,Ø                      Address last byte newline X'15'
         LA    RØ,NODEID                 Address Node name
         SR    R1,RØ                     Length of Node name
         MVC   CSCNODE,BLANKS            Clear field first
         BCTR  R1,Ø                      Prepare to EXecute
         EX    R1,CFGMVC                 Save local VM Node name
         MVC   CSCMSGC,CFGMSGC           Default CP command to send msgs
         MVC   CSCRSCS,CFGRSCS           Default RSCS Service Machine Id
         MVC   CSCLOCAL,BLANKS           Clear APPC/VM Node name
         SR    RØ,RØ                     Initialize totals
         ST    RØ,DFCURR                 Last written record
         ST    RØ,DFSSSLIN               Messages processed
         ST    RØ,DFEXPLIN               Messages expanded (DF file)
         ST    RØ,SCRTTLL                No default Title line
         ST    RØ,USRPTR                 USR User Table
         ST    RØ,MSGPTR                 MSG Message Table
         ST    RØ,RTEPTR                 RTE Route Table
         ST    RØ,RNDPTR                 RND Remote Nodes Table
         ST    RØ,TMRPTR                 TMR Time Based Events Table
         ST    RØ,HLDPTR                 HLD Hold Table
         ST    RØ,HLDLAST                Hold Table last entry
```

```
         ST    RØ,FSALL                Free storage allocated
         ST    RØ,FSALLDW
         ST    RØ,FSREL                Free storage released
         ST    RØ,FSRELDW
         STC   RØ,CFGOPTS              Clear Option byte
         L     RØ,DFRBDFLT             Default number of read buffers
         ST    RØ,DFRBUFFS             Start with it
         L     RØ,DFSZDFLT             Default DF File size
         ST    RØ,DFNEWTOT             Assume expected size
         LA    RØ,MSGDFLVL             Get default message level
         ST    RØ,MSGLEVEL             Store it
         LA    RØ,PFXSIZE              Initialize Prefix Table
         LINK  OBTAIN                  Allocate storage
         ST    R1,PFXPTR               Store entry address
         MVC   Ø(PFXSIZEB,R1),PFXRST   Build first entry (RST)
         BAS   R14,CONFIG              Process Configuration file
         SPACE
         L     RØ,TRACESZ          *T* Trace Table size (double words)
         LINK  OBTAINP             *T* Allocate storage (page aligned)
         ST    R1,TRACEPTR         *T*
         ST    R1,TRACEBEG         *T*
         SLL   RØ,3                *T*
         AR    R1,RØ               *T*
         ST    R1,TRACEEND         *T*
         SPACE
         L     RØ,CACHESZ
         LINK  OBTAINP                 Allocate storage (page aligned)
         ST    R1,CACHE                Save cache address
         LR    R2,RØ                   Size in double words
         SRL   R2,5                    Number of entries (256 bytes)
CFG1ØØ   LR    R7,R1                   Address entry
         LA    R1,CCHSIZEB(,R1)        R1 points to next entry
         ST    R1,CCHFWD               Store forward pointer
         ST    RØ,CCHBWD               Store backward pointer
         LR    RØ,R7                   RØ points to previous entry
         BCT   R2,CFG1ØØ               Loop back
         L     R1,CACHE                Address first entry
         ST    R1,CCHFWD               Forward pointer of last entry
         LR    R7,R1
         ST    RØ,CCHBWD               Backward pointer of first entry
         MVC   CCHUSER,BLANKS          Initialize first entry
         MVC   CCHDATA(L'TOF),TOF
         MVI   CCHCNUM,X'Ø1'
         LA    RØ,L'TOF
         STC   RØ,CCHRLEN
         LINK  PREFIX                  Get Prefix and Attribute fields
         ST    R7,CACHEPTR             Save pointer to current entry
         SPACE
         L     R2,DFRBUFFS
         LA    R3,RDFPTR
```

```
         USING RDFSECT,R3              Address RDF Block
CFG200   LA    R0,RDFSIZE
         LINK  OBTAIN                  Allocate storage
         ST    R1,RDFFWD               Store chain pointers
         ST    R3,RDFBWD-RDFSECT(,R1)
         LR    R3,R1
         LA    R0,512                  Allocate 4K buffer(512 dwords)
         LINK  OBTAINP                 Allocate storage (page aligned)
         ST    R1,RDFADDR              Store buffer address
         SR    R0,R0
         ST    R0,RDFREC               Invalidate buffer
         BCT   R2,CFG200               Do all buffers
         L     R1,RDFPTR               Chain first and last buffers
         ST    R1,RDFFWD
         ST    R3,RDFBWD-RDFSECT(,R1)
         DROP  R3
         SPACE
         USING FSCBD,R1
         FSOPEN FSCB=DFFILEW,FORM=E,CACHE=NO,OPENTYP=WRITE
         LTR   R15,R15
         BZ    CFG300
         MSG   0040,RC                 Unable to Open DF file
         OI    CSCFLG01,CFGERROR       Remember to close the shop
         B     CFG800                  That's all, no need to continue
         SPACE
CFG300   L     R1,FSCBFST              Is it a new file?
         LTR   R1,R1
         BNE   CFG400
         ST    R1,DFOLDTOT             Yes, zero records on file
         MSG   0041                    Creating new DF file
         B     CFG800
         DROP  R1
         SPACE
         USING FSTD,R1
         SPACE
CFG400   L     R0,FSTAIC               The file already exists
         ST    R0,DFOLDTOT             Store number of records
         GO    CSCRDFRS                Restart DF File
         ST    R4,DFCURR               Current is last record written
         MVC   DFBUFF,0(R1)            Copy to DF buffer for CSCCPW
         IC    R0,DFCNUM               Discard current cache record
         L     R7,CACHEPTR             ... this is the TOF record
         L     R7,CCHBWD
         ST    R7,CACHEPTR
         STC   R0,CCHCNUM
         MVC   CSCBUFF(8),BLANKS       Build RST message
         MVC   CSCBUFF+8(L'RST),RST
         LA    R0,CSCBUFF+8+L'RST
         ST    R0,CSCBUFFE
         SR    R9,R9                   Make it a valid address
```

```
         GO    CSCCPW                  Write to DF and override cache
         MSG   0042                    Restarted
         L     R8,CACHESZ              Cache size in bytes
         SRL   R8,5                    Number of entries (32 dwords)
         BCTR  R8,0                    Keep RST record
         L     R9,CACHEPTR             Fill cache from DF records
CFG600   GO    CSCRDFDP                Read previous record from disk
         BZ    CFG700                  Record found, move to cache
         L     R7,CCHBWD-CCHSECT(,R9)  Create new TOF record
         MVC   CCHUSER,BLANKS
         MVC   CCHDATA(L'TOF),TOF
         MVI   CCHCNUM,X'01'
         LA    R0,L'TOF
         STC   R0,CCHRLEN
         LINK  PREFIX                  Get Prefix and Attribute fields
         B     CFG800
         SPACE
CFG700   L     R9,CCHBWD-CCHSECT(,R9)  Address previous cache record
         L     R1,CCHFWD-CCHSECT(,R9)  Save forward and...
         L     R0,CCHBWD-CCHSECT(,R9)  ... backward pointers
         MVC   0(CCHSIZEB,R9),CCHSECT  Move data
         ST    R1,CCHFWD-CCHSECT(,R9)  Restore forward and...
         ST    R0,CCHBWD-CCHSECT(,R9)  ... backward pointers
         BCT   R8,CFG600
         SPACE
CFG800   TM    CSCFLG01,CFGERROR       Any configuration problems
         BO    CFG900                  Yes, done
         L     R0,FSALLDW              Storage allocated during init
         S     R0,FSRELDW              Subtract storage release
         ST    R0,FSINIDW              How much intialization used
         L     R0,FSALL                Same for number of allocations
         S     R0,FSREL
         ST    R0,FSINI
         SR    R2,R2
         ST    R2,CSCRC                Clear return code
         ST    R2,UIDPTR               Clear UID pointer
         ST    R2,SSSPTR               Clear SSS pointer
         ST    R2,CPMSGQ               Zero message counters
         ST    R2,CSCECB               Initialize ECB
         MSG   0043                    Configuration OK
CFG900   BACK
         SPACE
CFGMVC   MVC   CSCNODE(*-*),NODEID     Save local VM Node name
         SPACE 3
*
* Process Configuration file
*
*
CONFIG   EQU   *
         ST    R14,CONFSV14
```

```
         LA    RØ,CSCBUFF
         FSOPEN FSCB=CFGFILE,FORM=E,BUFFER=(RØ) Open file
         LTR   R15,R15
         BNZ   CONF5ØØ                 Not there, keep going
CONFREAD FSREAD FSCB=CFGFILE,FORM=E,BSIZE=256
         LTR   R15,R15
         BNZ   CONF6ØØ                 Read error, check for EOF
         LA    R6,CSCBUFF              Address I/O buffer for SCAN
         AR    RØ,R6
         ST    RØ,CSCBUFFE             End of data address
         LR    R1,RØ
         MVI   Ø(R1),C' '              Make CSCMSG happy, (MSGCHAR)
         SR    RØ,RØ
         ST    RØ,SCANLEN              Start new scan
         LA    RØ,CFGTABLE             Address commands table
         GO    CSCSCN                  Scan record
         BNZ   CONFREAD                Blank record, ignore it
         USING CMDSECT,R2
         MVC   CSCCOMM,CMDNAME         Save command name
         DROP  R2
         LTR   R15,R15
         BZ    CONF4ØØ                 Not valid, check for comments
         LA    R9,2Ø(,R15)             Address processing routine
         BASR  R14,R9                  Execute valid command
         B     CONFREAD                Read next record
         SPACE
CONF4ØØ  CLI   SCANUPP,C'*'            Not valid, could be a comment
         BE    CONFREAD                Ignore if it is
         MSG   ØØ44                    Display message
         B     CONFREAD
         SPACE
CONF5ØØ  MSG   ØØ45                    Configuration file note found
         B     CONF9ØØ
         SPACE
CONF6ØØ  C     R15,EOFRC               Is it End-Of-File?
         BE    CONF8ØØ                 Yes, all done
         MSG   ØØ46,RC                 Error reading configuration file
         OI    CSCFLGØ1,CFGERROR       Remember to close the shop
         B     CONF9ØØ
         SPACE
CONF8ØØ  FSCLOSE FSCB=CFGFILE          Close configuration file
         TM    CFGOPTS,CFGLOCAL+CFGRMTE Valid Local/Remote combination?
         BZ    CONF9ØØ                 Yes, nothing found
         BO    CONF81Ø
         NI    CSCFLGØ1,X'FF'-CSCAPPC  Invalid, reset option
         MSG   ØØ47                    Display error message
         MSG   ØØ48
         B     CONF9ØØ
         SPACE
CONF81Ø  OI    CSCFLGØ1,CSCAPPC        Set APPC (remote) option
```

```
CONF900  L     R14,CONFSV14
         BR    R14
         SPACE 3
*
*   Add entry to RND table
*
*
ADDRSRCE EQU   *                       Add entry to RND Table
         USING RNDSECT,R1
         ST    R14,ADDRSV14
         SR    R0,R0                   No table to look up
         GO    CSCSCN                  Get value
         BNZ   ADDR500
         LA    R0,8                    Maximum length for node name
         CR    R0,R1
         BL    ADDR600                 Too long
         MVC   CFGNODE,SCANUPP         Save Node name
         SR    R0,R0                   No table to look up
         GO    CSCSCN                  Get value
         BNZ   ADDR500                 Nothing found
         LA    R0,8                    Maximum length for resource name
         CR    R0,R1
         BL    ADDR600                 Too long
         MVC   CFGNAME,SCANUPP         Save Resource name
         LA    R0,RNDSIZE              RND entry length in double words
         LINK  OBTAIN                  Allocate storage
         L     R2,RNDPTR               Address first entry
         ST    R1,RNDPTR               Make it second
         ST    R2,RNDFWD
         SR    R0,R0
         ST    R0,RNDOPT1              Clear all option bytes
         ST    R0,RNDPIDS              Clear Send PATHID
         ST    R0,RNDPIDR              Clear Receive PATHID
         MVC   RNDNODE,CFGNODE         Move node and resource names
         MVC   RNDRSRC,CFGNAME
         L     R14,ADDRSV14
         CR    R14,R14                 Generate zero cc
         BR    R14
         SPACE
ADDR500  MSG   0050                    Missing value
         B     ADDR900
         SPACE
ADDR600  MSG   0053                    Value too long
*        B     ADDR900
         SPACE
ADDR900  L     R14,ADDRSV14
         LTR   R14,R14                 Generate non-zero cc
         BR    R14
         SPACE 3
*
```

```
*   Validate RND entry
*
*
VALRSRCE EQU   *                     Validate RND entry created
         ST    R14,VALRSV14
         SR    RØ,RØ                 No table to look up
         GO    CSCSCN                Get value
         BZ    VALR3ØØ               Too much, invalid operand
         L     R1,RNDPTR             Address RND table
VALR1ØØ  L     R1,RNDFWD             Skip first entry (new)
         LTR   R1,R1                 Check all entries
         BZ    VALR2ØØ
         CLC   CFGNODE,RNDNODE       Node names must be unique
         BE    VALR4ØØ               No good, duplicate found
         CLC   CFGNAME,RNDRSRC       Resource names must be unique
         BE    VALR5ØØ
         B     VALR1ØØ
         SPACE
VALR2ØØ  L     R1,RNDPTR             Address current entry again
         L     R14,VALRSV14
         CR    R14,R14               Generate zero cc
         BR    R14
         SPACE
VALR3ØØ  MSG   ØØ52                  Unexpected operand
         B     VALR9ØØ               Delete created entry
         SPACE
VALR4ØØ  LA    R2,CFGNODE            Duplicate Node name
         MSG   ØØ81
         B     VALR9ØØ
         SPACE
VALR5ØØ  LA    R2,CFGNAME            Duplicate Resource name
         MSG   ØØ82
*        B     VALR9ØØ
         SPACE
VALR9ØØ  L     R1,RNDPTR             Address new (first) entry
         L     R2,RNDFWD             Address second entry or zero
         ST    R2,RNDPTR             Remove first entry from list
         LA    RØ,RNDSIZE            RND entry length in double words
         LINK  RELEASE               Release storage
         L     R14,VALRSV14
         LTR   R14,R14               Generate non-zero cc
         BR    R14
         DROP  R1
         SPACE 3
         LTORG
         SPACE 3
CONFSV14 DS    F                     Save R14 - CONFIG
ADDRSV14 DS    F                                ADDRSRCE
VALRSV14 DS    F                                VALRSRCE
CFGCSV14 DS    F                                CONFIG commands
```

```
EOFRC    DC    F'12'                    End of file reached
FFFFFFFF DC    X'FFFFFFFF'              Mask to reverse bits
CPRESPB  DC    X'40000000'              Send CP response to user buffer
         SPACE
CFGFILE  FSCB  'CSC      CONFIG   * ',FORM=E
         SPACE
CFGWORK  DS    D                        Work area for conversions
CFGMSGC  DC    C'MSG     '              CP command to send messages
CFGRSCS  DC    C'RSCS    '              Default RSCS Service Machine-id
CFGNAME  DS    CL8                      Temporary Name
CFGNODE  DS    CL8                               Node
CFGUSER  DS    CL8                               User-id
CFGPREF  DS    X                                 Prefix value
CFGATTR  DS    X                                 Attribute value
CFGCLASS DS    X                                 Class
CFGOPTS  DS    X                        CFG Option byte
CFGLOCAL EQU   X'80'                    Local statement processed
CFGRMTE  EQU   X'40'                    Remote statement processed
DOTS     DC    C'... '
CFGARBCH EQU   C'*'                     Default Arbitrary Character
CFGANYCH EQU   C'%'                     Default Any Character
         SPACE
USERID   DS    CL8                      Response from CP Query USERID
         DS    CL4                      ' AT '
NODEID   DS    CL12
USERIDL  EQU   *-USERID
CPQUSER  DC    C'QUERY USERID'          CP command QUERY USERID
CFGTTL   DS    CL30                     Title line
         SPACE
PFXRST   DS    0D                       Prefix for TOF, EOF, and RST
         DC    A(0),C'>',AL1(EDSHIGH+EDSWHITE+EDSREVV),X'0000',CL8' '
         SPACE
CFGTABLE CMMD  (I,00,03,'DFRECS     ',DFRECS),   Configuration cmmds   *
               (I,00,03,'DFSIZE     ',DFRECS),                         *
               (I,00,01,'EVENT      ',EVENT),                          *
               (I,00,01,'LOCAL      ',LOCAL),                          *
               (I,00,01,'MESSAGE    ',MSG),                            *
               (I,00,03,'MSG        ',MSG),                            *
               (I,00,01,'OPTIONS    ',OPTIONS),                        *
               (I,00,01,'PREFIX     ',PREFIX),                         *
```

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

Fernando Duarte
Analyst (Canada)

Calculating lines of code

GENERAL DESCRIPTION

This procedure will help you calculate the number of lines of code for one or more files.

The procedure is called by typing:

```
LINESOFC fn ft fm
```

Note: wildcards are allowed.

LINESOFC EXEC

```
/*********************************************************************/
/* Calculating lines of code for one or more files                   */
/*********************************************************************/
/* LINESOFC <fn <ft <fm>>>                                           */
/*                 fn ft fm          : files to calculate            */
/*                                   : (wildcards are allowed)       */
/*********************************************************************/
trace off
parse upper arg fn ft fm .
if fn = '?' then signal help
if fn = '' then fn = '*'
if ft = '' then ft = '*'
if fm = '' then fm = 'A'
'SET CMSTYPE HT'
/*********************************************************************/
/* Selecting files                                                   */
/*********************************************************************/
'LISTFILE' fn ft fm '(STACK FIFO'
'GLOBALV SELECT $$LOC$$'
'GLOBALV SET $anz Ø'
anzfiles = queued()
do i = 1 to anzfiles
   pull file.i
end
do i = 1 to anzfiles
   'XEDIT' file.i '(PROF LINESOF1'
   mod = i // 25
   if mod = Ø then do
      'VMFCLEAR'
      call sayrt '...' i 'of' anzfiles 'files done'
```

```
   end
end
'GLOBALV SELECT $$LOC$$ GET $ANZ'
/*********************************************************************/
/* End                                                               */
/*********************************************************************/
ende:
'SET CMSTYPE RT'
say '============================================================='
say $anz 'lines found in' anzfiles 'files'
say '============================================================='
say ' '
say 'Please press ENTER'
pull .
exit

/*********************************************************************/
/* SAY combined with HT and RT                                       */
/*********************************************************************/
sayrt:
parse arg text
'SET CMSTYPE RT'
say text
'SET CMSTYPE HT'
return
/*********************************************************************/
/* Help                                                              */
/*********************************************************************/
help:
'VMFCLEAR'
address cms 'type linesofc exec * 1 Ø7'
```

LINESOF1 XEDIT

```
/* XEDIT macro to calculate the size of a file (number of lines) */

'GLOBALV SELECT $$LOC$$ GET $anz'
'EXTRACT /SIZE'
anz = $anz + size.1
'GLOBALV SELECT $$LOC$$ SET $anz' anz
'QQUIT'
```

Dr Reinhard Meyer (Germany)

Displaying ‘pseudo-graphics’

Having become tired of reading through long columns of data, I decided that I would like the data to be displayed with graphics – in the same way as on a PC. Following the inspiration of the ‘pseudo-graphics’ of our CICS monitor, I wrote the following two programs:

- GRAPH1 – which displays up to two data areas on the same Y axis.
- GRAPH2 – which displays up to two data areas with a Y axis for each data area.

REQUIREMENTS

The program was written for VM/ESA 2.1.0 but should work with other releases as well.

It uses the IBM product IOS3270-5785-HAX to display the panels and it needs to be run on screens with at least 32 lines.

The programs rely on the IBM module VMFE2E to transfer data to and from the GRAPH1 and GRAPH2 routines.

VARIABLES AND DATA AREAS

The following variables or data areas have to be set before you can call GRAPH1 or GRAPH2:

- X – the data for the x-axis. ‘X’ has to be set to '' as the initial value.
- Y1 – the first data area.
- Y1.0 – the number of records in the first data area.
- Y1TITLE – the title for the first data area.
- NBR – the number of the first displayed data item of Y1 and Y2.
- PF – the key definitions to be included from file PF IOS3270.

The following variables or data areas are optional:

- Y2 – the second data area.
- Y2.0 – the number of records in the second data area.
- Y2TITLE – the title for the second data area.
- CLEAR – 'CLEAR' or 'NOCLEAR' the screen before output (IOS3270).
- TITLE – the panel title.
- MESSAGE – display this text in the message line.
- MAXY1 – the maximum Y1 axis setting. This can have the following settings:
  - YES – use the maximum Y1 value of the complete stem to calculate the Y1 axis and the average for Y1 (all pages have the same Y1 axis).
  - Other value – use the maximum Y1 value of the current page to calculate the Y1 axis and the average for Y1 (so each page may look different).

  Note: in the GRAPH1 EXEC this variable is also used for the Y2 data values (because Y1 and Y2 are shown on the same axis).
- MAXY2 – the maximum Y2 axis setting. This can have the following settings:
  - YES – use the maximum Y2 value of the complete stem to calculate the Y2 axis and the average for Y2 (all pages have the same Y2 axis).
  - Other value – use the maximum Y2 value of the current page to calculate the Y2 axis and the average for Y2 (so each page may look different). This variable can only be set for the GRAPH2 EXEC.
- HIGHY1 – the highest value to be displayed on the Y1 axis. If values have been truncated because of HIGHY1 then a red plus sign (+) is shown at the upper side of the Y1 axis. If MAXY1 is set to YES, then the red plus will be shown if any Y1 data value

is truncated. If MAXY1 is not set, the red plus will only be shown if any Y1 data value on the current page is truncated. In the GRAPH1 EXEC this variable is also used for the Y2 data values (because Y1 and Y2 are shown on the same axis).

- HIGHY2 – the highest value to be displayed on the Y2 axis. If values have been truncated because of HIGHY2, a red plus is shown at the upper side of the Y2 axis. If MAXY2 is set to YES, then the red plus will be shown if any Y2 data value is truncated. If MAXY2 is not set, the red plus will only be shown if any Y2 data value on the current page is truncated. This variable can only be set for the GRAPH2 EXEC.
- LOWY1 – the lowest value to be displayed on the Y1 axis. If values are not shown, because of LOWY1, a red plus is shown at the lower side of the Y1 axis. If MAXY1 is set to YES, the red plus will be shown if any Y1 data value is truncated. If MAXY1 is not set then the red plus will only be shown if any Y1 data value on the current page is truncated. In the GRAPH1 EXEC this variable is also used for the Y2 data values (because Y1 and Y2 are shown on the same axis).
- LOWY2 – the lowest value to be displayed on the Y2 axis. If values are not shown, because of LOWY2, a red plus is shown at the lower side of the Y2 axis. If MAXY2 is set to YES then the red plus will be shown if any Y2 data value is truncated. If MAXY2 is not set then the red plus will only be shown if any Y2 data value on the current page is truncated. This variable can only be set for the GRAPH2 EXEC.
- ATTRY1 – the colour attributes for the Y1 data values. If you code 'Highlight=Default', the data values won't be displayed. (Default: Highlight=reverse Color=blue).
- ATTRY2 – the colour attributes for the Y2 data values. If you code 'Highlight=Default', the data values won't be displayed. (Default: Highlight=reverse Color=yellow).
- ATTROV – the colour attributes for overlaying data values. If you code 'Highlight=Default', the data values won't be displayed. (Default: Highlight=reverse Color=green).

- ATTRAY1 – the colour attributes for the Y1 average line. (Default: Highlight=default Color=blue).
- ATTRAY2 – the colour attributes for the Y2 average line. (Default: Highlight=default Color=yellow).
- ALINEY1 – a setting of 'NO' means the average line for the Y1 data area is not displayed.
- ALINEY2 – a setting of 'NO' means the average line for the Y2 data area is not displayed.

Note: If you define any of the ATTR... variables it should correspond to the .JX control of IOS3270:

- Highlight= Blink Default Reverse Underscore
- Color= Blue Default Green Pink Red Turquoise White Yellow

The following values will be set in the calling EXEC by GRAPH1 or GRAPH2:

- IOSK – pressed key (returned from IOS3270). This could be ENTER, PFnn, PA1, or PA2.
- IOSC – the cursor position rrccc (row and column).
- ZINPUT – the last entered command in the command line.
- CLEAR – NOCLEAR as set by the IOS subroutine.

DISPLAYING A GRAPHIC

The following code shows an example of how to display a graphic:

```
 1: /* Example of how to use GRAPH1 */
 2:          title = center('CPU% versus IORATE',40);
 3:          maxy1 = 'YES';
 4:          nr = 69;
 5:          call stat;
 6:          call display;
 7:          if iosk = 'PF03'
 8:            then signal exit;
 9:          /* Go back one page or to begin of data */
10:          if iosk = 'PF07'
11:            then
12:              do;
```

```
13:                nbr = max(nbr-nr,1);
14:                signal l1;
15:              end;
16:          /* Go forward one page or to end of data */
17:          if iosk = 'PF08'
18:            then
19:              do;
20:                nbr = min(nbr+nr,max(records-nr,1));
21:                signal l1;
22:              end;
23:          signal l0;
24: exit:
25:          exit;
26: /*************************************
27: * STAT - create data to be displayed *
28: *************************************/
29: stat:
30:          x. = ' ';
31:          /* Create 3 pages of data */
32:          do i = 1 to 207;
33:            y1.i  = random(0,100);
34:            y2.i  = random(0,300);
35:          end;
36:          /* Create description for the x-axis */
37:          x.1  = '10:21'; x.10 = '10:30'; x.20 = '10:40';
38:          x.30 = '10:50'; x.40 = '11:00'; x.50 = '11:10';
39:          x.60 = '11:20'; x.69 = '11:29';
40:          x.70 = '11:30'; x.79 = '11:39'; x.89 = '11:49';
41:          x.99 = '11:59'; x.109= '12:09'; x.119= '12:19';
42:          x.129= '12:29'; x.138= '12:38';
43:          x.139= '12:39'; x.148= '12:48'; x.158= '12:58';
44:          x.168= '13:08'; x.178= '13:18'; x.188= '13:28';
45:          x.198= '13:38'; x.207= '13:47';
46:          x.0 = i;
47:          y1.0 = i;
48:          y2.0 = i;
49:          records = i;
50:          nbr = 1;
51:          return;
52: /*************************
53: * DISPLAY - display data *
54: *************************/
55: display:
56:          message = '';
57:          if nbr = 1
58:            then
59:              if records <= nr
60:                then pf = '.i pf ;eonly';
61:                else
62:                  do;
```

```
63:                    pf = '.i pf ;efirst';
64:                    message = 'This is the first page';
65:                  end;
66:            else
67:              if nbr >= records-nr
68:                then
69:                  do;
70:                    pf = '.i pf ;elast';
71:                    message = 'This is the last page';
72:                  end;
73:                else pf = '.i pf ;emiddle';
74:          y1title = 'CPU%';
75:          y2title = 'IORATE';
76:          'EXEC GRAPH1';
77:          return;
```

An explanation of the code follows:

- Line 2 – this text is displayed in the middle of the first line of the panel.
- Line 3 – tells the EXEC to calculate the range of the Y axis from all data values of Y1 and Y2.
- Line 4 – the number of data values displayed on one page. GRAPH1 displays 69 values and GRAPH2 displays 59.
- Line 29 – you should place your data extraction routine here.
- Line 30 – clear the labels for the X axis.
- Line 33 – set all Y1 values between 0 and 100.
- Line 37 – create the labels for the X axis.
- Line 46 – set the number of labels for the X axis.
- Line 47 – set the number of values for the Y1 data area.
- Line 48 – set the number of values for the Y2 data area.
- Line 49 – save the number of data values for correct paging.
- Line 50 – the first data record to be displayed. If you move this line after line 4 then the display doesn't always jump back to page one after you press ENTER.

- Line 56 – reset the message line on the panel.
- Line 64 – the text in this variable is displayed on the message line.
- Line 74 – the title for the Y1 data values.
- Line 75 – the title for the Y2 data values. If you do not specify this variable then no data for Y2 will be displayed.
- Line 76 – execute program to display graphic.

Note: non-numeric values or unset values (ie the REXX function SYMBOL would return 'LIT') are ignored and are not counted for the average values.

The above sample will create the screen shown in Figure 1. The following lines can be seen on the screen:

- Line 3 – the title and colour symbol for the Y1 data area on the left

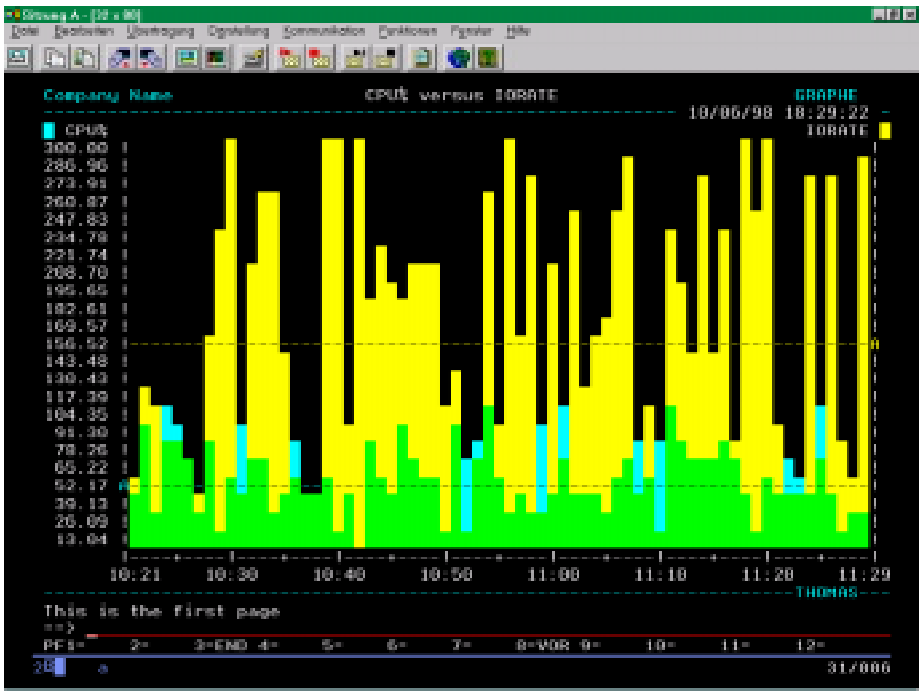

Figure 1: Screen created by sample code

(CPU%) and for the Y2 data area on the right (IORATE).

- Line 15 – the average line for the Y2 data area (shown in Figure 1 as '- - - A').
- Line 23 – the average line for the Y1 data area (shown in Figure 1 as 'A- - -').
- Line 29 – your VM user-id (shown in Figure 1 as '- - THOMAS - -').
- Line 30 – the message line. This shows the content of variable MESSAGE or the return code from the last command entered on line 31.
- Line 31 – the command line. You can enter any valid CP or CMS command.
- Line 32 – definition of PF keys from file PF IOS3270.

GRAPH1 EXEC

This is the program to display one Y-axis.

```
/* GRAPH1: display data in graphical format using 1 y-coordinate     */
         maximum1 = 0;     /* max. datavalue for Y1+Y2               */
         minimum1 = 0;     /* min. datavalue for Y1+Y2               */
         average1 = 0;     /* average for Y1                         */
         average2 = 0;     /* average for Y2                         */
         count1   = 0;     /* number of values in Y1                 */
         count2   = 0;     /* number of values in Y2                 */
         y1ctrl   = '(';   /* Control character for Y1 data (ea=1)   */
         y1char   = 'X';   /* Character for Y1 data         (ea=0)   */
         y2ctrl   = ')';   /* Control character for Y2 data (ea=1)   */
         y2char   = '+';   /* Character for Y2 data         (ea=0)   */
         ovctrl   = '/';   /* Control character for overlay (ea=1)   */
         ovchar   = '*';   /* Character for overlay         (ea=0)   */
         /* does the terminal support extended attributes? yes: ea=1 */
         ea = bitand(substr(diag('8C'),1,1),'40'x) = '40'x;
         if c2d(substr(diag('8C'),5,2)) < 32
           then
             do;
               iosk = 'PF03';
               say 'Screen is too small. Must be at least 32 lines.';
               signal exit;
             end;
         /*************************************************************
```

```
* Get data from calling EXEC:                                    *
* X.       data for x-axis.                                      *
* Y1.      first data area for y-axis.                           *
* Y2.      second data area for y-axis.                          *
* Y1TITLE  title for first data area.                            *
* Y2TITLE  title for second data area.                           *
* NBR      number of first displayed data item of Y1. and Y2.*
* CLEAR    CLEAR or NOCLEAR screen before output (IOS3270).  *
* TITLE    panel title.                                          *
* PF       which key definitions to include from PF IOS3270. *
* MESSAGE  Display this text in the message line.            *
* MAXY1    YES = Use maximum Y1 or Y2 value of complete stem *
*          to calculate Y-axis and not only current page.    *
* HIGHY1   Highest value to be displayed on the Y1 axis.     *
* LOWY1    Lowest value to be displayed on the Y1 axis.      *
* ATTRY1   Color attributes for the Y1 data values.          *
* ATTRY2   Color attributes for the Y2 data values.          *
* ATTROV   Color attributes for overlaying data values.      *
* ATTRAY1  Color attributes for the Y1 average line.         *
* ATTRAY2  Color attributes for the Y2 average line.         *
* ALINEY1  'NO' = don't display average line for Y1.         *
* ALINEY2  'NO' = don't display average line for Y2.         *
************************************************************/
'VMFE2E GET X. Y1. Y2. Y1TITLE Y2TITLE NBR CLEAR TITLE PF',
  'MESSAGE MAXY1 HIGHY1 LOWY1 ATTRY1 ATTRY2 ATTROV ATTRAY1',
  'ATTRAY2 ALINEY1 ALINEY2';
upper maxy1 maxy2 aliney1 aliney2;
/* Set default or user defined attributes for data values */
if attry1 ¬= 'ATTRY1' & attry1 ¬= ' '
  then attry1 = '.jx Set Ctl (' attry1;
  else attry1 = '.jx Set Ctl ( Hig=reverse Col=blu';
if attry2 ¬= 'ATTRY2' & attry2 ¬= ' '
  then attry2 = '.jx Set Ctl )' attry2;
  else attry2 = '.jx Set Ctl ) Hig=reverse Col=yel';
if attrov ¬= 'ATTROV' & attrov ¬= ' '
  then attrov = '.jx Set Ctl /' attrov;
  else attrov = '.jx Set Ctl / Hig=reverse Col=gre';
if attray1 ¬= 'ATTRAY1' & attray1 ¬= ' '
  then attray1= '.jx Set Ctl <' attray1;
  else attray1= '.jx Set Ctl < Hig=default Col=blu';
if attray2 ¬= 'ATTRAY2' & attray2 ¬= ' '
  then attray2= '.jx Set Ctl >' attray2;
  else attray2= '.jx Set Ctl > Hig=default Col=yel';
/* create header line */
if ea
  then
    do;
      header = 'FF'x substr(y1title,1,37);
      if y2title ¬= ' ' & y2title ¬= 'Y2TITLE'
        then header = header right(strip(y2title),37) 'FF'x;
```

```
    end;
  else
    do;
      header = y1char substr(y1title,1,37);
      if y2title ¬= ' ' & y2title ¬= 'Y2TITLE'
        then header = header right(strip(y2title),37) y2char;
    end;
/* calculate maximum and average values for Y1. and Y2. */
if maxy1 = 'YES'
  then
    do;
      from  = 1;
      count = y1.0;
    end;
  else
    do;
      from  = nbr;
      count = 69;
    end;
do i = from for count;
  if datatype(y1.i) = 'NUM'
    then
      do;
        maximum1 = max(maximum1,y1.i);
        minimum1 = min(minimum1,y1.i);
        average1 = average1 + y1.i;
        count1   = count1 + 1;
      end;
  if datatype(y2.i) = 'NUM'
    then
      do;
        maximum1 = max(maximum1,y2.i);
        minimum1 = min(minimum1,y2.i);
        average2 = average2 + y2.i;
        count2   = count2 + 1;
      end;
end;
/* Calculate upper and lower limit for the Y-axis */
if datatype(lowy1) ¬= 'NUM'
  then lowy1 = 0;
if datatype(highy1) ¬= 'NUM'
  then highy1 = maximum1;
/* calculate stepwidth for the Y-axis */
step1 = (highy1-lowy1) / 23;
if count1 > 0
  then
    do;
      average1 = average1/count1;
      avgline1 = trunc(average1/step1+.999);
    end;
```

```
if count2 > Ø
  then
    do;
      average2 = average2/count2;
      avgline2 = trunc(average2/step1+.999);
    end;
c. = ' ';
/* draw y-axis and description                    */
/* Y-values <= 999 are displayed with 2 decimals  */
/* Y-values >  999 are displayed without decimals */
v.   = ' |'copies(' ',69)'|';
do i = 1 to 23;
  if highy1 > 999
    then v.i = format(step1*i+lowy1,6,Ø)v.i;
    else v.i = format(step1*i+lowy1,3,2)v.i;
end;
/* + on the upper corner of the y-axis means that */
/* values have been truncated due to highy1.      */
if maximum1 > highy1
  then
    do;
      v.23 = overlay('+',v.23,7);
      c.23 = overlay('#',c.23,7);
    end;
/* draw average line for Y1. */
if average1 > lowy1 & average1 < highy1 & aliney1 ¬= 'NO'
  then
    do;
      v.avgline1 = overlay('A'repeat('-',68),v.avgline1,8);
      c.avgline1 = overlay(repeat('<',69),c.avgline1,8);
    end;
/* draw average line for Y2. */
if average2 > lowy1 & average2 < highy1 & aliney2 ¬= 'NO'
  then
    do;
      v.avgline2 = overlay(repeat('-',68)'A',v.avgline2,9);
      c.avgline2 = overlay(repeat('>',69),c.avgline2,9);
    end;
/* draw Y1 data values */
k = Ø;
do i = nbr for 69;
  k = k + 1;
  if datatype(y1.i) = 'NUM' & y1.i > lowy1
    then
      do;
        j = trunc((min(y1.i,highy1)-lowy1)/step1+.999);
        do ii = 1 to j;
          /* Display character or color */
          if ea
            then c.ii = overlay(y1ctrl,c.ii,k+8,1);
```

```
            else v.ii = overlay(y1char,v.ii,k+8,1);
        end;
      end;
end;
/* draw Y2 data values */
k = Ø;
do i = nbr for 69;
  k = k + 1;
  if datatype(y2.i) = 'NUM' & y2.i > lowy1
    then
      do;
        j = trunc((min(y2.i,highy1)-lowy1)/step1+.999);
        do ii = 1 to j;
          if (ea = 1 & substr(c.ii,k+8,1) = y1ctrl) |,
            (ea = Ø & substr(v.ii,k+8,1) = y1char)
            then
              /* Display overlay character or color */
              if ea
                then c.ii = overlay(ovctrl,c.ii,k+8,1);
                else v.ii = overlay(ovchar,v.ii,k+8,1);
            else
              /* Display character or color */
              if ea
                then c.ii = overlay(y2ctrl,c.ii,k+8,1);
                else v.ii = overlay(y2char,v.ii,k+8,1);
        end;
      end;
end;
/* draw X axis */
b1 = '       'copies('|——+——',7)'|';
/* + on the lower corner of the y-axis means that */
/* values have been truncated due to lowy1.       */
if lowy1 > minimum1
  then
    do;
      v.1 = overlay('+',v.1,7);
      c.1 = overlay('#',c.1,7);
    end;
zf1 = nbr+9;
zf2 = nbr+19;
zf3 = nbr+29;
zf4 = nbr+39;
zf5 = nbr+49;
zf6 = nbr+59;
zf7 = nbr+68;
b2 = '     ';
if nbr <= x.Ø
  then b2 = b2||center(x.nbr,7)'  ';
if zf1 <= x.Ø
  then b2 = b2||center(x.zf1,7)'   ';
```

```
          if zf2 <= x.Ø
            then b2 = b2||center(x.zf2,7)'   ';
          if zf3 <= x.Ø
            then b2 = b2||center(x.zf3,7)'   ';
          if zf4 <= x.Ø
            then b2 = b2||center(x.zf4,7)'   ';
          if zf5 <= x.Ø
            then b2 = b2||center(x.zf5,7)'   ';
          if zf6 <= x.Ø
            then b2 = b2||center(x.zf6,7)'  ';
          if zf7 <= x.Ø
            then b2 = b2||center(x.zf7,7);
          if ea
            then pname = 'GRAPHE';
            else pname = 'GRAPHM';
          /* Display panel */
          call ios pname '*';
          /*********************************************
          * Return values to calling EXEC:              *
          *   IOSK   pressed key (IOS327Ø)              *
          *   IOSC   Cursor position rrccc              *
          *   ZINPUT last entered command in commandline *
          *   CLEAR  NOCLEAR as set by IOS subroutine   *
          *********************************************/
exit:     'VMFE2E SET IOSK IOSC ZINPUT CLEAR';
          return;
/***********************
* IOS - Show the Panel *
***********************/
ios:
          'NUCXLOAD IOS327Ø';
          parse upper arg i1 i2 .;
          wer   = userid();
          date  = date('E');
          time  = time();
          pname = i1
          'IOS327Ø' i1 '( PA2 SUBSET' cursor clear ')'
          clear = 'NOCLEAR';
          if rc = 1 | rc = 2 | rc = 3 | rc = 5
            then
              do;
                say 'The panel' i1 'is not available.';
                say 'Please press the ENTER key';
                'CP SLEEP';
                exit;
              end;
          cursor = 'ØØØ1';
          message = ' ';
          if IOSK  = 'PFØ3'
            then
```

```
    if i2 = '*'
      then return;
      else signal value strip(i2);
if IOSK  = 'PFØ1'
  then
    do;
      'IOS327Ø' i1 'IOSHELP (' clear;
      signal 'IOS';
    end;
if input = ''
  then return;
input = strip(input,'T');
upper input;
interpret 'input';
message = 'Returncode' rc 'from' input;
zinput = input;
input = '';
signal 'IOS';
```

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

Thomas Rupp
Vorarlberger Illwerke AG (Austria)

Packing files

This procedure allows you to save as much space as possible by packing all files that are currently not packed.

The syntax is:

```
PACKALL fn ft <fm>
```

where fn ft fm are the files to be packed (wildcards are allowed).

PACKALL EXEC

```
/*********************************************************************/
/* Packing all files that are still unpacked                         */
/*********************************************************************/
/* Call:   PACKALL fn ft <fm>                                        */
/*               fn, ft, fm          : selected files (wildcards OK) */
/*********************************************************************/
```

```
trace off
parse upper arg fn ft fm .
if fn = '?' then signal hilfe
if fm = '' then fm = 'A'
'MAKEBUF'
'LISTFILE' fn ft fm '(ALL STACK FIFO'
anz = queued()
do i = 1 to anz
   pull xfn xft xfm xrecf xrecl .
   if xrecf = 'F' & xrecl = '1Ø24' then iterate
   if xrecf = '-' then iterate
   if xrecl = '-' then iterate
   'EXEC PACK' xfn xft xfm
end

ende:
'DROPBUF'
exit
/*********************************************************************/
/* Help                                                              */
/*********************************************************************/
hilfe:
'VMFCLEAR'
address cms 'type packall exec * 1 Ø6'
```

PACK EXEC

```
/*********************************************************************/
/* Packing files                                                     */
/*********************************************************************/
/* Call:   PACK fn ft fm                                             */
/*              fn ft fm              : file to be packed            */
/*********************************************************************/
parse upper arg fn ft fm .
if fn = '?' then signal hilfe
if fm = '' then fm = 'A'
'COPYFILE' fn ft fm '= = = (PACK OLDD REPL'
exit rc
/*********************************************************************/
/* Help                                                              */
/*********************************************************************/
hilfe:
'VMFCLEAR'
address cms 'type pack    exec * 1 Ø6'
```

IBM has announced Version 4 Release 7 of ACF/SSP (Advanced Communications Facility/ System Support Program) for VM/ ESA, OS/390, MVS/ESA, and VSE/ESA. This includes SSP support for the new functions in ACF/NCP 7.7 together with Frame-Relay logical line trace support offering increased diagnostic capability.

ACF/NCP 7.7 enhancements include: new High-Level Assembler support; BAN connection balancing on 3745 lines; switched support and CIR support for NCP-controlled 3746-900 lines; support for IP over Frame-Relay-switched 3745 connections; new APPN topology management support; and improved support for subarea dial connections.

For further information contact your local IBM representative.

* * *

Computer Associates has announced Release 3.2 of its CA-VMLib library management and control system for the CMS environment. CA-VMLib provides centralized management of files accessed by multiple users and tracks changes to those files, preventing duplicate files, and coordinating file changes. Release 3.2 now supports VM/XA and VM/ESA, running in the XA, ESA, or 370 system modes, while still supporting the VM/SP environment.

Enhancements in Version 3.2 include REXX procedures to replace the report load modules used to generate CA-VMLib reports, allowing modification of these procedures to customize the CA-VMLib reports.

CA-VMLib supports cross-reference reporting for PARENT elements and allows users to display or print the COPY/ INCLUDEs for a specific PARENT element, also stating the line number of the COPY/INCLUDE.

An integrated SCAN facility allows the scanning of all or selected CA-VMLib elements, residing on disk, for character strings. Pattern matching, similar to the CMS LISTFILE command, can be used to obtain a subset of CA-VMLib element names.

Release 3.2 also offers the option of a menu-driven interface, which can be selected during installation. This allows specification of CA-VMLib subcommands on ISPF-like panels.

For further information contact:
Computer Associates, One Computer Associates Plaza, Islandia, NY 11788-7000, USA.
Tel: (516) 342 5224.
Computer Associates, Computer Associates House, 183-187 Bath Road, Slough, Berks, SL1 4AA, UK.
Tel: (01753) 577733.
URL: http://www.cai.com.

* * *