153

VM update

May 1999

In this issue

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 £180.00 in the UK; $275.00 in the USA and Canada; £186.00 in Europe; £192.00 in Australasia and Japan; and £190.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1990 issue, are available separately to subscribers for £16.00 ($23.00) each including postage.

Editor

Robert Burgess

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*.

Keeping track of POWER data file usage

The usage of the POWER data file can be seen by means of the ‘D Q’ command. When this usage reaches a threshold (defined by the POWER macro SPLIM parameter), warning messages are shown on the VSE console. However, it may be desirable for operators to be alerted before this threshold is reached so that preparations for POFFLOAD can be made in an orderly manner.

This EXEC and Assembler program will enable you to determine the usage of the POWER data file from CMS. It may be used simply as a command, when the percentage utilization will be displayed at your CMS terminal. However, it might be more useful to call it at intervals as a function in an EXEC run by VMUTIL. For example:

```
if powfull()>8Ø then 'CP M' somebody 'POWER is' powfull()'% full'
if powfull()>95 then 'CP WNG' somebody 'START OFFLOAD NOW'
```

The Assembler code as it stands works for 3390 DASD – the source shows where to make changes to support other device types. Although I cannot test on 3380, I believe that the only changes would be to the number of 4096 bytes blocks per track (10 rather than 12).

The VSE/POWER Administration and Operation manual has a table of numbers for other CKD devices. The program does not currently support FBA devices. Instructions for generating the module are in the comments at the start of the program.

The POWFULL EXEC will need to be changed for the linkage address etc, as commented near the beginning. This has been tested under VM/ESA 2.2 with VSE/ESA 2.2.

POWFULL EXEC

```
/******************************
* Look at POWER Master Record *
******************************/
parse source . func .      /* how we were called                    */
                           /*----------------------------------------*/
                           /*    Installation-specific definitions   */
                           /*----------------------------------------*/
owner = 'VSEPROD'          /* machine with Q-file in directory entry */
owner_cuu = '2FØ'          /* Q-file virtual address for owner       */
```

```
rpass = 'VSEREAD'           /* read password                                */
l = 'VSE.POWER.QUEUE.FILE' /* Label of Queue File                          */
                            /*---------------------------------------------*/
call qlink                      /* make the link     */
call extent l                   /* get extents       */
'POWMAST' our_cuu cyl head      /* get Master Record */
pull master                     /* from stack        */
call detlink                    /* lose the link     */
   /* get statistics from Master Record */
parse var master versid 5 . 49 mmax 53 mfree 57 muse 61 mbad 65 .
   /* vers-id should be V6R1, but EXEC may be OK if not */
if versid¬='V6R1' then say 'POWER version is' versid
mmax  = c2d(mmax)                        /* Data blocks in total*/
mfree = c2d(mfree)                       /* Data blocks free    */
muse  = c2d(muse)                        /* Data blocks in use  */
mbad  = c2d(mbad)                        /* Data blocks bad     */
fullpct = format((mmax-mfree)*100/mmax,,0)
if func='FUNCTION' then return fullpct  /* REXX used "powfull()" */
say 'Data file is' fullpct'% full'      /* called as an EXEC     */
exit

/***********************
* Link Queue File disk *
***********************/
qlink:
'SET CMSTYPE HT'
'GETFMADR'        /* find spare device number and disk mode */
pull . fm our_cuu oldz .
'EXECIO * CP (SKIP STRING LINK' owner owner_cuu our_cuu 'RR' rpass
'ACCESS' our_cuu fm
'SET CMSTYPE RT'
return

/****************
* Detach Q file *
****************/
detlink:
'REL' fm
'EXECIO * CP (SKIP STRING DET' our_cuu
return

/****************************
* Get start of Q file extent *
****************************/
extent:
arg dsn
'PIPE CMS LISTDS' dsn fm '(EXTENT|STEM EXT.'
if ext.0¬=4 then
do
   say dsn 'not found on' our_cuu
   call detlink
   exit 16
end
parse var ext.4 . . cyl head .
return
```

POWMAST ASSEMBLE

```
POWMAST  TITLE 'Read Power Master Record'
****************************************************************
* This program passes the POWER Master Record to the caller    *
* via the CMS stack.                                           *
* Only the first 255 bytes are passed back (limit for stack),  *
* but these contain the usage statistics (how full, etc)       *
*                                                              *
* Called by POWFULL dddd ccccc hhhhh                           *
*                                                              *
*     where dddd is device number   (4 chars hex)              *
*           ccccc is start cylinder of Qfile (5 chars decimal) *
*           hhhhh is start head no. of Qfile (5 chars decimal) *
*                                                              *
* NB not free format - give number of characters shown         *
*                                                              *
*--------------------------------------------------------------*
*                                                              *
* Macros used are DIAG, REGEQU      (in DMSGPI)                *
*                 HCPSGIOP          (in HCPGPI)                *
*                                                              *
* Generate POWMAST MODULE by: GLOBAL MACLIB DMSGPI HCPGPI      *
*                             ASSEMBLE POWMAST                 *
*                             LOAD POWMAST                     *
*                             GENMOD POWMAST (ALL              *
*                                                              *
****************************************************************
         SPACE
POWMAST  CSECT
***********************************************************************
*                                                                     *
*                 Queue file settings for 339Ø                        *
*                                                                     *
*-------------------------------------------------------------------- *
*                                                                     *
*   The following installation-dependent variables must be set to     *
*   match the physical characteristics of the VSE/POWER files.        *
*                                                                     *
***********************************************************************
IQFTPC   EQU   15                    queue file tracks per cylinder
IQFRPT   EQU   12                     - blocks per track (339Ø)
*        EQU   1Ø                     - blocks per track (338Ø)
IQFQDBLK EQU   16                     - records per dblk
IQFQSZE  EQU   256                    - record size
         EJECT
         USING *,R12
         LR    R12,R15            set up base register
         B     PASTAMP
         DC    CL8'POWMAST'       eyecatcher
         DC    CL8'&SYSDATE'
         DC    CL8'&SYSTIME'
SAVRET   DC    F'Ø'               R14 on entry
PASTAMP  DS    ØH
         ST    R14,SAVRET
         SPACE
*        *****************************
*        * Get virtual device number, *
*        * start cylinder, start head *
*        *****************************
         SPACE
         MVC   QDEV,8(R1)         device no in characters
         TR    QDEV,TRTAB                   get it
         PACK  QDEV(L'QDEV+1),QDEV(L'QDEV+1)  into hex
         SPACE
         PACK  DOUB,16(5,R1)      pack start cyl
         CVB   R6,DOUB            make it hex
         STH   R6,QFBEGCC         store for CCW
         STH   R6,QFSEARCH        cyl addr for internal record
         SPACE
         PACK  DOUB,24(5,R1)      pack start head
         CVB   R6,DOUB            make it hex
         STH   R6,QFBEGHH         store for CCW
         STH   R6,QFHEAD          head addr for internal record
         SPACE
************************************************************
*     We read the first block on the Q file. This is the   *
*     "internal" record. The first 4 bytes of this contain *
*     the DBLK of the Master Record (at end of Q entries)  *
************************************************************
         SPACE
         BAL   R14,READINT        go read internal record
         SPACE
         MVC   QUEDBLK,QFREC      get DBLK for master record
         BAL   R14,READQF         go read master record
         SPACE
         LA    R1,SVCPARMS        stack it for caller
         SVC   2Ø2                 via CMS SVC
         DC    AL4(ERROR)         address of error routine
         SR    R15,R15            good return code
OUT      EQU   *
         L     R14,SAVRET         return to CMS
         BR    R14
         SPACE
ERROR    LA    R15,16             error
         B     OUT
         EJECT
*        ******************
*        * Read Queue file *
*        *******************
READQF   DS    ØH
         L     R3,QUEDBLK           convert queue DBLK to
         MH    R3,QFQDBLK+2          relative queue
         ST    R3,QUEDBLK             record number
         LH    R3,QFBEGCC           pick up beginning CC
         MH    R3,QFTPC+2           mult by tracks per cyl
         AH    R3,QFBEGHH           add beginning HH
         MH    R3,QFRPT+2           mult by recs per track
         MH    R3,QFQDBLK+2         mult by queues per QRB
         A     R3,QUEDBLK           add relative DBLK
         XR    R2,R2
         D     R2,QFQDBLK           divide by queues per QRB
         XR    R2,R2
         D     R2,QFRPT             divide by records per track
         LA    R2,1(R2)             convert rel rec no to seq rec no
         STC   R2,QFRNO             store record number
         SR    R2,R2                clear R2
         D     R2,QFTPC             divide by tracks per cyl
         STCM  R2,3,QFHEAD          store head number
         STCM  R3,3,QFSEARCH        store cylinder number
         SPACE
READINT  DS    ØH   enter here when disk addr. has been set already
         LA    R2,QFCCW             point to CCW chain
         L     R1,QDEV              point to device address
         SPACE
*        **********************************
*        * set up DIAG A8 parameter block *
*        **********************************
         LA    R7,DIAGBLK           parameter block
         USING SGIOP,R7
         STH   R1,SGIDEVNO          device number
         ST    R2,SGICPA            CCW address
         DIAG  R7,RØ,X'A8'          do synchronous I/O
         BNZ   ERROR                something wrong
         BR    R14                  return - OK
         SPACE
         REGEQU
         EJECT
***********************************************
*              Work Areas                     *
***********************************************
         SPACE
SVCPARMS DS    ØD                   Parameters for SVC2Ø2
         DC    CL8'ATTN'            Command
         DC    CL4'FIFO'            how to stack
         DC    AL1(255)             length of text to be stacked
         DC    AL3(QFREC)           address of text to be stacked
         DC    8X'FF'               end of PLIST
         SPACE
DOUB     DC    D'Ø'                 double-word work area
```

```
QDEV     DC    CL4' ',C' '
         SPACE
QRADDR   DS    ØF                  queue record address
         SPACE
QUEDBLK  DC    F'Ø'    Q record DBLK area (zero for internal record)
QFSEEK   DC    XL2'ØØ'             seek address
QFSEARCH DS    XL2'ØØ'             cyl
QFHEAD   DS    XL2'ØØ'             head
QFRNO    DC    X'Ø1'               record 1 for internal record
         SPACE
*********************************************
*      Installation-dependent values      *
*********************************************
* The following values reflect the physical VSE/POWER files
QFBEGCH  DS    ØF                  queue file start cylinder/head
QFBEGCC  DC    H'Ø'                start cylinder
QFBEGHH  DC    H'Ø'                start head
QFTPC    DC    A(IQFTPC)           queue file tracks per cylinder
QFRPT    DC    A(IQFRPT)           queue file records per track
QFQDBLK  DC    A(IQFQDBLK)         queue records per dblk
QFQSZE   DC    Y(IQFQSZE)          queue record size
         EJECT
         COPY  HCPSGIOP            DSECT of DIAG A8 parameter block
POWMAST  CSECT
         SPACE
***********************************
*     I/O areas and CCWs          *
***********************************
         SPACE
DIAGBLK  DC    (SGIDWSIZ)D'Ø'      parameter list for DIAG A8
         SPACE
*     only read 255 bytes of Internal Record or Master Record
QFCCW    CCW   X'Ø7',QFSEEK,X'4Ø',6     seek
         CCW   X'31',QFSEARCH,X'4Ø',5   search
         CCW   X'Ø8',*-8,Ø,Ø            TIC until found
         CCW   X'Ø6',QFREC,X'2Ø',255    read data - sili
         SPACE
QFREC    DC    CL255' '            Queue record block area
         SPACE
TRTAB    DC    XL256'ØØ'
         ORG   TRTAB+C'A'
         DC    X'ØAØBØCØDØEØF'     Make C'A' into X'ØA' etc
         ORG   TRTAB+C'Ø'
         DC    X'ØØØ1Ø2Ø3Ø4Ø5Ø6Ø7Ø8Ø9'  C'1' into X'Ø1' etc
         ORG
         END
```

John Illingworth
Systems Engineer
Wm Morrison Supermarkets (UK)

VM:Secure enhancement rules – part 2

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

Responses

When deleting a user’s Object file without the NOPROMPT option, VM:Secure will prompt you with the following messages:

```
VMXSYS0400I Do you wish to remove this user?
VMXSYS0404R Enter 'YES' or 'NO':
```

You should enter ‘YES’ if you want to remove the user’s Objects; enter ‘NO’ if you do not want to remove them. Any other response to the prompt will result in the following message:

```
    VMXSYS0431E Response 'answer' is invalid
```

followed by the prompt messages again. When VM:Secure successfully deletes the user’s Object Rules file, it will display the following message:

```
    VMXSYS8002I The user Objects have been removed for user-id
```

Return codes and error messages

Return codes and error messages for OBJDEL are shown in Figure 4.

Return code	Message number	Text
2	0038E	Missing parameter
10	0265E	Not authorized for OBJDEL user-id
28	8003E	User Object file for user-id does not exist
100	0099I	'OBJDEL' command cancelled
299	7000E	The OBJECT RULES are not active

Figure 4: OBJDEL return codes

OBJEDIT COMMAND

The OBJEDIT command allows an authorized user to create or modify a user's Object Rules file. You can add, delete, or change Object Rules statements and comments in the file. VM:Secure will check the entire file for Object Rules validity. OBJEDIT cannot be used from the VM:Secure server console. It has the format:

```
    OBJEDIT  userid1  [ userid2  [  (xeditparms
```

where:

- 'userid1' specifies the user-id of the Object Rules file modified (or created if it does not exist).
- 'userid2' specifies the prototype user Object Rules file to use when creating a new user-id's Object Rules file.
- 'xeditparms' specifies any valid CMS XEDIT command parameters you would like to use for the XEDIT session.

For example, to edit the user-id FRANK's Object Rules file, enter:

```
    vmsecure objedit frank
```

To create a new user-id Object Rules file for FRANK, using DOEJ's as a template, enter:

```
    vmsecure objedit frank doej
```

Responses

If VM:Secure finds an error in the file, it will display the type of error detected and will then prompt you with the following messages:

```
VMXSYSØ469I Do you wish to correct the problem?
VMXSYSØ4Ø4R Enter 'YES' or 'NO':
```

You should enter 'YES' if you want to go back into XEDIT to correct the problem. 'NO' will cancel the update and display:

```
    VMXSYS8Ø12I Objects not changed.
```

When the Object Rules file is successfully updated, VM:Secure displays the following message:

```
    VMXSYS8ØØ2I The User Objects have been loaded for user-id
```

Validation error messages

Validation error messages for OBJEDIT are shown in Figure 5. If any of the errors shown occur, the following prompt messages will also be displayed:

```
VMXSYS0056I On record recnum in file 'fn ft fm'
VMXSYS0469I Do you wish to correct the problem?
VMXSYS0404R Enter 'YES' or 'NO':
```

```
Message        Text
number

0039E          Invalid parameter 'parm'
8019E          Length of token word #num is less/more than
                  the defined min|max of length
8020E          Value for token word #num does not match any
                  of the allowed words
8022I          Defined: word1 word2 ... wordn 8200E
8200E          Object objectname does not exist
8201E          Tokens invalid for object objectname
8203E          Too many tokens specified for object objectname.
                  Maximum is num
8204E          Missing token num (no default) for object objectname
```

Figure 5: OBJEDIT validation error messages

Other OBJEDIT return codes and error messages are shown in Figure 6.

THE OBJFOR COMMAND

The OBJFOR command allows authorized users or administrators to query Object Rules authorization for another user-id. It has the format:

```
    OBJFOR  userid  objectname  [token-1] … [token-n]
```

where:

- ‘userid’ specifies the user-id to check for an Object Rule access.
- ‘objectname’ is the name of the object or resource being checked.

```
Return  Message      Text
 code   number


  2     82Ø1E     Tokens invalid or missing for object 'objectname'.
 1Ø     Ø265E     Not authorized for: OBJEDIT userid1
 11     Ø265E     Not authorized for: OBJLOAD userid1
 12     Ø38ØE     A read/write A disk is required
 14     Ø364E     File 'fn ft fm' is being updated
 16     Ø621E     Unexpected return code rc from COPYFILE
 17     Ø621E     Unexpected return code rc from COPYFROM
 22     Ø325E     Error 'rc' invoking 'XEDIT fn ft fm'
 24     ØØ38E     Missing parameter.
 28     8ØØ3E     User OBJECT file for userid2 does not exist.
1ØØ     ØØ99I     'OBJEDIT' command cancelled
299     7ØØØE     The OBJECT RULES are not active.
3ØØ     82Ø2E     Severe error rc reading 'file' from storage.
3Ø5     8ØØ5E     Error rc from EXECLOAD of fn ft fm
```

Figure 6: OBJEDIT return codes

The objectname should previously have been defined in an OBJDEF file and loaded for it to be valid.

- 'token-1...token-n' is the list of tokens associated with the object. The number of tokens specified must match the defined number in the Object Definition File (OBJDEF).

Usage

The primary use of the OBJFOR command is to validate whether a particular user-id would be allowed to access an object. It can be used by an administrator to verify that proper Object Rules have been defined for user-ids.

For the following examples, assume an object has been defined called REPORTS. It has two tokens, the first of which is the report name, with a minimum of four and a maximum of 12 characters, and the second is either READ or UPDATE. Also assume that the server machine name is VMSECURE and the user-id JOEUSER is defined with the Object Rules:

- To determine whether the user FRANK has UPDATE authority for a report named MGRSALARY, enter:

```
vmsecure objfor frank reports mgrsalary update
```

- To determine whether the user FRANK has READ authority for a report named STATUS, enter:

```
vmsecure objfor frank reports status read
```

Return codes

The OBJFOR command will return an ACCEPT or REJECT message for access to a resource for a particular user (see Figure 7). A zero return code denotes access would be granted; any non-zero return code would deny access.

```
Return   Text
code

  0     userid access would be ACCEPTED for token-1...token-n
298     userid access would be REJECTED for token-1...token-n
```

Figure 7: OBJFOR return codes and messages

Error messages

OBJFOR return codes and error messages are shown in Figure 8.

OBJLOG COMMAND

The OBJLOG command allows an authorized user to extract the Object Rules Audit file. It takes the format:

```
OBJLOG  EXTRACT [filename [filetype [filemode [(options
```

where:

- ‘filename’ specifies the file name you would like the audit file extracted to. The default is OBJECTS.
- ‘filetype’ specifies the file type you would like the audit file extracted to. The default is AUDIT.

```
Return   Message     Text
code     number

  2     8201E     Tokens invalid or missing for object 'objectname'.
  4     8206E     Token count does not match for object objectname.
                     Tokens allowed is count.
  6     8006E     Object name not specified.
 24     0038E     Missing parameter.
 28     8200E     Object objectname does not exist.
298     9001E     Access rejected for: 'objectname objectparms...'
299     7000E     The OBJECT RULES are not active.
300     8202E     Severe error rc reading 'file' from storage.
```

Figure 8: OBJFOR return codes

- ‘filemode’ specifies the file mode you would like the Audit file extracted to. The default is A.
- In ‘options’, ‘NOERASE’ keeps the Audit file on the VM:Secure server. If not specified, the Audit file is erased on the server once extracted.

For example, to extract the Object Rules Audit file to the file OBJ LOGFILE A, enter:

```
vmsecure objlog extract obj logfile a
```

Responses

When VM:Secure successfully extracts the Audit file, you will see:

```
Objects Audit file extracted to' tofn toft tofm
```

If VM:Secure cannot find the AUDIT file, you will see:

```
VMXSYS8003E The Audit file for OBJECTS does not exist.
```

OBJQUERY COMMAND

The OBJQUERY command allows users or administrators to query User Object Rules files that are loaded in the VM:Secure server. The command allows queries to show specific or wildcard searches and to show only those User files that have been referenced. It has the format:

```
OBJQUERY userid [ (options
```

where:

- ‘userid’ specifies the user-id(s) to query on. It can be specified as a specific user-id, as an asterisk to indicate all user-ids, or with a trailing asterisk to indicate a wildcard search.
- The option ‘used’ displays only the user-id Object Rules files found that have been referenced since they were last loaded into storage.

For example, to query the user-id FRANK’s Object Rules file, you enter ‘vmsecure objquery frank’. To query all user-ids that begin with FRA generically, you enter ‘vmsecure objquery fra*’.

Responses

If VM:Secure finds user-ids that satisfy the OBJQUERY, they will be displayed in the following way:

```
userid1 - nnnnn userid2 - nnnnn userid3 - nnnnn userid4 - nnnnn
userid5 - nnnnn userid6 - nnnnn userid7 - nnnnn userid8 - nnnnn
.... etc
```

Where ‘nnnnn’ is the number of times that the user-id file has been referenced since it was last loaded into storage.

If VM:Secure does not find a specific user-id the response will be:

```
VMXSYS8ØØ3E The Objects file for user-id does not exist.
```

If VM:Secure does not find user-ids for a generic request, the response will be:

```
VMXSYS8Ø23E No Object Files found loaded for 'user*'.
```

If VM:Secure does not find user-ids for a generic request with the USED option, the response will be:

```
VMXSYS8Ø24E No Object Files have been referenced for user-id
```

Return codes and error messages

OBJQUERY has no return codes or error messages.

OTHER COMMANDS

There are a few other Object Rule commands – used to control requests, load the files into storage, etc. These routines are not intended to be EXECuted by any users directly but are used as tools by the VM:Secure server or within other user commands.

- OBJDLOAD validates and loads Object Rule Definition files into storage. It is called from the OBJSTART macro.
- OBJEND is used to terminate Object Rules, but leave VM:Secure running.
- OBJLOAD validates and loads user Object Rules files. It can be used by an administrator to reload the user files if needed.
- OBJLOCK locks the use of Object Rules. It is normally used in OBJSTART and for disk maintenance, etc.
- OBJSETUP is used in the initial setting up of the Object Rules environment on the VM:Secure server. It accesses the Object Rules mini-disk and loads the default OBJECT SETTINGS definitions.
- OBJSTART calls OBJLOCK, OBJSETUP, OBJDLOAD, and OBJLOAD, and then OBJLOCK CLEAR. This macro is normally put in the SYSTEM VMSECURE macro to initialize the Object Rules environment.

MESSAGES AND CODES

This section provides a summary of messages and codes added by Object Rules. Each is followed by an explanation of the cause and a response, if appropriate.

```
7ØØØE  The OBJECT RULES are not active.
```

Cause: a request for Object Rules access or administration was used and the Object Rules system has not been initialized, or is locked for maintenance.

Response: check the VM:Secure server and make sure Object Rules are initializing correctly.

```
7001I   The Object Settings have been loaded.
```

Cause: initialization processing has loaded the Object Settings (disk address and file mode of the Object Rules disk and the default global access). No response is necessary.

```
7002I   The Object Rules have been locked/unlocked
```

Cause: Object Rules access has been locked/unlocked. No response is necessary.

```
8000I The OBJECT source data disk cuu has been
          accessed at mode mode
```

Cause: the Object Rules disk has been accessed for initialization. No response is necessary.

```
8001I   n% of nnnn files have been loaded.
```

Cause: OBJECTS files are being loaded into storage at initialization. This message is showing the progress. No response is necessary.

```
8002I   The object-part have been action for user-id
```

Cause: this is an Object Rule part. No response is necessary.

```
8003E   file-description file for user-id does not exist.
```

Cause: a file was not found for a particular request. This could be a user OBJECTS file or a definition file for an object.

Response: you should make sure your command information was correct. If it was, contact your VM:Secure administrator to find out what may have happened to the file.

```
8005E   Error rc from EXECLOAD of filename filetype filemode
```

Cause: while attempting to load a file into storage, EXECLOAD encountered an error.

Response: find HELP for EXECLOAD and see what the return code is. Correct the problem and re-initialize Object Rules.

```
8006E   Object name not specified.
```

Cause: the Object name was not specified on an OBJ command.

Response: supply a valid Object Name.

```
8007E   Object parameters not specified.
```

Cause: an OBJ command was issued that requires object parameters.

Response: supply valid object parameters for the command.

```
8009E   Duplicate Object Definition control word specified.
```

Cause: a duplicate control word was found while initializing an OBJDEF file.

Response: review the OBJDEF file that failed. Remove any duplicate entries and re-initialize Object Rules.

```
8012I   Objects not changed.
```

Cause: you have cancelled an Object Rules change command (OBJEDIT, etc). No response is necessary.

```
8013E   Load of object-name Objects Definitions failed.
```

Cause: the load of an OBJDEF file has failed.

Response: review all error messages and determine the cause of the failure. Correct and re-initialize Object Rules.

```
8014E   Token min|max length is missing or invalid.
```

Cause: an object's token that was defined with the 'min,max' format is invalid.

Response: review the minimum and maximum values for that token. Correct and re-initialize Object Rules.

```
8015E   Object Definition object-name is already active - no
             changes allowed.
```

Cause: you are attempting to OBJDLOAD an Object Definition file (OBJDEF) that is already loaded. Object Rules does not allow OBJDEF files to be re-loaded once they are already in storage.

Response: make sure you are trying to OBJDLOAD the correct OBJDEF file.

```
8016E   Object Definition control record out of order.
```

Cause: processing an Object Definition file (OBJDEF) found a record out of order.

Response: review the OBJDEF file that the error occurred on. Read the Object Rules Syntax section of this article to find the rules on the correct order of records for OBJDEF files.

```
     8017E   Index exceeds defined number of tokens for object.
```

Cause: a ‘Token.x’ or ‘Default.x record’ was found where ‘x’ exceeded the number of tokens defined on the ‘Tokens’ record.

Response: either increase the Tokens or remove the extra ‘Token.x’ or ‘Default.x’ record.

```
     8018E   Missing definition for token #nn
```

Cause: a Token.x record was not found in the Object Definition file (OBJDEF) for that token number.

Response: adjust the number of tokens for that object or add the missing Token.x entry.

```
     8019E Length of token word|default #nnn is more|less than the
                defined min|max of nn
```

Cause: a token word in an Object Rule file (OBJECTS) or Default.x value in an Object Definition file (OBJDEF) exceeds the character limits for that object’s token.

Response: correct the value of the token.

```
     8020E   Value for token word|default #nn does not match any
               of the allowed words.
```

Cause: a token word in an Object Rule file (OBJECTS) or Default.x value in an Object Definition file (OBJDEF) was not found in the allowed word list for that object’s token.

Response: correct the value of the token.

```
     8021E   Objects file already exists for user-id
```

Cause: an Object Rule file already exists for a user when attempting to use the OBJADD command.

Response: make sure you are specifying the correct user-id on the command.

```
     8023E   No Object Files found loaded for user-id
```

Cause: the user-id specified on the OBJQUERY command was not found.

Response: make sure you are specifying the correct user-id on the command.

```
    8Ø24E   No Object Files have been referenced for user-id(s)
```

Cause: the OBJQUERY command found that no Object Rules files have been referenced for the user-id(s) you were checking. No response is necessary.

```
    82ØØE   Object object-name does not exit.
```

Cause: you have specified an object that has not been defined.

Response: either correct the object name being used, or check that the Object Definition file has been loaded, or create a new one.

```
    82Ø1E   Tokens invalid or missing for object object-name.
```

Cause: a request for access to an object found that the tokens specified are invalid or missing.

Response: review the access request (OBJCHK or OBJFOR) for the object and correct the tokens on the command.

```
    82Ø2E   Severe error rc reading 'file-id' from storage.
```

Cause: when attempting to read a file from storage using CMS Pipelines, a severe error occurred.

Response: check the return code from the CMS PIPE command to find the reason for the failure.

```
    82Ø3E   Too many tokens specified for object object-name
                Maximum is nnn
```

Cause: too many tokens were specified for an object.

Response: correct the number of tokens specified for the object being used.

```
    82Ø4E   Missing token nnn (no default) for object object-name
```

Cause: token ‘nnn’ is missing for the object specified and there is no default defined for it.

Response: supply a valid token for that object.

```
    82Ø5E   Token #nnn is invalid for object object-name
```

Cause: the specified token is invalid for the object.

Response: correct the token value for that object.

```
    82Ø6E   Token count does not match for object object-name
              Tokens allowed is nnn
```

Cause: the number of tokens supplied for the object is incorrect.

Response: check the number of tokens that are required for the object and correct the request.

```
    9ØØ1E   Access rejected for: 'object-rule-request'
```

Cause: an access request for an object failed.

Response: verify that the rejection is valid. If access should be granted, contact your security administrator to have them add an ACCEPT rule for this Object Rule request.

AUTHORIZ CONFIG

```
*ED= 96/Ø4/Ø2 11:37:Ø6 VINCENJ  CONFIG   95/Ø8/Ø3

**** VM SYSTEMS PROGRAMMING AUTHORIZATIONS ****
LIST  *SYSPGMR VMSECUR2 VMX2ADM
GRANT *        TO *SYSPGMR

**** OBJECT AUTHS/COMMANDS ****
LIST  *ADMIN   VMX2ADM
LIST  *OBJCMDS $OBJEDIT $OBJLOAD $OBJADD $OBJDEL $OBJQUERY $OBJFOR,
               $OBJLOG
GRANT *OBJCMDS OVER *ALL TO *ADMIN

**** GENERAL AUTHORIZATION ****
GRANT $OBJCHK  OVER *SELF TO *ALL

**** WITHOLD COMMANDS ****
LIST  *DONTUSE ADDMDISK ASSIGN CAN CHANGE CHGMDISK CLASS,
               COMPRESS DELENTRY DELMDISK DISPLNK EDIT EDX,
               EXPIRE EXTRACT FEN* GENACI GENHS GENINCL GENENTRY,
               GROUP HISTORY IPLDISKX JOURNAL LOCK LOGMSG MAINT,
               MANAGE MAP MDSKSCAN MULTIPLE NEWIPL NOLOG OVERRIDE,
               PASSWORD QRULES RECLAIM REPENTRY RESET RULEMAP RULES,
               SYSWORD TRANSFER USER
WITHHOLD *DONTUSE FROM *ALL
```

DASD CONFIG

```
*ED= 95/Ø8/Ø3 21:45:27 VMSECUR2 CFGUPGRD 95/Ø8/Ø3
```

```
SUBPOOL BOGUS LARGEGAP LOWEND *
```

PRODUCT CONFIG

```
*ED= 95/Ø8/Ø3 21:45:27 VMSECUR2 CFGUPGRD 95/Ø8/Ø3
DIRECT 1AØ NODIR
MACLOAD OBJ*
DUMP OPERATNS NOCLEAR
SYSOPER OPERATOR
MESSAGE MSG
MSGCASE LOWER

ACCESS DRCT 1BØ U
ACCESS BKUP 1B1 B
ACCESS HOLD 1B2 H
ACCESS AUDT 1DØ T
```

SECURITY CONFIG

```
*ED= 95/Ø8/Ø3 21:45:27 VMSECUR2 CFGUPGRD 95/Ø8/Ø3
```

MAKEDIR EXEC

```
/** Create the directory entries **/
Arg userid .
'ACCESS 1BØ U'

'PIPE(name MAKEDIR)|',
   '< USER TEMPLATE A |',
   'DROP FIRST 1 |',
   'STEM TEMPLATE.'

Say 'Creating' userid
'PIPE(name MAKEDIR)|',
   'LITERAL USER' userid 'NOPASS|',
   'APPEND < USER TEMPLATE A |',
   '>' userid 'SECUREVM U'
Exit
```

PROFILE EXEC

```
/* ============================================================ */
/*  VM Software component and system administrator profile EXEC. */
/*  You may modify this EXEC as long as it accesses the 1FF      */
/*  mini-disk (VMRMAINT's 192) as Z and issues the proper        */
/*  start-up command (VMISTART for service virtual machines,     */
/*  VMISYSAD for system administrators).                             */
/* ================================================================= */

address('COMMAND')

   'CP SET EMSG ON'
   'CP SET RUN  ON'
   'CP SET PFØ3 RETRIEVE'
   'CP SET PF15 RETRIEVE'
   'CP SET PFØ5 RETRIEVE'
   'CP SET PF17 RETRIEVE'
   'CP SPOOL PRT TO *'
/* ---------------------------------------------------------------- */
/*  Start a console spool file class A spooled to the reader.       */
/* ---------------------------------------------------------------- */
   'CP SPOOL CONSOLE STOP CLOSE'
   'CP SPOOL CONSOLE START * CLASS A'

/* ---------------------------------------------------------------- */
/*  Access the VMRMAINT common disk as filemode Z.                  */
/* ---------------------------------------------------------------- */
   'ACCESS 1FF Z'
    accrc = rc
    if accrc ¬= Ø then do
        say ' '
        say 'The VMRMAINT user-id''s 192 mini-disk could not be'
        say 'accessed.  This disk must be linked RR at virtual'
        say 'address 1FF in' userid()||'''s directory entry.'
        say ' '
        say 'The PROFILE EXEC did not complete successfully.'
        exit accrc
        end
If Substr(Diag(24,-1),13,1)=2 Then Do
  Say
  Say'Start VMSECURE W/Object Rules?  (Y|N)'
  Parse Upper External ans .
  If ans ¬= '' & ans ¬= 'Y' Then Exit
  End
'PIPE(name PROFILE)|',
   '<' Userid() 'MDISKS * |',
   'FIND VMSI|',
   'SPECS W 3 1 |',
   'VAR PRODSFW'
'ACCESS' prodsfw 'D'
'PIPE(name PROFILE)|',
   '< VMSECURE MESSAGES D |',
   'APPEND < VMSECURE NEWMSG A |',
   '> VMSECURE MESSAGES A'
/* ---------------------------------------------------------------- */
/*  Invoke the verification and start-up EXEC                       */
/* ------------------------------------------------------------ */
'EXEC VMISTART VM:SECURE (NOPROMPT'
```

VMSECURE MANAGERS

```
MANAGER VMSECUR2 * BOGUS
SKELETON VMSECUR2 GEN1
DEVTYPE VMSECUR2 339Ø
```

VMSECURE NEWMSG

```
7ØØØE The OBJECT RULES are not active.
7ØØ1I The Object Settings have been loaded.
7ØØ2I The Object Rules have been ........
8ØØØI The OBJECT source data disk .... has been accessed at mode ..
8ØØ1I ...% of ..... files have been loaded.
8ØØ2I The ........ ........... have been ......... for ........
8ØØ3E .... .......... file for ........ does not exist.
8ØØ4E You cannot ERASE all .......... files.
8ØØ5E Error ........ from EXECLOAD of ........ ........ ..
8ØØ6E Object name not specified.
8ØØ7E Object parameters not specified.
8ØØ8E Invalid ............. '__________________________________________'
8ØØ9E Duplicate Object Definition control word specified.
8Ø1ØE Validation error loading ........ ........ ..
8Ø11E Record:
_________________________________________________________________________
8Ø12I Objects not changed.
8Ø13E Load of ........ Objects Definitions failed.
8Ø14E Token ........ length is missing or invalid.
8Ø15E Object Definition ........ is already active - no changes allowed.
8Ø16E Object Definition control record out of order.
8Ø17E Index exceeds defined number of tokens for object.
8Ø18E Missing definition for token #....
8Ø19E Length of token ...... #.... is ...... than the defined ... of ..
8Ø2ØE Value for token ...... #.... does not match any allowed words.
8Ø21E Objects file already exists for ........
8Ø22I Defined:
_________________________________________________________________________
8Ø23E No Object Files found loaded for ........
8Ø24E No Object Files have been referenced for ........
82ØØE Object ........ does not exit.
82Ø1E Tokens invalid or missing for object '________'.
82Ø2E Severe error ....... reading '___________________' from storage.
82Ø3E Too many tokens specified for object ........  Maximum is ....
82Ø4E Missing token .... (no default) for object ........
82Ø5E Token #.... is invalid for object ........
82Ø6E Token count does not match for object ..... Tokens allowed is ...
```

```
9ØØ1E Access rejected for:
'________________________________________________________________'
```

GEN1 SKELETON

```
USER GEN1 GENERAL 4M 4M G
*PW=
*ED= 92/Ø2/24 Ø8:55:24 VINCENJ  ADMIN
* GENERAL USER
ACCOUNT ØØØØØØ NOWHERE
IPL CMS
CONSOLE ØØ9 327Ø Ø
SPOOL ØØC 254Ø READER A
SPOOL ØØD 254Ø PUNCH A
SPOOL ØØE 14Ø3 A
```

OBJECT TEMPLATE

```
* Object Rules
*
* The general Object Rule statement format is:
*
*    ACCEPT object token1 token2 ...
*    REJECT object token1 token2 ...
*
* Example
*
*    ACCEPT CAR DRIVE HIGHWAY
*    REJECT CAR PARK HIGHWAY
*
```

USER TEMPLATE

```
*ED= 94/Ø5/27 Ø7:55:Ø5 VINCENJ  EDITNA   94/Ø5/27
*1 - ØØØØØØ NOT_A_USER
ACCOUNT Ø2Ø77Ø 3Ø8-Ø3
CONSOLE ØØ9 327Ø Ø
SPOOL ØØC 254Ø READER A
SPOOL ØØD 254Ø PUNCH A
SPOOL ØØE 14Ø3 V
```

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

James S Vincent
Software Specialist
Nationwide Insurance (USA)

Adding new functions to XEDIT

Continuing the Mouse on the mainframe *series of articles on the manipulation of System/390 applications with a PC or workstation mouse, the author examines adding new functions to XEDIT with alternative XEDIT customization macros.*

INTRODUCTION

Previous articles in this series have discussed adding mouse-clickable reserved lines and pop-up menus of subcommands to XEDIT screens; the HOTKEYS and KEYWIN XEDIT macros were presented and discussed at some length. PETs or 'Pointer Enabled Tools' increase productivity by bringing the convenience of mouse-control to XEDIT.

This article continues the discussion about augmenting XEDIT with mouse-clickable software. These programs are written in REXX and rely on XEDIT subcommands, virtual screens, and CMS windows. Included are a generalizable macro that can be used to define new text management functions, and a template that can be used to quickly and easily create alternative XEDIT customization macros.

DEFINING NEW TEXT MANIPULATION FUNCTIONS

Standard XEDIT functions, such as SPLTJOIN, are extremely valuable in manipulating file text within XEDIT. Assuming SPLTJOIN is assigned to PF11, one positions the 3270 cursor on a line of text and presses PF11. If the target line is blank from the cursor position to the end of the line (to the right), then the next line is appended to the target line to form a longer line of text. If one or more non-blank characters exist at or to the right of the cursor, then the target line is split into two shorter lines. SPLTJOIN is a function that is sensitive to the location of the 3270 cursor when the PF key is pressed.

Other interesting text manipulation functions can be developed using REXX and standard XEDIT macro coding techniques. For example, a function called 'SLIDE' might be developed which moves the text on a line such that the first non-blank character on the line is positioned under the cursor. Figure 1 offers a 'before' and 'after' view

Before:

```
===== * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== This line is to be moved to the right.
===== This line can stay where it is.
===== * * * End of File * * *
```

After:

```
===== * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
=====          This line is to be moved to the right.
===== This line can stay where it is.
===== * * * End of File * * *
```

Figure 1: Before and after invocation of the SLIDE function

of a line of text that has been changed by SLIDE. In this case, the cursor is set on column 10 of line 1, and then a PF key is pressed. (SLIDE is assigned to the PF key.)

Text can be moved to the right or to the left, a line at a time. The SLIDE function is useful in arranging statements in a program or other text.

The XFUN XEDIT macro implements a number of text manipulation functions, including SLIDE. Programmers can add additional functions into XFUN by coding new cases within the SELECT statement. XFUN makes extensive use of XEDIT variables, which provide cursor location and other information about the XEDIT session. Help for XEDIT variables is found by entering ‘HELP XEDIT EXTRACT’ on the CMS or XEDIT command lines.

XFUN functions can be assigned to PF keys within a ‘PROFILE XEDIT’ or other macro, with a command similar to the following:

```
'SET PF9 MACRO XFUN SLIDE'
```

where:

- ‘PF9’ is the PF key assigned to the function.
- ‘MACRO’ signifies an XEDIT macro.
- ‘XFUN’ is the name of the macro.
- ‘SLIDE’ is the requested function.

THE XFUN XEDIT MACRO

```
/* XFUN XEDIT extending XEDIT with new functions                     */
/*********************************************************************/
/* Functions:                                                        */
/*                                                                   */
/* XFUN ADDSTAY  - adds a line under the cursor, cursor stationary   */
/* XFUN INSDATE  - inserts the date based on the cursor position     */
/* XFUN MOVEVIEW - repositions the file left or right based on cursor */
/* XFUN PRINTFIL - prints current file from memory                   */
/* XFUN SETCURL  - repositions the file based on cursor position     */
/* XFUN SLIDE    - repositions line text to cursor position          */
/* XFUN TABEOL   - tabs to end of the pointed to line, or the next   */
/*********************************************************************/

/*********************************************************************/
/* Retrieve the argument, extract some XEDIT variables, set some     */
/* constants, and turn XEDIT messages off.                           */
/*********************************************************************/
Parse Upper Arg function .
'EXT/CURS/FN/FL/LINE/MSGM/SIZ/SPILL/TRUNC/VERS/'
functions='ADDSTAY INSDATE MOVEVIEW PRINTFIL SETCURL SLIDE TABEOL'
textfuncs='INSDATE SETCURL SLIDE TABEOL'
comment=''
oldmsgm=msgmode.1
oldspill=spill.1
'MSGMODE OFF'
/*********************************************************************/
/* Abort further processing under certain conditions:                */
/*                                                                   */
/* 1  If the requested function is not supported                     */
/*                                                                   */
/* 2  If the file is empty and ADDSTAY was not requested             */
/*                                                                   */
/* 3  If a 'text function' was requested but the cursor is not in the */
/*     text area (not in the file)                                   */
/*********************************************************************/
Select;
   When (Wordpos(function,functions)=Ø)
      Then comment='Unknown request.'
   When (size.1=Ø & function ¬= 'ADDSTAY')
      Then comment='File is empty.'
   When (Wordpos(function,textfuncs)>Ø &,
        (cursor.3<1 | cursor.3>size.1 | cursor.4<1))
      Then
         Do
            comment='Cursor must be placed in the text area.'
            ':'line.1
            End
   Otherwise Signal PROCESS
   End
Signal MACROEND
PROCESS:
/**********************************************************************/
/* If the file is not empty, reposition the file according to cursor  */
/* location if necessary, and extract the text of the (then) current  */
/* line.                                                              */
/**********************************************************************/
If (size.1>Ø)
   Then
      Do
         If cursor.3 ¬= line.1 & cursor.3 > Ø & cursor.3 < size.1+1
            Then ':'cursor.3
         'EXT/CURL/'
         oldline=Strip(curline.3)
         End

/**********************************************************************/
/* Process the requested function.                                    */
/**********************************************************************/
Select;
   When (function='ADDSTAY') /* add a line, retain cursor position */
      Then
         Do
            'ADD 1'
            ':'line.1
            If (cursor.3 < 1 | cursor.3 > size.1)
               Then lineno=line.1
               Else lineno=cursor.3
            'CURSOR FILE' 1+lineno Max(1,cursor.4)
            End
   When (function='INSDATE') /* insert the current date */
      Then
         Do
            ':'cursor.3
            'EXT /CURL/SPILL/'
            'SET SPILL ON'
            parta=Substr(curline.3,1,(cursor.4)-1)
            partb=Substr(curline.3,cursor.4)
            'R' parta||Date('N')||partb
            'SET SPILL' spill.1
            ':'line.1
            'CURSOR FILE' cursor.3 cursor.4 'PRIORITY 3Ø'
            End
   When (function='MOVEVIEW') /* reposition the file left or right */
      Then
         Do
            If (cursor.4 < 1)
               Then "Left" Max(vershift.1,-vershift.1)
               Else "Right" cursor.4-vershift.1-1
            ':'line.1
            End
   When (function='PRINTFIL') /* print the file from memory */
      Then
         Do
            Address 'COMMAND' 'PIPE CP Q V PRT | TAKE 1' ,
               '| SPECS WORDS 5 1 | VAR SPCONT'
            'TOP'
            'NEXT'
            'STACK *'
            Address 'COMMAND' 'PIPE STACK | SPECS PAD 40 1-* 2' ,
               '| ASATOMC | PRINTMC'
            pc=rc
            If (pc=0)
               Then
                  If (spcont='CONT')
                     Then comment='File spooled for printing.'
                     Else
                        Do
                           Address 'COMMAND' 'CP CLOSE PRINTER'
                           comment='File printed; printer closed.'
                           End
               Else comment='Pipe error' pc 'printing current file.'
            ':'line.1
            End
   When (function='SETCURL') /* reposition the file up or down */
      Then
         Do
            ':'cursor.3
            comment='File repositioned.'
            End
   When (function='SLIDE')    /* reposition text left or right */
      Then
         Do
            If (Length(oldline)=0)
               Then comment='Line is empty.'
               Else
                  Do
                     'DELETE 1'
                     newlnumb=-1+cursor.3
                     newl=Left(' ',-1+cursor.4)Strip(curline.3)
                     lnewl=Length(newl)
                     If (lnewl>trunc.1)
                        Then
```

```
                           Do
                              ':'newlnumb 'I 'Substr(newl,1+trunc.1)
                              ':'newlnumb 'I 'Substr(newl,1,trunc.1)
                              comment='Text spilled to next line.'
                              End
                        Else ':'newlnumb 'I 'newl
                     End
            ':'line.1
            End
   When (function='TABEOL') /* reposition cursor eol or next line */
      Then
         Do
            lcurline=Length(Strip(curline.3,'T'))
            If (1+lcurline=cursor.4)
               Then
                  If (cursor.3=size.1)
                     Then
                        Do
                           comment='End of file reached.'
                           ':'line.1
                           'CURSOR CMDLINE'
                           End
                     Else
                        Do
                           'NEXT'
                           'EXT/CURL/'
                           newcol=Length(Strip(curline.3,'T'))+1
                           ':'line.1
                           'EXT/FL/'
                           If (flscreen.2<1+cursor.3)
                              Then
                                 Do
                                    'NEXT'
                                    comment='File repositioned.'
                                    End
                           'CURSOR FILE' 1+cursor.3 newcol
                           End
               Else
                  Do
                     ':'line.1
                     'CURSOR FILE' cursor.3 1+lcurline
                     End
            End
   Otherwise NOP
   End
MACROEND:

/***********************************************************************/
/* Turn message mode on, issue a message, restore msgmode and spill.  */
/***********************************************************************/
'MSGMODE ON'
If (Length(comment)>0) Then 'MSG' 'XFUN:' comment
```

```
'MSGMODE' oldmsgm
'SPILL' oldspill
Exit(Ø)
```

CREATING ALTERNATIVE XEDIT CUSTOMIZATION MACROS

The file ‘XPROFILE TEMPLATE’ provides a good basis for defining alternative XEDIT customization macros. The template should be copied to another file prior to modification. For example, to create a new macro called ‘PROFA XEDIT’, one would issue the following command:

```
    COPYFILE XPROFILE TEMPLATE A PROFA XEDIT A
```

Macro ‘PROFA XEDIT’ can then be changed as desired. The colour used to display the reserved line help text can be selected, PF keys can be labelled and assigned, and any other XEDIT customization subcommands can be added in the appropriate places, as indicated by the comments within the template.

Please note that macros created from ‘XPROFILE TEMPLATE’ redefine the ENTER key to invoke the HOTKEYS macro (discussed in *VM Update*, Issue 151, March 1999) whenever the ENTER key is pressed. HOTKEYS turns the XEDIT reserved line help text into a series of ‘hotspots’, which can be mouse-clicked using properly configured 3270 terminal emulation software.

THE XPROFILE TEMPLATE FILE

```
/*******************************************************************/
/* XEDIT Profile Template.  Make changes, save as 'filename XEDIT' .  */
/*******************************************************************/

/* Set reserved line PF key help text colour.                        */
c = 'T'                                   /* Options: B D G P R T W Y */
/* Enter specific XEDIT session tailoring commands below.            */
'SET CASE MIXED IGNORE'                                /* for example */
/* Assign PF key functions and labels below. Limit label text to nine */
/* characters.                                                        */
pf1function  = 'HELP         '
pf1label     = 'Help         '
pf2function  = 'SOS LINEADD  '
pf2label     = 'LineAdd      '
pf3function  = 'QUIT         '
pf3label     = 'Quit         '
Pf4function  = '             '
```

```
pf4label     = '             '
pf5function  = '             '
pf5label     = '             '
pf6function  = '             '
pf6label     = '             '
pf7function  = 'BACKWARD     '
pf7label     = 'Backward     '
pf8function  = 'FORWARD      '
pf8label     = 'Forward      '
pf9function  = '             '
pf9label     = '             '
PF1Øfunction = '             '
PF1Ølabel    = '             '
Pf11function = 'SPLTJOIN     '
pf11label    = 'SpltJoin     '
Pf12function = 'CURSOR HOME  '
pf12label    = 'Cursor       '

/* Ensure the COMMAND LINE is on; enable XEDIT for mouse clicks;      */
/* set the PF KEYS as defined above; set RESERVED LINES.              */

'CMDLINE ON'
'ENTER BEFORE MACRO HOTKEYS'
'PF1'  pf1function; 'PF2'  pf2function; 'PF3'  pf3function
'PF4'  pf4function; 'PF5'  pf5function; 'PF6'  pf6function
'PF7'  pf7function; 'PF8'  pf8function; 'PF9'  pf9function
'PF1Ø' pf1Øfunction;'PF11' pf11function;'PF12' pf12function
'RESERVE -4' c 'N  P',
  '1='Left(pf1label,1Ø) '2='Left(pf2label,1Ø) '3='Left(pf3label,1Ø),
  '4='Left(pf4label,1Ø) '5='Left(pf5label,1Ø) '6='Left(pf6label,1Ø)
'RESERVE -3' c 'N  F',
  '7='Left(pf7label,1Ø) '8='Left(pf8label,1Ø) '9='Left(pf9label,9),
  '1Ø='Left(pf1Ølabel,9) '11='Left(pf11label,9) '12='Left(pf12label,9)
Exit(Ø)
```

In a first example, the macro ‘PROFA XEDIT’ assigns various XFUN functions to several of the PF keys. One might invoke ‘PROFA XEDIT’ to customize an XEDIT session, by issuing the following command on the CMS command line:

```
     X TEST FILE (PROF PROFA
```

which bypasses the standard ‘PROFILE XEDIT’ macro and invokes ‘PROFA XEDIT’ instead.

THE PROFA XEDIT MACRO

```
/*******************************************************************/
/* PROFA XEDIT: Sets PF keys to XFUN calls.                        */
/*******************************************************************/
```

```
/* Set reserved line PF key help text colour.                        */
c = 'T'                                  /* Options: B D G P R T W Y */
/* Enter specific XEDIT session tailoring commands below.            */
'SET CASE MIXED IGNORE'                               /* for example */
/* Assign PF key functions and labels below. Limit label text to nine */
/* characters.                                                       */
pf1function  = 'HELP         '
pf1label     = 'Help         '
pf2function  = 'MACRO XFUN ADDSTAY'
pf2label     = 'AddStay      '
pf3function  = 'QUIT         '
pf3label     = 'Quit         '
Pf4function  = 'MACRO XFUN TABEOL'
pf4label     = 'TabEOL       '
pf5function  = 'MACRO XFUN SETCURL'
pf5label     = 'SetCurl      '
pf6function  = 'MACRO XFUN PRINTFIL'
pf6label     = 'PrintFile    '
pf7function  = 'BACKWARD     '
pf7label     = 'Backward     '
pf8function  = 'FORWARD      '
pf8label     = 'Forward      '
pf9function  = 'MACRO XFUN SLIDE'
pf9label     = 'Slide        '
PF10function = 'BEFORE RGTLEFT'
PF10label    = 'RightLeft    '
Pf11function = 'SPLTJOIN     '
pf11label    = 'SplitJoin    '
Pf12function = 'CURSOR HOME  '
pf12label    = 'Cursor       '

/* Ensure the COMMAND LINE is on; enable XEDIT for mouse clicks;     */
/* set the PF KEYS as defined above; set RESERVED LINES.             */
'CMDLINE ON'
'ENTER BEFORE MACRO HOTKEYS'
'PF1'  pf1function; 'PF2'  pf2function; 'PF3'  pf3function
'PF4'  pf4function; 'PF5'  pf5function; 'PF6'  pf6function
'PF7'  pf7function; 'PF8'  pf8function; 'PF9'  pf9function
'PF10' pf10function;'PF11' pf11function;'PF12' pf12function
'RESERVE -4' c 'N  P',
  '1='Left(pf1label,10) '2='Left(pf2label,10) '3='Left(pf3label,10),
  '4='Left(pf4label,10) '5='Left(pf5label,10) '6='Left(pf6label,10)
'RESERVE -3' c 'N  F',
  '7='Left(pf7label,10) '8='Left(pf8label,10) '9='Left(pf9label,9),
  '10='Left(pf10label,9) '11='Left(pf11label,9) '12='Left(pf12label,9)
Exit(0)
```

In a second example, the macro 'PROFB XEDIT' further customizes XEDIT by changing the colour of the XEDIT reserved line help text to yellow, and assigning KEYWIN menus to PF1 and PF12. (The

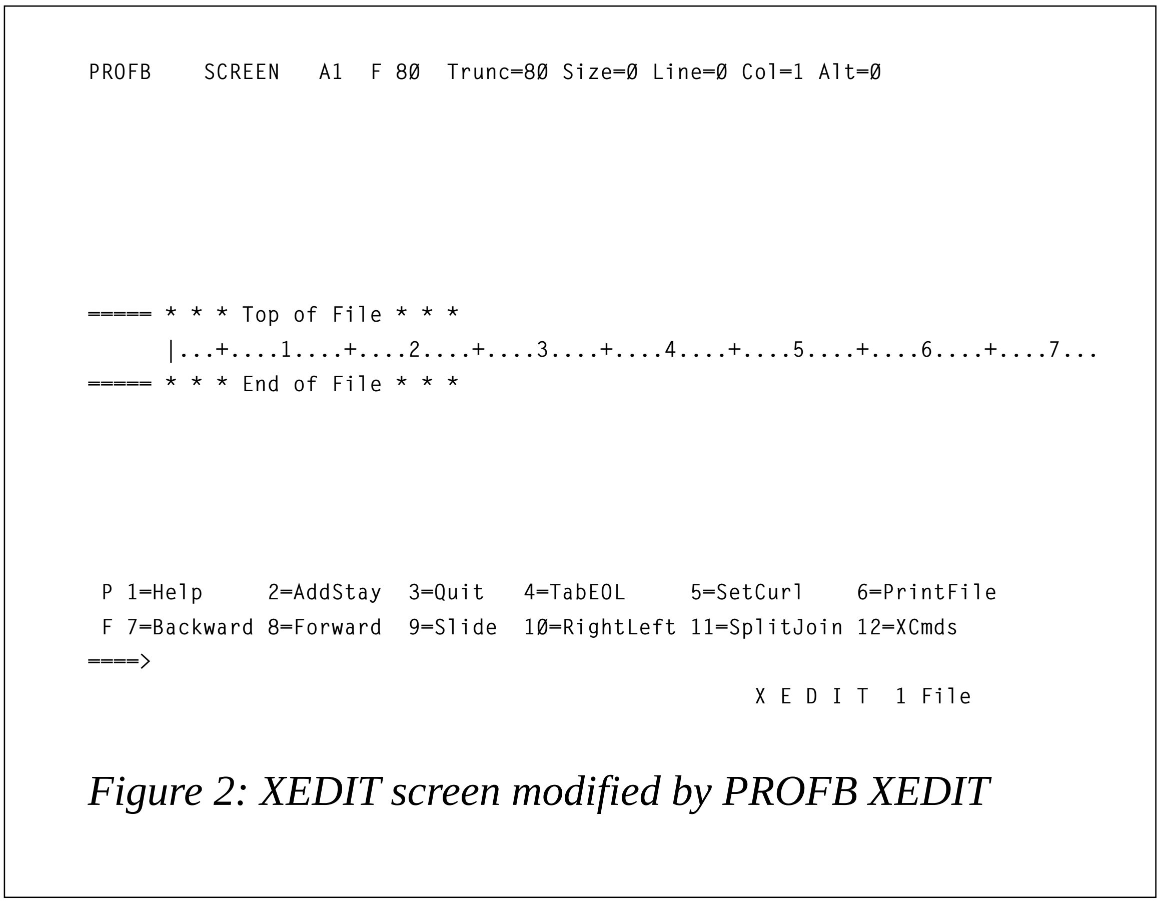
KEYWIN macro, which provides pop-up menus of commands within XEDIT, was discussed in *VM Update*, Issues 151 and 152, March and April 1999.)

THE PROFB XEDIT MACRO

```
/*********************************************************************/
/* PROFB XEDIT: Sets PF keys to KEYWIN and XFUN Macro Calls.         */
/*********************************************************************/

/* Set reserved line PF key help text colour.                        */
c = 'Y'                                  /* Options: B D G P R T W Y */
/* Enter specific XEDIT session tailoring commands below.            */
'SET CASE MIXED IGNORE'                               /* for example */
/* Assign PF key functions and labels below. Limit label text to nine */
/* characters.                                                        */
pf1function  = 'MACRO KEYWIN 1 HELP'
pf1label     = 'Help         '
pf2function  = 'MACRO XFUN ADDSTAY'
pf2label     = 'AddStay      '
pf3function  = 'QUIT         '
pf3label     = 'Quit         '
Pf4function  = 'MACRO XFUN TABEOL'
pf4label     = 'TabEOL       '
pf5function  = 'MACRO XFUN SETCURL'
pf5label     = 'SetCurl      '
pf6function  = 'MACRO XFUN PRINTFIL'
pf6label     = 'PrintFile    '
pf7function  = 'BACKWARD     '
pf7label     = 'Backward     '
pf8function  = 'FORWARD      '
pf8label     = 'Forward      '
pf9function  = 'MACRO XFUN SLIDE'
pf9label     = 'Slide        '
PF1Øfunction = 'BEFORE RGTLEFT'
PF1Ølabel    = 'RightLeft    '
Pf11function = 'SPLTJOIN     '
pf11label    = 'SplitJoin    '
Pf12function = 'MACRO KEYWIN 12 XCMDS'
pf12label    = 'XCmds        '

/* Ensure the COMMAND LINE is on; enable XEDIT for mouse clicks;      */
/* set the PF KEYS as defined above; set RESERVED LINES.              */
'CMDLINE ON'
'ENTER BEFORE MACRO HOTKEYS'
'PF1'  pf1function; 'PF2'  pf2function; 'PF3'  pf3function
'PF4'  pf4function; 'PF5'  pf5function; 'PF6'  pf6function
'PF7'  pf7function; 'PF8'  pf8function; 'PF9'  pf9function
'PF1Ø' pf1Øfunction;'PF11' pf11function;'PF12' pf12function
```



```
PROFB    SCREEN   A1  F 8Ø  Trunc=8Ø Size=Ø Line=Ø Col=1 Alt=Ø




===== * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== * * * End of File * * *



 P 1=Help     2=AddStay  3=Quit   4=TabEOL     5=SetCurl    6=PrintFile
 F 7=Backward 8=Forward  9=Slide  1Ø=RightLeft 11=SplitJoin 12=XCmds
====>
                                                    X E D I T  1 File
```

Figure 2: XEDIT screen modified by PROFB XEDIT

```
'RESERVE -4' c 'N  P',
  '1='Left(pf1label,1Ø) '2='Left(pf2label,1Ø) '3='Left(pf3label,1Ø),
  '4='Left(pf4label,1Ø) '5='Left(pf5label,1Ø) '6='Left(pf6label,1Ø)
'RESERVE -3' c 'N  F',
  '7='Left(pf7label,1Ø) '8='Left(pf8label,1Ø) '9='Left(pf9label,9),
  '1Ø='Left(pf1Ølabel,9) '11='Left(pf11label,9) '12='Left(pf12label,9)
Exit(Ø)
```

Invoking ‘PROFB XEDIT’ results in the XEDIT screen changes as shown in Figure 2.

SUMMARY

Several XEDIT customization techniques have been discussed in this and previous articles, including:

- Establishing reserved lines with PF key help text.
- Assigning alternative functions to PF keys.

- Enabling reserved line help text for mouse clicks.
- Adding mouse-clickable pop-up command menus.
- Automatically selecting initialization macros, depending on filetype.
- Creating new text manipulation functions.
- Creating alternative initialization macros from a standard template.

The combination of these techniques sets the stage for considering how filetype-specific development aids might be created.

FURTHER INFORMATION

Further information about the PETs project can be found at the following Web location: http://vm.uconn.edu/~pets/index.html.

Editor's note: in a future article, the author will explore one way of implementing filetype-specific development aids.

Richard G Ellis
Director of Computing and Information Systems
University of Connecticut (USA)

A full screen console interface – part 10

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

CSCSEV ASSEMBLE

```
         TITLE 'CSCSEV - CSC Sever IUCV connection with CSCUSR'
CSCSEV   START X'Ø15728'
         PRINT NOGEN
         CSCHDR                         Terminate user session
```

```
*
* Terminate user session
*
         USING IPARML,R9               IUCV Parameter List
         USING UIDSECT,R8              UID (user) Block
         SPACE
*
* Sever IUCV connection
*
*        Input R0 contains the IUCV PATHID (first two bytes)
*              R9 addresses the IUCV Parameter List
*       Output R8 points to the previous UID block or zero if not found
*
*
         LA    R8,SSSPTR               Scan active sessions
SEV100   LR    R2,R8                   Keep address of previous entry
         L     R8,UIDFWD               Address next entry
         LTR   R8,R8
         BZ    SEV400                  Not found, session never active
         CLM   R0,B'1100',UIDPID
         BNE   SEV100
         LR    R3,R0                   Save PATHID
         TM    UIDOPT1,UIDCONN         Is user connected?
         BO    SEV200                  Yes, don't de-allocate UID block
         L     R1,UIDFWD               Found
         ST    R1,0(,R2)               Alter chain pointer
         LA    R0,UIDSCRSZ             Release screen
         L     R1,UIDSCRN
         LINK  RELEASE
         LA    R0,UIDBUFSZ
         L     R1,UIDBUFF              Release User buffer
         LINK  RELEASE
SEV200   TM    UIDOPT1,UIDRMTE         Is user remote?
         BO    SEV800                  Yes, forget about IUCV
SEV300   ST    R3,IPPATHID             Copy to IUCV Parameter List
         MVI   IPFLAGS1,X'00'          Clear all flags
         LA    R0,7                *T* Create trace entry
         LINK  TRACE               *T*
         LA    R3,CSCNAME              Address
         CMSIUCV SEVER,NAME=(R3),PRMLIST=(R9)
         LTR   R15,R15                 Check for errors
         BZ    SEV700
         MSG   0018,RC                 Something happened
         B     SEV800
         SPACE
SEV400   LA    R8,UIDPTR               Scan pending sessions
SEV500   LR    R2,R8                   Keep address of previous entry
         L     R8,UIDFWD               Address next entry
         LTR   R8,R8
         BZ    SEV600                  Not found, display message
         CLM   RØ,B'11ØØ',UIDPID
         BNE   SEV5ØØ
         L     R1,UIDFWD              Found
         ST    R1,Ø(,R2)              Alter chain pointer
         LR    R3,RØ                  Save PATHID
         MSG   Ø15Ø                   Display cancelled message
         B     SEV3ØØ                 De-allocate and sever
         SPACE
SEV6ØØ   MSG   Ø151                   Active session not found
         B     SEV9ØØ
         SPACE
SEV7ØØ   MSG   Ø152                   Display info message
SEV8ØØ   TM    UIDOPT1,UIDCONN        Is user connected?
         BZ    SEV81Ø                 No...
         GO    CSCUSATC               Terminate connected sessions
         B     SEV82Ø
         SPACE
SEV81Ø   LA    RØ,UIDSIZE             De-allocate UID block
         LR    R1,R8
         LINK  RELEASE
SEV82Ø   LR    R8,R2
SEV9ØØ   BACK
         SPACE 3
         CSCDATA
         CSCDS (UID)
         PUSH  PRINT
         PRINT OFF
         COPY  IPARML
         POP   PRINT
         REGEQU
         END
```

It is now possible to regen CSCSVP and establish user sessions with limited functionality. Note that some commands may cause CSCSVP to abend. Create a configuration file named CSC CONFIG, with at least a USER and a PREFIX statements. Use the following sample as a guide:

```
USER    *       Classes Ø1 Ø2
PREFIX  R RSCS     Class 25    Blue
MESSAGE User *      High White  Underline                Locate <CSC *
MESSAGE User * hold High Alarm  Red RevVideo NoCase     Locate *Abend*
MESSAGE User * hold High Alarm Red Rev Route USERØØ1 Loc *CP entered;*

OPTIONS MSG
TITLE   Selected Title
DFRECS  16384
```

Use the SWAP command to change the fields in the output area; for example, SWAP DATE TIME CMS.

CSCWRP ASSEMBLE

```
         TITLE 'CSCWRP - CSC Process User SWITCH WRAP command'
CSCWRP   START X'Ø1A3B8'
         PRINT NOGEN
         CSCHDR                        SWITCH WRAP command
*
* Build partial lines for current screen
*
*
         USING UIDSECT,R8              UID (user) Block
         USING CCHSECT,R7              CCH (cache) Block
         SPACE
         TM    UIDOPT3,UIDCMS          Is CMS scroll active
         BO    WRP1ØØ
         GO    CSCWRPBT                No, adjust screen from bottom
         BAS   R14,MSGHOLD             Overlay messages on hold
         B     WRP9ØØ
         SPACE
WRP1ØØ   GO    CSCWRPTP                Adjust screen from top
         L     R7,UIDBUFF2             Address last line (current)
         SR    RØ,RØ
WRP2ØØ   C     RØ,CCHRECNO             Is a normal message?
         BNE   WRP3ØØ
         L     R7,CCHBWD               No, blank line, get previous
         LTR   R7,R7                   Anything left?
         BNZ   WRP2ØØ                  Yes, test it
         B     WRP9ØØ
         SPACE
WRP3ØØ   CLI   CCHLINE2,X'ØØ'          Can we display it?
         BE    WRP4ØØ                  No, try move up one line
         BAS   R14,GETLINES            Number of lines for this msg
         SR    RØ,RØ
         IC    RØ,CCHLINE1             First partial line on screen
         AR    RØ,R1
         IC    R1,CCHLINE2             Last partial line on screen
         CR    RØ,R1                   Is line truncated?
         BE    WRP9ØØ                  No, we are done
WRP4ØØ   LR    R4,R7                   Save address of new line
         L     R7,UIDBUFF1             Yes, let's move up one line
WRP41Ø   TM    CCHOPTS,CCHHOLD         Is it on hold?
         BZ    WRP5ØØ
         L     R7,CCHFWD               Yes, check next one
         LTR   R7,R7                   If there is another message
         BNZ   WRP41Ø
         B     WRP9ØØ                  All done
         SPACE
WRP5ØØ   CLI   CCHLINE2,X'ØØ'          Is this first message on screen
         BE    WRP9ØØ                  No, all screen on hold
         CR    R4,R7                   only one line left not on hold?
         BE    WRP9ØØ                  Yes, not much we can do
         L     RØ,CCHRECNO              Load record number
         C     RØ,UIDCMSTP              Already defined as top line?
         BE    WRP6ØØ                   Yes, almost clear the screen
         L     R4,CCHFWD                Delete first line and move
         LINK  DELETE                      everything up one line
         LINK  ADDBLKB                  Add a blank line at the bottom
         SR    RØ,RØ
         C     RØ,UIDCMSTP              Any CMS top line defined?
         BE    WRP1ØØ                   No, check screen again
         LR    R7,R4                    Yes, define new top line
         L     RØ,CCHRECNO              Load record number
         ST    RØ,UIDCMSTP              Save as new CMS top line
         B     WRP1ØØ                   Can we display the last message?
         SPACE
WRP6ØØ   L     RØ,CCHRECNO-CCHSECT(,R4) Simulate CLEAR function
         ST    RØ,UIDCMSTP              Define new CMS top line
WRP61Ø   L     R4,CCHFWD                Save address of following line
         LINK  DELETE                   Delete first line not on hold
         LINK  ADDBLKBT                 Add blank line at the bottom
         LR    R7,R4                    Address following line again
WRP62Ø   L     RØ,CCHRECNO              Is it the new expected top line?
         C     RØ,UIDCMSTP
         BE    WRP1ØØ                   Yes, check screen again
         TM    CCHOPTS,CCHHOLD          Is line on hold?
         BZ    WRP61Ø                   No, delete line
         L     R7,CCHFWD                Yes, skip it
         B     WRP62Ø                   Check all lines
         SPACE
WRP9ØØ   BACK
         SPACE 3
*
* Adjust screen from bottom
*
CSCWRPBT RELOC                          BOTTOM
         BAS   R14,GETSIZE
         L     R7,UIDBUFF2              Address last line
         SR    R4,R4                    Required by next IC
         IC    R4,UIDSCRL               Number of screen's detail lines
BOT1ØØ   STC   R4,CCHLINE2              Last detail line for this msg
         BAS   R14,GETLINES             Number of lines - 1 for this msg
         SR    R4,R1
         STC   R4,CCHLINE1              First detail line for this msg
         BCTR  R4,Ø                     Decrement one
         LTR   R4,R4                    Still a valid line? (> Ø)
         BNP   BOT2ØØ
         L     R7,CCHBWD                Yes, address previous message
         B     BOT1ØØ
         SPACE
BOT2ØØ   C     R7,UIDBUFF1              Is it the first line?
         BE    BOT9ØØ                   Yes, all messages on screen
         SR    R4,R4                     No...
BOT3ØØ   L     R7,CCHBWD                    make sure all previous lines
         STC   R4,CCHLINE1                  are not displayed
         STC   R4,CCHLINE2
         C     R7,UIDBUFF1
         BNE   BOT3ØØ
*        B     BOT9ØØ
         SPACE
BOT9ØØ   BACK
         SPACE 3
*
* Adjust screen from Top
*
CSCWRPTP RELOC                           TOP
         BAS   R14,GETSIZE
         L     R7,UIDBUFF1               Address first line
         SR    R4,R4                     Required by next IC
         IC    R4,UIDSCRL                Number of screen's detail lines
         LA    R3,1                      Start with line one
TOP1ØØ   STC   R3,CCHLINE1               First detail line for this msg
         BAS   R14,GETLINES              Number of lines - 1 for this msg
         AR    R3,R1
         STC   R3,CCHLINE2               Last detail line for this msg
         LA    R3,1(,R3)                 Increment one
         CR    R3,R4                     Space left on physical screen?
         BH    TOP2ØØ
         L     R7,CCHFWD                 Yes, address next message
         B     TOP1ØØ
         SPACE
TOP2ØØ   STC   R4,CCHLINE2               Truncate last msg if necessary
         C     R7,UIDBUFF2               Is it the last line?
         BE    TOP9ØØ                    Yes, all messages displayed
         SR    R4,R4                     No...
TOP3ØØ   L     R7,CCHFWD                    make sure all following lines
         STC   R4,CCHLINE1                  are not displayed
         STC   R4,CCHLINE2
         C     R7,UIDBUFF2
         BNE   TOP3ØØ
*        B     TOP9ØØ
         SPACE
TOP9ØØ   BACK
         SPACE 3
*
* Locate TOP line on user's screen
*
*       Output R7 addresses top line
*
CSCWRPGT RELOC                           Locate TOP line
         L     R7,UIDBUFF1               It could be the first one
GTOP1ØØ  CLI   CCHLINE2,X'ØØ'            Is it displayed?
```

```
         BNE   GTOP9ØØ                 Yes, done
         L     R7,CCHFWD               No, try next one
         C     R7,UIDBUFF2
         BNE   GTOP1ØØ
         L     R7,UIDBUFF1             WRAP is not on, use first line
GTOP9ØØ  BACK
         SPACE 3
*
* Locate BOTTOM line on user's screen
*
*       Output R7 addresses bottom line
*
CSCWRPGB RELOC                         Locate BOTTOM line
         L     R7,UIDBUFF2             It could be the last one
GBOT1ØØ  CLI   CCHLINE2,X'ØØ'          Is it displayed?
         BNE   GBOT9ØØ                 Yes, done
         L     R7,CCHBWD               No, try the previous one
         C     R7,UIDBUFF1
         BNE   GBOT1ØØ
         L     R7,UIDBUFF2             WRAP is not on, use last line
GBOT9ØØ  BACK
         SPACE 3
*
* Return number of display columns for message text
*
CSCWRPGS RELOC                         External call to GETSIZE
         BAS   R14,GETSIZE
         BACK
         SPACE
GETSIZE  EQU   *                       Get columns to display message
         LA    R5,5Ø                   This is the minimum
         TM    UIDOPT2,UIDDATE         If DATE is not displayed
         BO    GETS1ØØ
         LA    R5,9(,R5)               Add a few more columns
GETS1ØØ  TM    UIDOPT2,UIDTIME         And if TIME is not displayed...
         BO    GETS2ØØ
         LA    R5,9(,R5)
GETS2ØØ  TM    UIDOPT2,UIDUSER
         BO    GETS3ØØ
         LA    R5,9(,R5)
GETS3ØØ  BR    R14
         SPACE 3
*
* Get number of lines required for current message
*
*       Output R1 contains number of lines minus one
*
GETLINES EQU   *                       Screen lines to display msg
         SR    RØ,RØ                   Required by next IC
         IC    RØ,CCHRLEN              Get message length
         SR    R1,R1                     Zero counter
GETL1ØØ  LA    R1,1(,R1)                 Increment
         SR    RØ,R5                     Count required lines
         BP    GETL1ØØ
         BCTR  R1,Ø                      Remember we expect lines - 1
         BR    R14
         SPACE 3
*
* Overlay messages with the hold attribute
*
*
MSGHOLD  EQU   *                         Overlay messages on hold
         ST    R14,MSGHSV14
         BAS   R14,GETSIZE               Get number of display columns
         SR    R3,R3
         IC    R3,UIDSCRL                Number of screen lines
         SR    R4,R4                     Zero line pointer
         L     R7,UIDBUFF1               Check first message
MSGH1ØØ  TM    CCHOPTS,CCHHOLD           Is it on hold?
         BZ    MSGH8ØØ                   No, check another one
         SR    RØ,RØ
         IC    RØ,CCHLINE1               Get first screen line
         CR    RØ,R4                     Is it already displayed?
         BH    MSGH9ØØ                   Yes, all done
         LA    R4,1(,R4)                 No, next available screen line
         CR    R4,R3                     Is it valid
         BH    MSGH4ØØ                   No, screen full
         STC   R4,CCHLINE1               Position message on screen
         BAS   R14,GETLINES              Number of lines for this msg
         AR    R4,R1                     Possible last line for this msg
         CR    R4,R3                     Is it still valid?
         BH    MSGH5ØØ                   No, truncate message
         STC   R4,CCHLINE2               Yes, position message
MSGH8ØØ  L     R7,CCHFWD                 Address following message
         LTR   R7,R7                     Anything left?
         BNZ   MSGH1ØØ                   Yes, process it
         B     MSGH9ØØ
         SPACE
MSGH4ØØ  SR    R3,R3                     Screen full
         STC   R3,CCHLINE1               Do not display msgs following
MSGH5ØØ  STC   R3,CCHLINE2
         SR    R3,R3
MSGH6ØØ  L     R7,CCHFWD                 No space for these messages
         LTR   R7,R7
         BZ    MSGH9ØØ
         STC   R3,CCHLINE1
         STC   R3,CCHLINE2
         B     MSGH6ØØ
         SPACE
MSGH9ØØ  L     R14,MSGHSV14
```

```
         BR    R14
         SPACE 3
MSGHSV14 DS    F                        Save R14 for MSGHOLD
         SPACE 3
         CSCDATA
         CSCDS (UID,CCH)
         REGEQU
         END
```

Add class 03 to the USER statements of the configuration file and regen CSCSVP to make the browse PF keys functional.

CSCUSB ASSEMBLE

This module adds support for the UP and DOWN commands.

```
         TITLE 'CSCUSB - CSC Process User commands (browse)'
CSCUSB   START X'Ø19B6Ø'
         PRINT NOGEN
         CSCHDR                         User browse commands
*
* Process User browse commands
         USING UIDSECT,R8               UID (user) Block
         USING CCHSECT,R7               CCH (cache) Block
         SPACE
*
* Return to caller
RETURN   BACK
         SPACE 3
*
* Process UP command
CSCUSBUP RELOC                          UP command
         SR    RØ,RØ                    No table to search
         GO    CSCSCN                   Scan parameter
         LA    R2,1                     Default is 1
         BNZ   UPC1ØØ                   Nothing found, use default
         GO    CSCSCNVN                 Validate parameter
         BNZ   UPC5ØØ                   Not numeric, that's an error
         SR    RØ,RØ                    Do not search any table
         GO    CSCSCN                   Anything left?
         BZ    UPC6ØØ                   Yes, bad news, only one parm
UPC1ØØ   LTR   R6,R2                    Copy repetition factor
         BZ    UPC9ØØ                   It is zero, all done
         GO    CSCWRPGT                 Locate first line on screen
         C     R7,UIDBUFF1              Same as first line in buffer
         BE    UPC2ØØ                   Yes, half done
         GO    CSCUSCTL
         L     R7,UIDBUFF1              Address new top line
UPC2ØØ   SR    RØ,RØ
```

```
         C     RØ,CCHRECNO              Is it TOF already?
         BNE   UPC3ØØ                   No, do the work
         OI    UIDOPT4,UIDBALM          Yes, beep beep
         B     UPC9ØØ                   Done...
         SPACE
UPC3ØØ   L     R7,UIDBUFF2              Address last screen line
         LINK  DELETE                   Delete it
         L     R7,UIDBUFF1              Address first screen line
         GO    CSCRDFPR                 Get previous record
         BNZ   UPC4ØØ                   Not found, add TOF
         SR    R1,R1                    Add as first record
         LINK  ADD
         BCT   R6,UPC3ØØ                Do all lines
         B     UPC7ØØ
         SPACE
UPC4ØØ   LINK  ADDTOFT                  Add TOF as first record
UPC7ØØ   TM    UIDOPT2,UIDAUTO          Is user in Refresh mode
         BZ    UPC8ØØ                   No, do it
         NI    UIDOPT2,X'FF'-UIDAUTO    Reset AUTO Refresh option
         OI    UIDOPT4,UIDBHDR          Remember to refresh Header line
         L     R7,UIDBUFF1              Make Top line valid
         MVI   CCHLINE2,X'FF'
         GO    CSCUSCRB                 Rebuild screen
         B     UPC9ØØ
         SPACE
UPC8ØØ   L     R7,UIDBUFF1              Make Top and Bottom lines valid
         MVI   CCHLINE2,X'FF'              in case WRAP is turned off
         L     R7,UIDBUFF2
         MVI   CCHLINE2,X'FF'
         OI    UIDOPT4,UIDBSCR          Option to build user screen
         TM    UIDOPT3,UIDWRAP          Is WRAP switch on?
         BZ    UPC9ØØ                   No, done
         GO    CSCWRPTP                 Yes, build partial lines
UPC9ØØ   B     RETURN
         SPACE
UPC5ØØ   MSG   Ø311,USER                We got an invalid operand
         B     UPC9ØØ                   That's all
         SPACE
UPC6ØØ   MSG   Ø312,USER                Too many operands
         B     UPC9ØØ                   That's all
         SPACE 3
*
* Process DOWN command
*
*
CSCUSBDN RELOC                          DOWN command
         SR    RØ,RØ                    No table to search
         GO    CSCSCN                   Scan parameter
         LA    R2,1                     Default is 1
         BNZ   DNC1ØØ                   Nothing found, use default
         GO    CSCSCNVN                Validate parameter
         BNZ   DNC5ØØ                  Not numeric, that's an error
         SR    RØ,RØ                   Do not search any table
         GO    CSCSCN                  Anything left?
         BZ    DNC6ØØ                  Yes, bad news, only one parm
DNC1ØØ   LTR   R6,R2                   Copy repetition factor
         BZ    DNC9ØØ                  It is zero, all done
         L     R7,UIDBUFF2             Check last line on screen
         GO    CSCWRPGB                Locate last line on screen
         C     R7,UIDBUFF2             Same as first line in buffer
         BE    DNC2ØØ                  Yes, half done
         GO    CSCUSCBL
         L     R7,UIDBUFF2             Address new top line
DNC2ØØ   SR    RØ,RØ
         C     RØ,CCHRECNO             Is it EOF already?
         BNE   DNC3ØØ                  No, do the work
         OI    UIDOPT4,UIDBALM         Yes, beep beep
         B     DNC9ØØ                  Done...
         SPACE
DNC3ØØ   L     R7,UIDBUFF1             Address first screen line
         LINK  DELETE                  Delete it
         L     R7,UIDBUFF2             Address last screen line
         GO    CSCRDFNT                Get next record
         BNZ   DNC4ØØ                  Not found, add EOF
         L     R1,UIDBUFF2             Add after last record
         LINK  ADD
         BCT   R6,DNC3ØØ               Do all lines
         B     DNC7ØØ
         SPACE
DNC4ØØ   LINK  ADDEOFB                 Add EOF after last record
DNC7ØØ   TM    UIDOPT2,UIDAUTO         Is user in Refresh mode
         BZ    DNC8ØØ                  No, do it
         NI    UIDOPT2,X'FF'-UIDAUTO   Reset AUTO Refresh option
         OI    UIDOPT4,UIDBHDR         Remember to refresh Header line
         L     R7,UIDBUFF2             Make Bottom line valid
         MVI   CCHLINE2,X'FF'
         GO    CSCUSCRB                Rebuild screen
DNC8ØØ   L     R7,UIDBUFF1             Make Top and Bottom lines valid
         MVI   CCHLINE2,X'FF'             in case WRAP is turned off
         L     R7,UIDBUFF2
         MVI   CCHLINE2,X'FF'
         OI    UIDOPT4,UIDBSCR         Option to build user screen
         TM    UIDOPT3,UIDWRAP         Is WRAP switch on?
         BZ    DNC9ØØ                  No, done
         GO    CSCWRPBT                Yes, build partial lines
DNC9ØØ   B     RETURN
         SPACE
DNC5ØØ   MSG   Ø311,USER               We got an invalid operand
         B     DNC9ØØ                  That's all
         SPACE
```

```
DNC6ØØ   MSG   Ø312,USER              Too many operands
         B     DNC9ØØ                 That's all
         SPACE 3
         CSCDATA
         CSCDS (UID,CCH)
         REGEQU
         END
```

CSCUPR ASSEMBLE

This module adds support for the WRITE and PRINT commands.

```
         TITLE 'CSCUPR - CSC Process User Print/Write commands'
CSCUPR   START X'Ø1E29Ø'
         PRINT NOGEN
         CSCHDR                       Print/Write commands
*
* Process PRINT command
*
         USING UIDSECT,R8             UID (user) Block
         USING CCHSECT,R7             CCH (cache) Block
         SPACE
         LA    R1,COMMPRT             Address PRINT command
         B     PWRITE                 Execute common code
         SPACE 3
*
* Process WRITE Command
*
CSCUPRWR RELOC                        Process WRITE Command
         LA    R1,COMMWRT
*        B     PWRITE
         SPACE 3
*
* Common code to PRINT/WRITE Commands
*
PWRITE   EQU   *                      PRINT/WRITE common code
         L     R2,UIDSCRN             Address user screen
         MVC   4(L'COMMWRT,R2),Ø(R1)  Move command name
         SR    RØ,RØ                  Clear control word
         ST    RØ,4+L'COMMWRT(,R2)
         L     R7,UIDFREE1            Possible next line to transfer
         TM    UIDOPT3,UIDPPROG       Is command already in progress?
         BO    PWR3ØØ                 Yes, process another block
         SR    RØ,RØ                  No table to search
         GO    CSCSCN                 Locate next operand
         L     R2,PWDFLT              Default lines to print
         BNZ   PWR13Ø                 No operand found, use default
         CLI   SCANUPP,C'*'           Is it an asterisk
         BNE   PWR1ØØ                 No check for numeric
         CLI   SCANUPP+1,C' '          Make sure it is a single "*"
         BNE   PWR1ØØ
         SR    R2,R2                   Do not limit records to process
         B     PWR12Ø
         SPACE
PWR1ØØ   GO    CSCSCNVN                Is operand numeric?
         BNZ   PWR11Ø                  No, display error message
         LTR   R2,R2                   Zero is not a valid number
         BNZ   PWR12Ø
PWR11Ø   MSG   Ø311,USER               Display error message
         B     PWR9ØØ
         SPACE
PWR12Ø   SR    RØ,RØ                   No table to search
         GO    CSCSCN                  Locate next operand
         BNZ   PWR13Ø                  Nothing found... as expected
         MSG   Ø312,USER               Display error message
         B     PWR9ØØ
         SPACE
PWR13Ø   ST    R2,UIDPWREM             Number of records to process
         L     R6,UIDSCRN              Address user screen
         L     R7,UIDBUFF1             Address first line displayed
         L     RØ,CCHRECNO             Record number of first line
         LTR   RØ,RØ                   Is it a Top-Of-Data-File line
         BNZ   PWR2ØØ
         GO    CSCRDFFT                Yes, start with first record
         BZ    PWR2ØØ
         MSG   Ø34Ø                    Data file is empty
         B     PWR9ØØ
         SPACE
PWR2ØØ   OI    UIDOPT3,UIDPPROG        Set In Progress option
         LA    RØ,PWFIRST              Set First bit in control word
         ST    RØ,4+L'COMMWRT(,R6)
         NI    UIDOPT3,X'FF'-UIDPAUTO  Reset option
         TM    UIDOPT2,UIDAUTO         Was user in refresh mode?
         BZ    PWR32Ø                  No, let's do the job
         NI    UIDOPT2,X'FF'-UIDAUTO   Yes, reset option for now
         OI    UIDOPT3,UIDPAUTO        Remember to turn it back
         B     PWR32Ø
         SPACE
PWR3ØØ   L     RØ,6(,R6)               Load user code?
         LTR   RØ,RØ                   Is it zero?
         BZ    PWR32Ø                  Yes, process another block
         BAS   R14,CANCEL              No, user wants to cancel command
         B     PWR9ØØ
         SPACE
PWR32Ø   L     R6,UIDSCRN              Address user screen
         LA    R6,4+L'COMMWRT+4(,R6)   First available byte
PWR4ØØ   LA    RØ,UIDSCRSZ             Screen size in double words
         SLL   RØ,3                    Convert to bytes
         A     RØ,UIDSCRN              End of buffer address
         SR    R1,R1                   Required by next IC
         IC    R1,CCHRLEN              Length of message
         LA    R1,4+L'CCHPREF+1(R1,R6) Add prefix field
         TM    UIDOPT2,UIDDATE         Is Date being displayed
         BZ    PWR41Ø                  No...
         LA    R1,L'CCHDATE+1(,R1)     Add Date length plus separator
PWR41Ø   TM    UIDOPT2,UIDTIME         Is Time being displayed?
         BZ    PWR42Ø
         LA    R1,L'CCHTIME+1(,R1)     Add Time length plus separator
PWR42Ø   TM    UIDOPT2,UIDUSER
         BZ    PWR5ØØ
         LA    R1,L'CCHUSER+1(,R1)     Add User length plus separator
PWR5ØØ   CR    R1,RØ                   Do we have space in the block?
         BH    PWR7ØØ                  No, block is full, send it
         LA    R1,4(,R6)               Skip record length prefix
         MVC   Ø(L'CCHPREF,R1),CCHPREF Move Prefix
         MVI   L'CCHPREF(R1),C' '      Separate with a blank
         LA    R1,L'CCHPREF+1(,R1)     Advance pointer
         TM    UIDOPT2,UIDDATE         Is date being displayed?
         BZ    PWR51Ø
         MVC   Ø(L'CCHDATE,R1),CCHDATE Move Date
         MVI   L'CCHDATE(R1),C' '      Blank separator
         LA    R1,L'CCHDATE+1(,R1)     Advance pointer
PWR51Ø   TM    UIDOPT2,UIDTIME
         BZ    PWR52Ø
         MVC   Ø(L'CCHTIME,R1),CCHTIME Move Time
         MVI   L'CCHTIME(R1),C' '      Blank separator
         LA    R1,L'CCHTIME+1(,R1)     Advance pointer
PWR52Ø   TM    UIDOPT2,UIDUSER
         BZ    PWR53Ø
         MVC   Ø(L'CCHUSER,R1),CCHUSER Move User
         MVI   L'CCHUSER(R1),C' '
         LA    R1,L'CCHUSER+1(,R1)     Advance pointer
PWR53Ø   SR    R2,R2                   Required by next IC
         IC    R2,CCHRLEN              Get length of message text
         BCTR  R2,Ø                    Prepare to Execute
         EX    R2,PWRMVC               Move message text
         LA    R1,1(R2,R1)             Advance pointer
         SR    R1,R6                   Calculate total record length
         ST    R1,Ø(,R6)               Store as length prefix
         AR    R6,R1                   Next free byte
         L     RØ,UIDPWREM             Number of record left to print
         S     RØ,ONE                  Subtract one
         BZ    PWR6ØØ                  All done if zero
         ST    RØ,UIDPWREM             Store new value
         GO    CSCRDFNT                Read next record
         BZ    PWR4ØØ                  Found it, process it
         SR    RØ,RØ                   No more records in Data file
PWR6ØØ   ST    RØ,UIDPWREM             Zero records left to process
         L     R2,UIDSCRN              Address user screen
         LA    RØ,PWLAST                Set Last bit
         O     RØ,4+L'COMMWRT(,R2)      Add with previous Control word
         ST    RØ,4+L'COMMWRT(,R2)      Store new combined value
         TM    UIDOPT3,UIDPAUTO         Was user in refresh mode?
         BZ    PWR8ØØ
         OI    UIDOPT2,UIDAUTO          Yes, set option back
         B     PWR8ØØ                   Time to go home
         SPACE
PWR7ØØ   L     R1,UIDFREE1              Borrow free entry from user
         L     RØ,CCHRECNO              Store record number
         ST    RØ,CCHRECNO-CCHSECT(,R1)
         IC    RØ,CCHPREF               Store record Prefix
         STC   RØ,CCHPREF-CCHSECT(,R1)
         MVC   CCHDFREC-CCHSECT(L'CCHDFREC,R1),CCHDFREC Copy all DFREC
PWR8ØØ   L     R1,UIDSCRN               Address user buffer
         SR    R6,R1                    Total data length
         ST    R6,Ø(,R1)                Store as length prefix
         ST    R6,UIDSCRNL              Store as length to send
*        B     PWR9ØØ                   Go back, CSCUSC will send it
         SPACE
PWR9ØØ   BACK                           Bye
         SPACE
PWRMVC   MVC   Ø(*-*,R1),CCHDATA
         SPACE 3
*
* Command PRINT / WRITE cancelled by user program
*
CANCEL   EQU   *
         ST    R14,CANCSV14
         NI    UIDOPT3,X'FF'-UIDPPROG  Reset In progress option
         TM    UIDOPT3,UIDPAUTO         Was user in refresh mode?
         BZ    CANC1ØØ
         OI    UIDOPT2,UIDAUTO          Yes, set option back
CANC1ØØ  MSG   Ø341,(USER,NOCMD)        Display generic error message
         L     R14,CANCSV14
         BR    R14
         SPACE 3
CANCSV14 DS    F                        Save R14 - CANCEL
PWDFLT   DC    F'2ØØ'                   Default records to PRINT/WRITE
PWFIRST  EQU   X'Ø1'                    First data block processed
PWLAST   EQU   X'Ø2'                    Last data block
         SPACE 3
         CSCDATA
         CSCDS (UID,CCH)
         REGEQU
         END
```

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

Fernando Duarte
Analyst (Canada)

VM news

Users of Query Management Facility (QMF) for VM can benefit from IBM's recently announced QMF for Windows.

QMF for Windows is a multi-purpose, multi-database query engine that provides a comprehensive user environment for formulating and sharing business reports, a robust Windows-based API for automating database querying, updating, and report distribution tasks, and centralized control over database resource consumption.

The software provides a multi-purpose enterprise query environment for large-scale business reporting, data sharing, server resource protection, application development, and native connectivity to all of the DB2 hosts (VM, OS/390, MVS, and VSE) and DB2 workstation platforms.

Enhancements include support for QMF linear processes, DB2 stored procedures, command line interface for enhanced automation, and additional Web publishing capabilities with HTML forms.

For further information contact your local IBM representative

* * *

Mirasoft has announced its VM Timing Facility Monitors, providing a set of diagnostic and testing aids for applications that are date and/or time sensitive. VM Timing Facility Monitors can analyse and test programs that need to run in the year 2000 and beyond.

For further information contact:
Mirasoft, 60 Alban Street, Boston, MA 02124-3709, USA.
Tel: (617) 825 9121.
Jemasys, 37 Ridgeway, Wargrave, Berkshire, RG10 8AS, UK.
Tel: (01189) 404878.
URL: http://www.mirasoft.com.

* * *

IBM has announced ReaderThief, a functional replacement for the OV/VM component responsible for moving mail from an OV/VM client's virtual reader to their in-basket.

ReaderThief provides performance improvements for processing mail and enhancements for better Internet mail interaction. ReaderThief is an optional add-on to OV/VM and will be made available via the Web. It will not be shipped as part of OV/VM via the service process.

For further information contact your local IBM representative.

* * *