# 152 CICS update

*July 1998*

## In this issue


3 Altering CICS date and time
8 Using EIBFN codes – revisited
10 Managing CICS printers
26 Determining the library using PINQPGM
34 SAS CICS workload manager
48 CICS news

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

## North American office
Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75067
USA
Telephone: 940 455 7050

## Contributions
Articles published in *CICS Update* are paid for at the rate of £170 ($250) per 1000 words and £90 ($140) per 100 lines of code 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*.

## *CICS Update* on-line
Code from *CICS Update* can be downloaded from our Web site at http://www.xephon.com; you will need the user-id shown on your address label.

## Editor
Robert Burgess

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

## Subscriptions and back-issues
A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £170.00 in the UK; $260.00 in the USA and Canada; £176.00 in Europe; £182.00 in Australasia and Japan; and £180.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £14.50 ($22.00) each including postage.



*Printed in England.*

# Altering CICS date and time

Today's installations often need to alter the CICS date and time – perhaps the enterprise spans multiple time zones and must display local times to end-users; or perhaps developers wish to simulate the year 2000 and beyond when preparing their applications for the next millenium. This article shows how to achieve the desired result by exploiting the EXEC interface program global user exits.

## METHOD

The XEIIN global user exit (GLUE) receives control before the execution of EXEC CICS commands. To control the date and time displayed on application screens, we need to intercept the ABSTIME input parameter to FORMATTIME requests and alter it to our specification. The example in our XEIIN$ source code assumes that we want to add one hour to the time that CICS returns. In a more realistic case, we would probably interrogate USERID and/or LUNAME to identify individual local time zones.

This method works well when we only want to modify the date and time as formatted and presented on the screen. If, in addition to this requirement, we want to change the date and time as handled by the application, we must code an XEIOUT GLUE instead of XEIIN. XEIOUT gains control after the execution of EXEC CICS commands. For our purposes, this exit allows us to intercept output parameters from ASKTIME requests, and gives us an opportunity to alter the returned ABSTIME as well as the EIBDATE and EIBTIME.

Deploying our XEIOUT$ sample GLUE simulates the moment everyone is talking about these days – twelve midnight on 31 December 1999.

Using this simple framework, we can construct a sophisticated Year 2000 testing tool. An interface could be written to allow users to specify the dates and times against which they wish to test individual transactions. The data could be stored in a Global Work Area (GWA) associated with the XEIOUT GLUE. The exit code would then be

written to pass back the date and time found in the GWA for each user/transaction combination.

Both of the exits presented in this article were coded and tested on a CICS/ESA Version 4 Release 1 system.

To enable either exit, use the following command in a PLTPI program:

```
EXEC CICS ENABLE EXIT(exit_point) PROGRAM(exit_program) START
```

IBM documentation for all user exits is presented in the *Customization Guide*.

SOURCE CODE FOR XEIIN$:

```
         TITLE 'XEIIN GLOBAL USER EXIT'
         PRINT ON,NOGEN
         DFHUEXIT TYPE=EP,ID=XEIIN
         DFHUEXIT TYPE=XPIENV      EXIT PROGRAMMING INTERFACE (XPI)
         COPY DFHSMMCY             PARMLIST FOR STORAGE CONTROL XPI
*
XEIINDS  DSECT                     DUMMY SECT FOR GETMAINED STORAGE
XEIPLIST DS    A                           POINTER TO APPL PARMLIST
XEISAVR  DS    2F
XEIABST  DS    PL8                         WORK AREA FOR ABSTIME
XEIEDIT  DS    CL15                        EDIT AREA FOR ABSTIME
XEIINLN  EQU   *-XEIINDS
         USING XEIINDS,R3
*
XEIIN$   CSECT
XEIIN$   AMODE 31
XEIIN$   RMODE ANY
         SAVE  (14,12)                    SAVE CALLER REGS
         LR    R12,R15                    ESTABLISH PGM BASE REG
         USING XEIIN$,R12
         LR    R2,R1                      ADDRESS GLUE PARMLIST
         USING DFHUEPAR,R2
CHKARG0  DS    0H
         L     R6,UEPARG
         L     R6,0(R6)                   LOOK AT ARGUMENT 0
         CLC   0(2,R6),FTIME              IS IT FORMATTIME?
         BNE   GLUEXIT                    N - THEN SEE YA
GETMAIN  DS    0H
         LA    R9,XEIINLN                 SET LENGTH FOR GETMAIN
         L     R4,UEPXSTOR                ADDRESS XPI STORAGE
         USING DFHSMMC_ARG,R4             MAP XPI STORAGE
         L     R13,UEPSTACK               SAVE EXIT HANDLER STACK
         DFHSMMCX CALL,CLEAR,IN,                                    X
               FUNCTION(GETMAIN),                                   X
```

```
              GET_LENGTH((R9)),                                        X
              INITIAL_IMAGE(X'ØØ'),                                    X
              STORAGE_CLASS(USER),                                     X
              SUSPEND(NO),                                             X
              OUT,                                                     X
              ADDRESS((R3)),                                           X
              RESPONSE(*),                                             X
              REASON(*)
         CLI  SMMC_RESPONSE,SMMC_OK       DO WE HAVE STORAGE?
         BE   TIMEWARP                    Y - CONTINUE
         LA   R9,MSGBADGM                 N - POINT TO BAD NEWS
         BAL  R5,NOTIFY                       WRITE IT
         B    GLUEXIT                         RETURN
TIMEWARP DS   ØH
         L    R6,UEPRSA                   POINT TO APPL PGM'S RSA
         L    R6,24(R6)                   PICK UP REGISTER ONE
         ST   R6,XEIPLIST                 SAVE IT
         L    R6,4(R6)                    GET 2ND PARM IN LIST
         MVC  XEIABST,Ø(R6)               THIS IS OUR ABSTIME
         TM   XEIABST+7,X'ØC'             HI ORDER SIGN BITS ON?
         BC   12,EDITFAIL                 N - CAN'T TOUCH THIS
         TM   XEIABST+7,X'Ø3'             LO ORDER SIGN BITS ØØ OR 11?
         BC   4,EDITFAIL                  N - SIGN NEGATIVE OR GARBAGE
         UNPK XEIEDIT,XEIABST             SET UP
         OI   XEIEDIT+14,X'FØ'            ... NUMERIC EDIT
         STM  R1,R2,XEISAVR               SAVE TO UNDO TRT CORRUPTION
         TRT  XEIEDIT,XLTAB               ALL X'FØ' THRU X'F9'?
         LM   R1,R2,XEISAVR
         BNZ  EDITFAIL                    N - ABSTIME INVALID
         AP   XEIABST,ONEHR               HAVE OUR WAY WITH ABSTIME
         LA   R6,XEIABST                  R6 -> MODIFIED ABSTIME
         L    R7,XEIPLIST                 R7 -> PARMLIST
         ST   R6,4(R7)                    REDIRECT PARM 2
         B    FREEMAIN
EDITFAIL DS   ØH
         LA   R9,MSGEDTFL
         BAL  R5,NOTIFY
         B    FREEMAIN
NOTIFY   DS   ØH
         WTO  MF=(E,(R9))
         BR   R5
FREEMAIN DS   ØH
         L    R4,UEPXSTOR                 ADDRESS XPI STORAGE
         USING DFHSMMC_ARG,R4             MAP XPI STORAGE
         L    R13,UEPSTACK                SAVE EXIT HANDLER STACK
         DFHSMMCX CALL,CLEAR,IN,                                       X
               FUNCTION(FREEMAIN),                                     X
               ADDRESS((R3)),                                          X
               STORAGE_CLASS(USER),                                    X
               OUT,                                                    X
               RESPONSE(*),                                            X
```

```
               REASON(*)
          CLI  SMMC_RESPONSE,SMMC_OK
          BE   GLUEXIT
          LA   R9,MSGBADFM
          WTO  MF=(E,(R9))
          B    GLUEXIT
GLUEXIT   DS   ØH                          STANDARD GLUE EXIT CODE
          L    R13,UEPEPSA
          RETURN (14,12),RC=UERCNORM
*
*         WTO MACROS:
*
MSGBADFM WTO   'XEIIN - FREEMAIN FAILED',ROUTCDE=(14),MF=L
MSGBADGM WTO   'XEIIN - GETMAIN FAILED',ROUTCDE=(14),MF=L
MSGEDTFL WTO   'XEIIN - DETECTED INVALID ABSTIME',ROUTCDE=(14),MF=L
*
*         CONSTANTS:
*
FTIME     DC   XL2'4AØ4'                  ARGØ FOR FORMATTIME
ONEHR     DC   PL8'36ØØØØØ'                6Ø MINUTES IN MILLISECONDS
*
*         TRANSLATE TABLE:
*
XLTAB     DS   ØH
          DC   24ØX'FF'
          DC   1ØX'ØØ'
          DC   6X'FF'
*
          LTORG
          END  XEIIN$
```

## SOURCE CODE FOR XEIOUT$:

```
          TITLE 'XEIOUT GLOBAL USER EXIT'
          PRINT ON,NOGEN
          DFHUEXIT TYPE=EP,ID=XEIOUT
          DFHUEXIT TYPE=XPIENV      EXIT PROGRAMMING INTERFACE (XPI)
          COPY DFHSMMCY             PARMLIST FOR STORAGE CONTROL XPI
          COPY DFHEIBLK             EXECUTE INTERFACE BLOCK DSECT
*
DFHEIBR   EQU  R11                  EIB REGISTER
*
XEIOUTDS DSECT                      DUMMY SECT FOR GETMAINED STORAGE
XEIPLIST DS   A                             POINTER TO APPL PARMLIST
XEIABST  DS   PLØ8                          WORK AREA FOR ABSTIME
XEIOUTLN EQU  *-XEIOUTDS
          USING XEIOUTDS,R3
*
XEIOUT$  CSECT
XEIOUT$  AMODE 31
```

```
XEIOUT$  RMODE ANY
         SAVE  (14,12)                    SAVE CALLER REGS
         LR    R12,R15                    ESTABLISH PGM BASE REG
         USING XEIOUT$,R12
         LR    R2,R1                      ADDRESS GLUE PARMLIST
         USING DFHUEPAR,R2
CHKARGØ  DS    ØH
         L     R6,UEPARG
         L     R6,Ø(R6)                   LOOK AT ARGØ
         CLC   Ø(2,R6),ASKTIME            IS IT ASKTIME?
         BNE   GLUEXIT                    N - THEN SEE YA
GETMAIN  DS    ØH
         LA    R9,XEIOUTLN                SET LENGTH FOR GETMAIN
         L     R4,UEPXSTOR                ADDRESS XPI STORAGE
         USING DFHSMMC_ARG,R4             MAP XPI STORAGE
         L     R13,UEPSTACK               SAVE EXIT HANDLER STACK
         DFHSMMCX CALL,CLEAR,IN,                                      X
               FUNCTION(GETMAIN),                                     X
               GET_LENGTH((R9)),                                      X
               INITIAL_IMAGE(X'ØØ'),                                  X
               STORAGE_CLASS(USER),                                   X
               SUSPEND(NO),                                           X
               OUT,                                                   X
               ADDRESS((R3)),                                         X
               RESPONSE(*),                                           X
               REASON(*)
         CLI   SMMC_RESPONSE,SMMC_OK      DO WE HAVE STORAGE?
         BE    TIMEWARP                   Y - CONTINUE
         LA    R9,MSGBADGM                N - POINT TO BAD NEWS
         BAL   R5,NOTIFY                      WRITE IT
         B     GLUEXIT                        RETURN
TIMEWARP DS    ØH
         L     R6,UEPRSA                  POINT TO APPL PGM'S RSA
         L     R6,24(R6)                  PICK UP REGISTER UNO
         ST    R6,XEIPLIST                SAVE IT
         ZAP   XEIABST,Y2KABST
         L     R11,UEPEXECB               ADDRESS SYSEIB
         ZAP   EIBDATE,Y2KDATE
         ZAP   EIBTIME,Y2KTIME
         LA    R6,XEIABST          R6 -> MODIFIED ABSTIME
         L     R7,XEIPLIST         R7 -> PARMLIST
         L     R7,4(R7)               ... NOW TO ABSTIME PARM
         MVC   Ø(8,R7),XEIABST     OVERLAY APPL ABSTIME WITH OUR OWN
         B     FREEMAIN
NOTIFY   DS    ØH
         WTO   MF=(E,(R9))
         BR    R5
FREEMAIN DS    ØH
         L     R4,UEPXSTOR                ADDRESS XPI STORAGE
         USING DFHSMMC_ARG,R4             MAP XPI STORAGE
         L     R13,UEPSTACK               SAVE EXIT HANDLER STACK
```

```
         DFHSMMCX CALL,CLEAR,IN,                                       X
               FUNCTION(FREEMAIN),                                     X
               ADDRESS((R3)),                                          X
               STORAGE_CLASS(USER),                                    X
               OUT,                                                    X
               RESPONSE(*),                                            X
               REASON(*)
         CLI   SMMC_RESPONSE,SMMC_OK
         BE    GLUEXIT
         LA    R9,MSGBADFM
         WTO   MF=(E,(R9))
         B     GLUEXIT
GLUEXIT  DS    ØH                          STANDARD GLUE EXIT CODE
         L     R13,UEPEPSA
         RETURN (14,12),RC=UERCNORM
*
*        WTO MACROS:
*
MSGBADFM WTO   'XEIOUT - FREEMAIN FAILED',ROUTCDE=(14),MF=L
MSGBADGM WTO   'XEIOUT - GETMAIN FAILED',ROUTCDE=(14),MF=L
*
*        CONSTANTS:
*
ASKTIME  DC    XL2'4AØ2'                   ARGØ FOR ASKTIME(ABSTIME)
Y2KABST  DC    PL8'+3155673600000'         12 MIDNIGHT, DEC 31 1999
Y2KDATE  DC    PL4'+ØØØØ1'
Y2KTIME  DC    PL4'+ØØØØØ'
*
         LTORG
         END   XEIOUT$
```

*Russell Hunt*
*Senior Systems Programmer*
*Great Lakes Higher Education Corporation (USA)* 

## Using EIBFN codes – revisited

*Using EIBFN codes* was published in *CICS Update*, Issue 115, June 1995. Here is a simpler method of performing a nibble to byte conversion in COBOL. All you have to do is make it a nested subprogram of XXEIBFN and then CALL it passing two fields – one byte of the function code each time.

Example of call:

```
    Call 'Convert-Nibble-2-Byte' Using L-EIBFN         W-FCODE-X.
```

 

```
Call 'Convert-Nibble-2-Byte' Using L-EIBFN (2:)   W-F-CODE-X (3:)
```

There is no looping, just simple COBOL at its best.

## NIB2BYTE

```
ID DIVISION.
PROGRAM-ID. NIB2BYTE.
DATA DIVISION.
WORKING-STORAGE SECTION.
Ø1  XDATA     PIC XX     VALUE  X'EFA3'.
Ø1  ANS-FLD   PIC X(Ø4)  VALUE  SPACES.
PROCEDURE DIVISION.
    CALL 'Convert-Nibble-To-Byte' using XDATA (1:) ans-fld    .
    CALL 'Convert-Nibble-To-Byte' using XDATA (2:) ans-fld(3:).
    DISPLAY 'THE ANSWER IS ', ANS-FLD.
    GOBACK.

ID DIVISION.
PROGRAM-ID.  Convert-Nibble-To-Byte.
AUTHOR.      Jerome A Lumpkin.
DATA DIVISION.

WORKING-STORAGE SECTION.
Ø1  A-BIN                      PIC 9(2) COMP.
Ø1  A-BIN-REDEF REDEFINES A-BIN.
    Ø5  A-HI-ORDER             PIC X.
    Ø5  A-LO-ORDER             PIC X.
Ø1  TBL PIC X(16) VALUE 'Ø123456789ABCDEF'.

LINKAGE SECTION.
Ø1  A-HEX-VALUE                PIC X .
Ø1  ANS                        PIC XX.

PROCEDURE DIVISION USING A-HEX-VALUE, ANS.
Convert-One-Byte-To-Two.
    MOVE Ø             TO    A-BIN.
    MOVE A-HEX-VALUE   TO    A-LO-ORDER.
    MOVE TBL ((A-BIN / 16) + 1 : 1)                 TO ANS(1:).
    MOVE TBL ((A-BIN +  1) - ((A-BIN / 16) * 16):1) TO ANS(2:).
    EXIT PROGRAM.
END PROGRAM  Convert-Nibble-TO-Byte.
END PROGRAM  NIB2BYTE.
```

*Jerome A Lumpkin*
*Consultant*
*Support Work (USA)*

# Managing CICS printers

INTRODUCTION

The programs PRINTMNU, PRINTALL, and PRINTBRW allow you to manage CICS SNA printers without the need to invoke the master terminal transaction CEMT.

In addition, it is also possible to store and update information about the printers, such as users, location, type, netname, emulation, etc in a VSAM KSDS file.

The programs are written in COBOL II and are designed in a pseudo-conversational style. They are developed under CICS/ESA Version 4 Release 1, but should also run under previous releases.

SOLUTION

The transactions to manage CICS printers are:

- PENU – display the Operator Instruction Menu; program PRINTMNU.
- PADD – add a printer record to file PRINT; program PRINTALL.
- PDEL – delete a printer record; program PRINTALL.
- PINQ – inquire about printer information and status; program PRINTALL.
- PUPD – update printer information and status; program PRINTALL.
- PBRW – browse all available printer records; program PRINTBRW.

Note: every transaction dispatched for the first time starts with the Operator Instruction Menu.

A sample output from the program PRINTALL is shown in Figure 1. Figure 2 shows a sample output from the program PRINTBRW.

```
    PRINTER UPDATE
     Printer-ID  : PZØ3
     Netname     : PZØ3A78Ø
     TD-Queue    : PZØ3
     Manufacturer: Rank-Xerox Docu
     Model       : 4517
     Emulation   : XES / LJ3D-PCL
     JES-Printer : PRT18
     Outclass    : O
     Application : IJES,RP5Ø,MBØ4
     Users       : Halassek, Lentz, Lorenzen
     Department  : FR
     Building    : C2
     Floor       : 1. Etage
     Room        : 121
     Status      : In/Out     Cre/No     Rel/Acq    TTI/No     ATI/No  Transid
                   INSERV     CREATE     RELEAS     TTI        ATI

    CHANGE FIELDS AND PRESS ENTER
```

*Figure 1: Sample output from program PRINTALL*

## FCT DEFINITION FOR KSDS PRINT FILE

```
OBJECT CHARACTERISTICS                          CICS RELEASE = Ø41Ø
 CEDA  View File( PRINT    )
  File            : PRINT
  Group           : FCT1T
  DEScription     :
 VSAM PARAMETERS
  DSNAme          :
  Password        :                    PASSWORD NOT SPECIFIED
  Lsrpoolid       : 5                  1-8 | None
  DSNSharing      : Allreqs            Allreqs | Modifyreqs
  STRings         : ØØ3                1-255
  Nsrgroup        :
 REMOTE ATTRIBUTES
  REMOTESystem    :
  REMOTEName      :
  RECORDSize      : ØØ14Ø              1-32767
  Keylength       : ØØ4                1-255
 INITIAL STATUS
  STAtus          : Enabled            Enabled | Disabled | Unenabled
```

```
PRT   NETNAME   JES    USERS                                    MANUFACTURER
PK16  PK16A021         Pfeiffer, Ditzel, Herger, Wunderlich     IBM
PSØ7  PSØ7C515         Mayer,Simon, Heeg, Heil,Dorr(Michelbach) IBM
PZØ1  PZØ1A78Ø  PRT12  Becker, Cress, Klug, Nickel, Gerstmann   Rank-Xerox
PZØ2  PZØ2A78Ø         Vanco,Salm,Becker,Cress,Konig,Nickel     IBM
PZØ3  PZØ3A78Ø  PRT18  Halassek, Lentz, Lorenzen                Rank-Xerox Docu
PZØ6  PZØ6A78Ø  PRT15  Vanco,Salm,Becker,Cress,Konig,Nickel     Rank-Xerox

PRESS CLEAR TO END BROWSE OPERATION
PRESS PF8 OR TYPE F TO PAGE FORWARD
PRESS PF7 OR TYPE B TO PAGE BACKWARD
```

*Figure 2: Sample output from program PRINTBRW*

```
 Opentime       : Firstref           Firstref | Startup
 DIsposition    : Share              Share | Old
BUFFERS
 DAtabuffers    : ØØØØ4              2-32767
 Indexbuffers   : ØØØØ3              1-32767
DATATABLE PARAMETERS
 Table          : No                 No | CICS | User
 Maxnumrecs     :                    16-16777215
DATA FORMAT
 RECORDFormat   : F                  V | F
OPERATIONS
 Add            : Yes                No | Yes
 BRowse         : Yes                No | Yes
 DELete         : Yes                No | Yes
 REAd           : Yes                Yes | No
 Update         : Yes                No | Yes
```

## PRINTALL

```
*****************************************************************
*                                                               *
* MODULE NAME = PRINTALL                                        *
*                                                               *
* DESCRIPTIVE NAME = FILE INQUIRY/UPDATE FOR PRINTER            *
*                                                               *
*****************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. PRINTALL.
```

```
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77  RECLENGTH    PIC S9(Ø4) COMP VALUE +14Ø.
 77  SERVSTATUS   PIC S9(Ø8) COMP.
 77  CRESTATUS    PIC S9(Ø8) COMP.
 77  TERMSTATUS   PIC S9(Ø8) COMP.
 77  TTISTATUS    PIC S9(Ø8) COMP.
 77  ATISTATUS    PIC S9(Ø8) COMP.
 77  PRIDNUM      PIC X(4).
 77  COMLEN       PIC S9(4) COMP.
 77  RESPONSE     PIC 9(9)  COMP.
 77  MESSAGES     PIC X(39).
               COPY PRTMAPA.
               COPY PRTMAPB.
 Ø1  PRINT.     COPY PRINTFIL.
               COPY DFHBMSCA.
 Ø1  COMMAREA.  COPY PRINTFIL.
 LINKAGE SECTION.
 Ø1  DFHCOMMAREA. COPY PRINTFIL.

 PROCEDURE DIVISION.
*
*    THE LENGTH OF THE COMMAREA IS TESTED. IF NOT ZERO THEN
*    THIS IS THE VALIDATION STAGE OF AN ADD OR UPDATE.
*
*    IF IT HAS A LENGTH, THE COMMAREA RETURNED IS MOVED TO
*    WORKING STORAGE IN THE PROGRAM.
*
     IF EIBCALEN NOT = Ø THEN
        MOVE DFHCOMMAREA TO COMMAREA GO TO READ-INPUT.
*
*    THE MENU MAP PRTMAPA IS RECEIVED. THE PRINTER-ID, IF
*    ENTERED, IS MAPPED INTO KEYI IN THE DSECT FOR PRTMAPA.
*    THE RESPONSE TO THE COMMAND IS EXPLICITLY TESTED BY
*    THE PROGRAM.
*
     EXEC CICS RECEIVE MAP('MENU') MAPSET('PRTMAPA')
               RESP(RESPONSE) END-EXEC.
*
*    CHECK RESPONSE TO COMMAND
*
     IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP.
     IF KEYL = ZERO THEN GO TO BADLENG.
*
*    THE PRINTER-ID IS VALIDATED AND SAVED.
*
     MOVE KEYI TO PRIDNUM.
     MOVE LOW-VALUES TO DETAILO.
*
```

```
*    IF THE PROGRAM IS INVOKED BY PDEL DELETE PRIDNUM
*    IS PERFORMED AND RESPONSE TO COMMAND CHECKED.
*
     IF EIBTRNID = 'PDEL' THEN
        PERFORM FILE-DELETE
        MOVE 'RECORD DELETED' TO MESSAGES GO TO MENU.
*
*    IF THE PROGRAM IS INVOKED BY PADD, A TITLE AND COMMAND
*    MESSAGE ARE MOVED TO THE MAP AREA.
*    THE RECORD KEY IS MOVED TO THE MAP AREA AND SAVED IN
*    COMMAREA.
*
     IF EIBTRNID = 'PADD' THEN
        MOVE 'PRINTER ADD' TO TITLEO
        MOVE 'ENTER DATA AND PRESS ENTER KEY' TO MSG3O
*
*       THE RECORD KEY IS MOVED TO THE COMMAREA AND TO THE
*       MAP AREA.
*
        MOVE KEYI TO PRID IN COMMAREA, PRIDO
        MOVE 5 TO COMLEN GO TO MAP-SEND.
*
*    THE FILE CONTROL READ COMMAND READS THE FILE RECORD INTO
*    THE FILE AREA PRINT IN PGM PRINTALL.
*
     EXEC CICS READ FILE('PRINT') INTO(PRINT) RIDFLD(PRIDNUM)
          LENGTH(RECLENGTH) RESP(RESPONSE) END-EXEC.
*
*    CHECK RESPONSE TO COMMAND
*
     IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP.
     IF EIBTRNID = 'PINQ' THEN
*
*       IF PROGRAM IS INVOKED BY PINQ, A TITLE AND COMMAND
*       MESSAGE ARE MOVED TO THE MAP AREA.
*
        MOVE 'PRINTER INQUIRY' TO TITLEO
        MOVE 'PRESS ENTER TO CONTINUE' TO MSG3O
*
*       ALL FIELD ATTRIBUTES IN PRTMAPB ARE PROTECTED.
*
        MOVE DFHBMPRO TO PRIDA
        MOVE DFHBMPRO TO NETNAMA
        MOVE DFHBMPRO TO TDQNAMA
        MOVE DFHBMPRO TO FIRMA
        MOVE DFHBMPRO TO MODELA
        MOVE DFHBMPRO TO PRMODEA
        MOVE DFHBMPRO TO PRMODEA
        MOVE DFHBMPRO TO JESPRTA
```

```
           MOVE DFHBMPRO TO OUTCA
           MOVE DFHBMPRO TO APPLICA
           MOVE DFHBMPRO TO USERA
           MOVE DFHBMPRO TO ABTA
           MOVE DFHBMPRO TO BUILDA
           MOVE DFHBMPRO TO FLOORA
           MOVE DFHBMPRO TO ROOMA
           MOVE DFHBMPRO TO SERVA
           MOVE DFHBMPRO TO CREA
           MOVE DFHBMPRO TO ACQA
           MOVE DFHBMPRO TO TTIA
           MOVE DFHBMPRO TO ATIA
*
*          THE FILE RECORD FIELDS ARE MOVED TO THE MAP AREA, AND
*          THE INQUIRY SCREEN IS DISPLAYED.
*
           PERFORM PRINTER-STATUS
           PERFORM MAP-BUILD THRU MAP-SEND
*
*          THE INVOCATION OF THE PROGRAM TERMINATES. THE TRANS-ID
*          OF PENU CAUSES THE OPERATOR INSTRUCTION PROGRAM TO BE
*          INVOKED WHEN THE NEXT RESPONSE IS RECEIVED FROM THE
*          TERMINAL.
*
           EXEC CICS RETURN TRANSID('PENU') END-EXEC.

       IF EIBTRNID = 'PUPD' THEN
*
*          IF THE PROGRAM IS INVOKED BY PUPD, A TITLE AND COMMAND
*          MESSAGE ARE MOVED TO THE MAP AREA.
*
           PERFORM PRINTER-STATUS
           MOVE 'PRINTER UPDATE' TO TITLEO
           MOVE 'CHANGE FIELDS AND PRESS ENTER' TO MSG3O
*
*          THE FILE RECORD AND THE PRINTER STATUS INPUT IS MOVED
*          TO THE COMMAREA AND THE LENGTH OF THE COMMAREA
*          TO BE RETURNED IS SET UP.
*
           MOVE FILEREC IN PRINT TO FILEREC IN COMMAREA
           MOVE 152 TO COMLEN.

 MAP-BUILD.
       MOVE PRID    IN PRINT TO PRIDO.
       MOVE NETNAM  IN PRINT TO NETNAMO.
       MOVE TDQNAM  IN PRINT TO TDQNAMO.
       MOVE FIRM    IN PRINT TO FIRMO.
       MOVE MODEL   IN PRINT TO MODELO.
       MOVE PRMODE  IN PRINT TO PRMODEO.
       MOVE JESPRT  IN PRINT TO JESPRTO.
```

```
     MOVE OUTC    IN PRINT TO OUTCO.
     MOVE APPLIC  IN PRINT TO APPLICO.
     MOVE USER    IN PRINT TO USERO.
     MOVE ABT     IN PRINT TO ABTO.
     MOVE BUILD   IN PRINT TO BUILDO.
     MOVE FLOOR   IN PRINT TO FLOORO.
     MOVE ROOM    IN PRINT TO ROOMO.

 MAP-SEND.
*
*    MAP-SEND SENDS THE MAP PRTMAPB TO THE SCREEN SPECIFYING
*    THAT THE SCREEN IS TO BE ERASED BEFORE THE MAP IS DISPLAYED.
*
     EXEC CICS SEND MAP('DETAIL') MAPSET('PRTMAPB')
               ERASE END-EXEC.
 FIN.
     EXEC CICS RETURN TRANSID(EIBTRNID) COMMAREA(COMMAREA)
               LENGTH(COMLEN) END-EXEC.
*
*    CONTROL IS PASSED HERE WHEN THE TEST OF EIBCALEN, AT THE
*    BEGINNING  OF THE PROGRAM, FINDS THAT A COMMAREA HAS BEEN
*    RECEIVED. THIS PART OF THE PROGRAM MAPS IN DATA FOR AN ADD
*    OR UPDATE REQUEST, PERFORMS VALIDATION, AND UPDATES PRINT.
*
 READ-INPUT.
*
*    THE RECEIVE MAP COMMAND MAPS IN THE VARIABLES FROM THE
*    SCREEN.
*
     EXEC CICS RECEIVE MAP('DETAIL') MAPSET('PRTMAPB')
               RESP(RESPONSE) END-EXEC.
*
*    CHECK RESPONSE TO COMMAND
*
     IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP.
     IF EIBTRNID = 'PUPD' THEN
*
*       IF THIS IS AN UPDATE REQUEST A FILE CONTROL READ UPDATE
*       READS THE EXISTING RECORD USING THE PRINTER STORED IN
*       COMMAREA BY THE LAST INVOCATION OF THIS PROGRAM.
*
        EXEC CICS READ UPDATE FILE('PRINT') INTO(PRINT)
             RESP(RESPONSE) RIDFLD(PRID IN COMMAREA)
             LENGTH(RECLENGTH) END-EXEC
*
*    CHECK RESPONSE TO COMMAND
*
        IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP
        ELSE
*
```

```
*       IF THE CURRENT FILE RECORD IS NOT THE SAME AS THE ONE
*       SAVED IN THE COMMAREA THEN ANOTHER USER HAS UPDATED THE
*       RECORD. A WARNING MESSAGE IS DISPLAYED, WITH FIELDS FROM
*       THE RECORD READ FROM PRINT, FOR RE-ENTRY OF THE UPDATES.
*
        IF FILEREC IN PRINT NOT = FILEREC IN COMMAREA THEN
           MOVE 'RECORD UPDATED BY OTHER USER, TRY AGAIN' TO MSG1O
           MOVE DFHBMASB TO MSG1A
           MOVE DFHPROTN TO MSG3A
           PERFORM MAP-BUILD
           EXEC CICS SEND MAP('DETAIL') MAPSET('PRTMAPB')
                     END-EXEC
           MOVE 152 TO COMLEN
           MOVE FILEREC IN PRINT TO FILEREC IN COMMAREA
           GO TO CICS-CONTROL
         ELSE
*
*          THE UPDATE FLAG IS SET IN THE RECORD AREA AND THE
*          MESSAGE RECORD UPDATED IS MOVED TO THE MESSAGE AREA
*          READY FOR DISPLAY ON THE OPERATOR INSTRUCTION SCREEN.
*
           PERFORM SET-PRINTER
           MOVE 'U' TO STAT IN PRINT
           PERFORM CHECK THROUGH FILE-WRITE
           MOVE 'RECORD UPDATED/STATUS CHANGED' TO MESSAGES
           GO TO MENU.
*
*    IF THIS IS AN ADD REQUEST THE ADD FLAG IS SET IN THE NEW
*    RECORD AND THE MESSAGE RECORD ADDED IS MOVED TO THE MESSAGE
*    AREA READY FOR DISPLAY ON THE OPERATOR INSTRUCTION SCREEN.
*
     IF EIBTRNID = 'PADD' THEN
           MOVE LOW-VALUES TO FILEREC IN PRINT
           MOVE 'A' TO STAT IN PRINT
           PERFORM CHECK THRU FILE-WRITE
           MOVE 'RECORD ADDED' TO MESSAGES GO TO MENU.
*
*    CHECK FIELDS ADDED/UPDATED
*
 CHECK.
     IF PRIDI    = LOW-VALUES AND
        NETNAMI  = LOW-VALUES AND
        TDQNAMI  = LOW-VALUES AND
        FIRMI    = LOW-VALUES AND
        MODELI   = LOW-VALUES AND
        PRMODEI  = LOW-VALUES AND
        JESPRTI  = LOW-VALUES AND
        OUTCI    = LOW-VALUES AND
        APPLICI  = LOW-VALUES AND
        USERI    = LOW-VALUES AND
```

```
           ABTI     = LOW-VALUES AND
           BUILDI   = LOW-VALUES AND
           FLOORI   = LOW-VALUES AND
           ROOMI    = LOW-VALUES
         THEN MOVE 'RECORD NOT MODIFIED/STATUS CHANGED' TO MESSAGES
                GO TO MENU.

 FILE-WRITE.
     IF EIBTRNID = 'PADD' THEN MOVE PRID IN COMMAREA TO
                                    PRID IN PRINT.
     IF PRIDI    NOT = LOW-VALUE MOVE PRIDI   TO PRID   IN PRINT.
     IF NETNAMI  NOT = LOW-VALUE MOVE NETNAMI TO NETNAM IN PRINT.
     IF TDQNAMI  NOT = LOW-VALUE MOVE TDQNAMI TO TDQNAM IN PRINT.
     IF FIRMI    NOT = LOW-VALUE MOVE FIRMI   TO FIRM   IN PRINT.
     IF MODELI   NOT = LOW-VALUE MOVE MODELI  TO MODEL  IN PRINT.
     IF PRMODEI  NOT = LOW-VALUE MOVE PRMODEI TO PRMODE IN PRINT.
     IF JESPRTI  NOT = LOW-VALUE MOVE JESPRTI TO JESPRT IN PRINT.
     IF OUTCI    NOT = LOW-VALUE MOVE OUTCI   TO OUTC   IN PRINT.
     IF APPLICI  NOT = LOW-VALUE MOVE APPLICI TO APPLIC IN PRINT.
     IF USERI    NOT = LOW-VALUE MOVE USERI   TO USER   IN PRINT.
     IF ABTI     NOT = LOW-VALUE MOVE ABTI    TO ABT    IN PRINT.
     IF BUILDI   NOT = LOW-VALUE MOVE BUILDI  TO BUILD  IN PRINT.
     IF FLOORI   NOT = LOW-VALUE MOVE FLOORI  TO FLOOR  IN PRINT.
     IF ROOMI    NOT = LOW-VALUE MOVE ROOMI   TO ROOM   IN PRINT.

     IF EIBTRNID = 'PUPD' THEN
*
*       FOR AN UPDATE REQUEST THE UPDATED PRINTER RECORD IS
*       REWRITTEN TO PRINT.
*
        EXEC CICS REWRITE FILE('PRINT') FROM(PRINT)
             LENGTH(RECLENGTH) RESP(RESPONSE) END-EXEC
*
*       CHECK RESPONSE TO COMMAND
*
        IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP
        ELSE NEXT SENTENCE
     ELSE
*
*       TRANSACTION IS 'PADD'. FOR AN ADD REQUEST THE NEW PRINTER
*       RECORD IS WRITTEN TO PRINT.
*
        EXEC CICS WRITE FILE('PRINT') FROM(PRINT)
             RIDFLD(PRID IN COMMAREA) RESP(RESPONSE)
             LENGTH(RECLENGTH) END-EXEC
        IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP.
 CICS-CONTROL.
*
*    AFTER THE FILE ADD OR FILE UPDATE SCREEN HAS BEEN
*    DISPLAYED THE PROGRAM BRANCHES HERE TO RETURN TO CICS
```

```
*    AWAITING A RESPONSE FROM THE TERMINAL. THE RETURN GIVES
*    CICS THE TRANSACTION IDENTIFIER FOR NEXT TRANSACTION AT
*    THIS TERMINAL, TOGETHER WITH A COMMAREA CONTAINING ALL
*    INFORMATION THAT THE PROGRAM NEEDS TO CONTINUE THE UPDATE.
*    THE COMMAREA IS PASSED TO THE NEXT INVOCATION OF THIS
*    PROGRAM.
*
     EXEC CICS RETURN TRANSID(EIBTRNID) COMMAREA(COMMAREA)
               LENGTH(COMLEN) END-EXEC.
*
*    THIS ROUTINE GAINS CONTROL WHENEVER A CICS COMMAND RETURNS A
*    NON-NORMAL RESPONSE. THE ROUTINE EXPLICITLY CHECKS FOR THE
*    RESPONSES NOTFND, DUPREC, AND MAPFAIL. THERE IS ALSO A
*    'CATCH ALL' TO TRAP ANY RESPONSE THAT IS NOT NORMAL. THE
*    ROUTINE MENU IS CALLED TO RE-DISPLAY THE MENU MAP, ALONG
*    WITH AN ERROR MESSAGE.
*
 CHECK-RESP.
     IF RESPONSE = DFHRESP(TERMIDERR) THEN
        MOVE 'Invalid VTAM-Printer - Please Re-enter' TO MESSAGES
        GO TO MENU.
     IF RESPONSE = DFHRESP(INVREQ) THEN
       MOVE 'Invalid Request for SET Printer Command' TO MESSAGES
        GO TO MENU.
     IF RESPONSE = DFHRESP(NOTOPEN) THEN
        MOVE 'File PRINT closed - Please open' TO MESSAGES
        GO TO MENU.
     IF RESPONSE = DFHRESP(NOTFND) THEN
        MOVE 'Invalid Printer-ID - Please Re-enter' TO MESSAGES
        GO TO MENU.
     IF RESPONSE = DFHRESP(DUPREC) THEN
        MOVE 'Duplicate Record' TO MESSAGES
        GO TO MENU.
     IF RESPONSE = DFHRESP(MAPFAIL) THEN
        IF EIBCALEN = Ø THEN
           MOVE 'Press Clear to Exit' TO MESSAGES
           GO TO MENU
        ELSE
           MOVE 'PRINTER RECORD/STATUS NOT MODIFIED' TO MESSAGES
           GO TO MENU.
     IF RESPONSE NOT = DFHRESP(NORMAL) THEN
        EXEC CICS DUMP DUMPCODE('ERRS') END-EXEC.
        MOVE 'Transaction Terminated' TO MESSAGES
        GO TO MENU.
*
*    THESE SHORT ERROR ROUTINES SET UP AN ERROR MESSAGE IN
*    MESSAGES AND BRANCH TO MENU TO DISPLAY THE MESSAGE
*    ON THE OPERATOR INSTRUCTION MENU PRTMAPA.
*
```

```
 BADLENG.
     MOVE 'PLEASE ENTER A PRINTER-ID' TO MESSAGES.
     GO TO MENU.
*
*    IF A CICS COMMAND FAILS WITH THE ERROR CONDITION, THE
*    MESSAGE 'ERROR. TRANSACTION TERMINATED' IS MOVED TO
*    MESSAGES FOR DISPLAY ON THE MENU SCREEN.
*
 ERRORS.
     MOVE 'ERROR. TRANSACTION TERMINATED' TO MESSAGES.
     GO TO MENU.
 MENU.
*
*    THIS CODE GETS CONTROL WHEN AN ADD, DELETE, OR UPDATE
*    REQUEST IS COMPLETE.
*    AN INFORMATION OR ERROR MESSAGE IS IN "MESSAGES".
*    THE OPERATOR INSTRUCTION MAP AREA IS CLEARED. THE MESSAGE
*    IS MOVED TO THE MAP AREA AND HIGHLIGHTED.
*
     MOVE LOW-VALUE TO MENUO.
     MOVE DFHBMASB  TO MSGA.
     MOVE MESSAGES  TO MSGO.
*
*    THE OPERATOR INSTRUCTION MAP "PRTMAPA" IS DISPLAYED ON AN
*    ERASED SCREEN.
*
     EXEC CICS SEND MAP('MENU') MAPSET('PRTMAPA')
               ERASE END-EXEC.
*
*    THE PROGRAM TERMINATES BY RETURNING TO CICS.
*    NO TRANSACTION IDENTIFIER OR "COMMAREA" IS SPECIFIED.
*
     EXEC CICS RETURN END-EXEC.
     GOBACK.
*
*       FOR A DELETE REQUEST THE "PRIDNUM" PRINTER RECORD IS
*       DELETED
*
 FILE-DELETE.
        EXEC CICS DELETE FILE('PRINT') RIDFLD(PRIDNUM)
                  RESP(RESPONSE) END-EXEC
*
*       CHECK RESPONSE TO COMMAND
*
        IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP
        ELSE NEXT SENTENCE.
*
*       INQUIRE TERMINAL STATUS OF "PRIDNUM"
*
```

```
 PRINTER-STATUS.
        EXEC CICS INQUIRE TERMINAL(PRIDNUM)
                      SERVSTATUS(SERVSTATUS)
                      CREATESESS(CRESTATUS)
                      TERMSTATUS(TERMSTATUS)
                      TTISTATUS(TTISTATUS)
                      ATISTATUS(ATISTATUS)
                      TRANSACTION(TRANO)
         RESP(RESPONSE) END-EXEC
*
*       CHECK RESPONSE TO COMMAND
*
        IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP.
        EVALUATE TRUE
          WHEN SERVSTATUS = DFHVALUE(GOINGOUT)
            MOVE 'GOINGO' TO SERVO
          WHEN SERVSTATUS = DFHVALUE(INSERVICE)
            MOVE 'INSERV' TO SERVO
          WHEN SERVSTATUS = DFHVALUE(OUTSERVICE)
            MOVE 'OUTSER' TO SERVO
        END-EVALUATE
        EVALUATE TRUE
          WHEN CRESTATUS = DFHVALUE(CREATE)
            MOVE 'CREATE' TO CREO
          WHEN CRESTATUS = DFHVALUE(NOCREATE)
            MOVE 'NOCRE ' TO CREO
          WHEN CRESTATUS = DFHVALUE(NOTAPPLIC)
            MOVE 'NOTAPP' TO CREO
        END-EVALUATE
        EVALUATE TRUE
          WHEN TERMSTATUS = DFHVALUE(ACQUIRED)
            MOVE 'ACQURE' TO ACQO
          WHEN TERMSTATUS = DFHVALUE(ACQUIRING)
            MOVE 'ACQING' TO ACQO
          WHEN TERMSTATUS = DFHVALUE(RELEASED)
            MOVE 'RELEAS' TO ACQO
          WHEN TERMSTATUS = DFHVALUE(RELEASING)
            MOVE 'RELING' TO ACQO
        END-EVALUATE
        EVALUATE TRUE
          WHEN TTISTATUS = DFHVALUE(NOTTI)
            MOVE 'NO TTI' TO TTIO
          WHEN TTISTATUS = DFHVALUE(TTI)
            MOVE 'TTI   ' TO TTIO
        END-EVALUATE
        EVALUATE TRUE
          WHEN ATISTATUS = DFHVALUE(NOATI)
            MOVE 'NO ATI' TO ATIO
          WHEN ATISTATUS = DFHVALUE(ATI)
            MOVE 'ATI   ' TO ATIO
```

```
        END-EVALUATE.
*
*       PERFOTM REQUIRED ACTION FOR PRINTER STATUS
*
 SET-PRINTER.
     IF SERVI(1:1) = 'I' OR 'i'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) INSERVICE
        RESP(RESPONSE) END-EXEC.
     IF SERVI(1:1) = 'O' OR 'o'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) OUTSERVICE
        RESP(RESPONSE) END-EXEC.
     IF CREI(1:1) = 'C' OR 'c'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) CREATE
        RESP(RESPONSE) END-EXEC.
     IF CREI(1:1) = 'N' OR 'n'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) NOCREATE
        RESP(RESPONSE) END-EXEC.
     IF ACQI(1:1) = 'A' OR 'a'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) ACQUIRED
        RESP(RESPONSE) END-EXEC.
     IF ACQI(1:1) = 'R' OR 'r'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) RELEASED
        RESP(RESPONSE) END-EXEC.
     IF TTII(1:1) = 'T' OR 't'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) TTI
        RESP(RESPONSE) END-EXEC.
     IF TTII(1:1) = 'N' OR 'n'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) NOTTI
        RESP(RESPONSE) END-EXEC.
     IF ATII(1:1) = 'A' OR 'a'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) ATI
        RESP(RESPONSE) END-EXEC.
     IF ATII(1:1) = 'N' OR 'n'
        EXEC CICS SET TERMINAL(PRID IN COMMAREA) NOATI
        RESP(RESPONSE) END-EXEC
     ELSE NEXT SENTENCE.
*
*       CHECK RESPONSE TO COMMAND
*
        IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO CHECK-RESP.
```

## PRINTBRW

```
*****************************************************************
*                                                               *
* MODULE NAME = PRINTBRW                                        *
*                                                               *
* DESCRIPTIVE NAME = File browse for printer                    *
*                                                               *
*****************************************************************
```

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID. PRINTBRW.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 Ø1  COMMAREA.
     Ø2  STATS    PIC X(1).
*                                          BUILDS PREV BACK PAGE
     Ø2  RIDB     PIC X(4).
*                                          BUILDS NEXT FWD PAGE
     Ø2  RIDF     PIC X(4).
*
 77  MESSAGES     PIC X(39) VALUE ' '.
*
 77  RESPONSE     PIC S9(8)  COMP.
*                                          BMS STD ATTRIBUTES
 77  RECLENGTH    PIC S9(Ø4) COMP VALUE +14Ø.
               COPY DFHBMSCA.
*
               COPY DFHAID.
*                                          PRINT RECORD DESCRIPT'N
 Ø1  PRINT.    COPY PRINTFIL.
*                                          GENERAL MENU MAP
               COPY PRTMAPA.
*                                          BROWSE PRINT MAP
               COPY PRTMAPC.
*
 LINKAGE SECTION.
  Ø1 DFHCOMMAREA.
     Ø2  STATS  PIC X(1).
     Ø2  RIDB   PIC 9(4).
     Ø2  RIDF   PIC 9(4).
*
 PROCEDURE DIVISION.
*
*    THE LENGTH OF THE COMMAREA IS TESTED. IF NOT ZERO THEN
*    THE NEXT OPERATOR COMMAND MUST BE RECEIVED FROM THE TERMINAL.
*
*    IF IT HAS A LENGTH, THE COMMAREA RETURNED IS MOVED TO
*    WORKING STORAGE IN THE PROGRAM.
*
     IF EIBCALEN NOT = Ø THEN
        MOVE DFHCOMMAREA TO COMMAREA
        GO TO PROMPT.
*
*    THIS COMMAND MAPS IN THE PRINTER-ID FROM THE OPERATOR
*    INSTRUCTION MENU. THE RESPONSE OF THE COMMAND IS
*    EXPLICITLY TESTED.
*
     EXEC CICS RECEIVE MAP('MENU') MAPSET('PRTMAPA')
```

```
           RESP(RESPONSE) END-EXEC.
*
*    CHECK RESPONSES
*
     IF RESPONSE = DFHRESP(MAPFAIL) THEN
        MOVE 'PRESS CLEAR TO EXIT' TO MESSAGES
        GO TO MENU.
     IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO ERRORS.
*
*    THE PRINTER-ID IS USED TO SET UP THE PROGRAM'S BROWSE
*    POINTERS. IF NO PRINTER-ID IS ENTERED, BROWSING BEGINS
*    AT THE START OF THE FILE.
*
     IF KEYL NOT = ZERO THEN
*
*          VALID INPUT
*
           MOVE KEYI TO RIDF IN COMMAREA
           MOVE KEYI TO RIDB IN COMMAREA
     ELSE
*
*       PRINTER-ID OMITTED
*
        MOVE '$$$$' TO RIDF IN COMMAREA.
*
*    THE STARTBR COMMAND ESTABLISHES THE BROWSE STARTING POINT.
*
     EXEC CICS STARTBR FILE('PRINT') RIDFLD(RIDF IN COMMAREA)
          RESP(RESPONSE) END-EXEC.
*
*    IF THE NOTFND CONDITION OCCURS AT THE START BROWSE, THE
*    MESSAGE 'END OF FILE - PLEASE RESTART' IS MOVED TO MESSAGES
*    FOR DISPLAY ON THE OPERATOR INSTRUCTION SCREEN.
*
     IF RESPONSE = DFHRESP(NOTFND) THEN
        MOVE 'END OF FILE - PLEASE RESTART' TO MESSAGES
        GO TO MENU.
     IF RESPONSE = DFHRESP(NOTOPEN) THEN
        MOVE 'FILE PRINT CLOSED - PLEASE OPEN' TO MESSAGES
        GO TO MENU.
     IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO ERRORS.
*
*    ENTERING THE MAXIMUM VALUE (9999) FOR THE PRINTER-ID
*    BEGINS A BACKWARD BROWSE FROM THE END OF FILE.
*
     IF RIDF IN COMMAREA NOT EQUAL '9999' THEN
        GO TO PAGE-FORWARD.
     MOVE 'H' TO STATS IN COMMAREA.
     GO TO PAGE-BACKWARD.
*
```

```
*    BUILD NEXT FORWARD PAGE
*
 PAGE-FORWARD.
*
*    TOP END OF FILE
*    RESET MAP PRTMAPC.
*
     MOVE LOW-VALUES TO BROWSEO.
*
*    IF THE PROGRAM HAS BEEN PASSED A COMMAREA, THEN THE BROWSE
*    MUST BE RESTARTED AT THE POINT THAT THE BROWSE FINISHED
*    PREVIOUSLY. THE KEY CONTAINED IN RIDF IS USED TO START
*    THE BROWSE.
*
     IF EIBCALEN = Ø THEN GO TO NEXT-LINE.
     EXEC CICS STARTBR FILE('PRINT') RIDFLD(RIDF IN COMMAREA)
          RESP(RESPONSE) END-EXEC.
*
*    CHECK RESPONSES
*
     IF RESPONSE = DFHRESP(NOTFND) THEN
        MOVE 'END OF FILE - PLEASE RESTART' TO MESSAGES
        GO TO MENU.
     IF RESPONSE NOT = DFHRESP(NORMAL) THEN GO TO ERRORS.
*
*    READ AND DISCARD THE RECORD POINTED TO BY RIDF ONLY IF
*    THE LOW END OF THE FILE HAS NOT BEEN REACHED
*
     IF RIDF IN COMMAREA = 'ØØØØ' THEN GO TO NEXT-LINE.
*
*    AN EXTRA CALL TO THE READ-NEXT ROUTINE IS NEEDED WHEN THE
*    PROGRAM ATTEMPTS TO BROWSE FORWARD FROM A POSITION THAT IS
*    NOT AT THE LOW END OF THE FILE. THE READ-NEXT ROUTINE READS
*    THE NEXT RECORD FROM THE FILE.
*
     PERFORM READ-NEXT.
 NEXT-LINE.
*
*    THE ROUTINE READ-NEXT READS THE FIRST RECORD INTO THE FILE
*    AREA
*
     PERFORM READ-NEXT.
*
*    CHECK RESPONSES
*
```

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

*Giselher Wieland (Germany)*

## Determining the library using PINQPGM

CICS systems programmers and application programmers often require a method for determining from which library in the DFHRPL concatenation a program is loaded.

Without a transaction to do this in the required CICS region, the programmer is forced to:

1 Get the library concatenation from JCL.

2 Browse each library in the concatenation to find the first occurrence of the program.

In addition to locating programs in DFHRPL, systems programmers also need a facility to determine which library in the STEPLIB concatenation certain modules are loaded from, whether it be CICS authorized modules, LE/370 modules, or vendor code.

This can be a time-consuming process and prone to error, even at shops where the CICS JCL information is readily available to the programmers. However, some shops limit the access to JOBLIB/ PROCLIB where the CICS region JCL resides, or have SDSF restrictions which may prevent programmers from finding the concatenations in the SDSF log. The time-consuming task of locating programs for application programmers then becomes a burden for the systems programmer.

PINQPGM was written with this in mind. It is a simple command-level CICS BAL program, assigned to a transaction-id of the systems programmer's choosing; it does not need a BMS map. The macro libraries needed are:

- CICS.SDFHMAC
- SYS1.MACLIB
- SYS1.AMODGEN.

PINQPGM receives the input from screen, an eight-byte program name. This name can be any load module or alias name for a load module in the DFHRPL or STEPLIB concatenation.

Firstly, PINQPGM locates the TCB where the DEB for DFHRPL can be located. Together with the DEB addressability comes DFHRPL's DCB address and the offset into TIOA for DFHRPL's dataset allocation information. A BLDL is done for the program name against DFHRPL's DCB, and, if the locate is successful, we get the DSNAME for the library from TIOA, using the concatenation number returned from the BLDL.

We then chain to the initiator TCB, which has the DEB for the STEPLIB. In this case the DDNAME field will contain X'0000000000000000' instead of C'STEPLIB' because STEPLIB is opened by the initiator. When the DEB is found, PINQPGM again issues a BLDL, getting the DSNAME from TIOA once the BLDL is successful, and the screen is displayed. The transaction is terminated by a CLEAR or PF3/15 key.

The sample JCL and source code for PINQPGM follows:

```
//TRN       EXEC PGM=DFHEAP1$,
//          REGION=4096K
//STEPLIB   DD DSN=CICS.REL41.SDFHLOAD,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSPUNCH  DD DSN=&&SYSCIN,
//          DISP=(,PASS),UNIT=SYSALLDA,
//          DCB=BLKSIZE=400,
//          SPACE=(400,(400,100))
//SYSIN     DD *

*****************************************************************
*   WRITTEN BY CHORNG S (JACK) HWANG                            *
*****************************************************************

         PRINT NOGEN
         TITLE 'PINQPGM - FIND DFHRPL FOR PROGRAM'
NEWLINE  EQU   X'15'
STFIELD  EQU   X'1D'
         COPY  DFHAID
         COPY  DFHBMSCA
         DCBD  DSORG=PO,DEVD=DA
         IEFTIOT1
         IEZDEB
         IHAPSA
         IKJTCB
*
DFHEISTG DSECT
HEADERA  DS    CL5
```

```
HEADERT  DS    CL8
         DS    CL24
HEADERC  DS    CL6
HEADERS  DS    CL4
         DS    CL24
HEADERD  DS    CL8
HEADERNL DS    CL2
PGMNAMCA DS    CL5
PGMNAMC  DS    CL8'PGMNAME:'
PGMNAMA  DS    CL2
PGMNAM   DS    CL8
PGMNAMEA DS    CL2
HEADERLE EQU   *-HEADERA
CURSOR   DS    H
RECVLEN  DS    H
TEXTLEN  DS    H
TEXTPTR  DS    F
*
ABSTIME  DS    D
DSNAME   DS    CL44
CONCAT   DS    CL4
DDNAME   DS    CL8
BALSAVE  DS    F
TCBSAVE  DS    F
*
BLDLAREA DS    CL2Ø
REGSTORE DS    16F
MVSREGSA DS    18F
*
TEXTOUT  DS    CL256
*
*  REGISTER USAGE TABLE
*
*    RØ   WORK REG
*    R1   WORK REG
*    R2   WORK REG
*    R3   BASE REG FOR CODE
*    R4   BASE REG FOR CODE
*    R5   WORK REG
*    R1Ø  BASE REG FOR RECEIVED DATA
*    R11  BASE REG FOR EIB
*    R12  BASE REG FOR WORKAREA
*    R13  MVS SAVE AREA
*

PINQPGM  DFHEIENT CODEREG=(3,4),DATAREG=(12)
         CLI   EIBAID,DFHCLEAR      IS THIS CLEAR?
         BE    RETURN               YES, RETURN AND END
         CLI   EIBAID,DFHPF3        PF3?
```

```
         BE    RETURN               YES, RETURN AND END
         CLI   EIBAID,DFHPF15       PF3?
         BE    RETURN               YES, RETURN AND END
         OC    PGMNAM,=CL8' '       CLEAR PGMNAM
         EXEC CICS RECEIVE SET(1Ø) LENGTH(TEXTLEN)
         CLC   Ø(4,1Ø),EIBTRNID     IS THIS UNFORMATTED?
         BE    SENDINIT             YES, GO SEND INITIAL
         LH    2,TEXTLEN            GET LENGTH OF TEXT
         SH    2,=H'3'              SUBTRACT 3 TO BYPASS FIRST SA
         BNP   DOPGMNAM             NOT > Ø, GO DO PROCESS
         LA    1,PGMNAM             GET STARTING ADDRESS OF PGMNAM
         LA    1Ø,3(1Ø)             BUMP PAST SA
PGMNAML  DS    ØH
         MVC   Ø(1,1),Ø(1Ø)         MOVE IN PGMNAM
         LA    1,1(1)               GO TO NEXT BYTE TO MOVE TO
         LA    1Ø,1(1Ø)             GO TO NEXT BYTE TO MOVE FROM
         BCT   2,PGMNAML            GO DO NEXT BYTE
         B     DOPGMNAM             GO PROCESS
*
* PROCESS PGMNAM FOUND
*
DOPGMNAM DS    ØH
         MVC   TEXTOUT(DSNAMEL),DSNAMES MOVE IN SEND TEXT
*
         MVC   DDNAME,=CL8'DFHRPL' GET DFHRPL GUY FIRST
         BAL   1,PROCESSØ
         LA    1Ø,TEXTOUT           GET ADDRESS OF OUTPUT AREA
         MVC   DFHRPLO-DSNAMES(L'DFHRPLO,1Ø),DSNAME MOVE DSNAME
         MVC   CONCATDO-DSNAMES(L'CONCATDO,1Ø),CONCAT MOVE CONCAT #
*
*        MVC   DDNAME,=CL8'STEPLIB' NOW GET STEPLIB GUY
         MVC   DDNAME,=XL8'ØØØØØØØØØØØØØØØØ' NOW GET STEPLIB GUY
         BAL   1,PROCESSØ
         LA    1Ø,TEXTOUT           GET ADDRESS OF OUTPUT AREA
         MVC   STEPLIBO-DSNAMES(L'STEPLIBO,1Ø),DSNAME MOVE DSNAME
         MVC   CONCATSO-DSNAMES(L'CONCATSO,1Ø),CONCAT MOVE CONCAT #
*
         MVC   TEXTLEN,=AL2(DSNAMEL) MOVE SEND LENGTH
         B     PROCESS2             GO SEND IT
*
PROCESSØ DS    ØH
         ST    1,BALSAVE            STORE RETURN ADDRESS
         MVC   CONCAT,=CL4' '
DDNLOOP  DS    ØH
         USING PSA,Ø
         L     1,PSATOLD            GET TCB'S ADDRESS
         USING TCB,1
TCBLOOP  DS    ØH
         ST    1,TCBSAVE
```

```
         SR    2,2                CLEAR R2
         ICM   2,15,TCBDEB        GET FIRST DEB ADDRESS
         BZ    NORPL              INDICATE DFHRPL NOT FOUND
         L     5,TCBTIO           GET TIOT ADDRESS
         DROP  1
         USING DEBBASIC,2
DEBLOOP  DS    ØH
         SR    1,1                CLEAR 1
         ICM   1,7,DEBDCBB        GET DCB ADDRESS
         BZ    NEXTDEB            ZERO, GO GET NEXT DEB
         USING IHADCB,1
         LH    1Ø,DCBTIOT         GET OFFSET INTO TIOT FOR THIS ENTRY
         AR    1Ø,5               GET TRUE TIOT ENTRY
         USING TIOENTRY,1Ø
         CLC   TIOEDDNM,DDNAME    DDNAME FOUND?
         BE    PROCESS
*        MVC   CSHWTO+2Ø(8),TIOEDDNM
*SHWTO   WTO   'PINQPGM                    '
*        EXEC  CICS DELAY
NEXTDEB  DS    ØH
         SR    1,1                CLEAR 1
         ICM   1,7,DEBDEBB        GET NEXT DEB ADDRESS
         BZ    NORPL              INDICATE DFHRPL NOT FOUND
         LR    2,1                GET DEB ADDRESS
         B     DEBLOOP            GO GET'EM TIGER
         DROP  1,2
*
NORPL    DS    ØH
*        WTO   'PINQPGM - GOING TO NEXT TCB'
*        EXEC  CICS DELAY
         L     2,TCBSAVE          GET TCB'S ADDRESS
         USING TCB,2
         SR    1,1
         ICM   1,15,TCBBACK       GET NEXT TCB
         DROP  2
         BZ    TCBLOOPD           NO, CONTINUE TO PROCESS
         C     1,PSATOLD          SEE IF WE'VE HIT END
         BNE   TCBLOOP
TCBLOOPD DS    ØH
         MVC   DSNAME,=CL44'DCB NOT FOUND   '
         B     PROCESS1
*
*
*
PROCESS  DS    ØH
         STM   Ø,15,REGSTORE      STORE REGISTERS
         LA    13,MVSREGSA        GET ADDRESS OF MVS SA
         MVC   BLDLAREA(2),=H'1'  INDICATE 1 ENTRY
         MVC   BLDLAREA+2(2),=H'14' 14 BYTE ENTRY
```

```
         MVC   BLDLAREA+4(8),PGMNAM MOVE IN PROGRAM NAME
         BLDL  (1),BLDLAREA       GO DO BLDL
         LM    Ø,14,REGSTORE      STORE REGISTER
         LTR   15,15              TEST 15
         BNZ   NOMEMBER           NOT FOUND
*
         USING IHADCB,1
         LH    1Ø,DCBTIOT         GET OFFSET INTO TIOT FOR THIS ENTRY
         DROP  1
         L     1,PSATOLD
         USING TCB,1
         L     5,TCBTIO           GET TIOT ADDRESS
         AR    1Ø,5               GET TRUE TIOT ENTRY
         DROP  1
*
         SR    1,1                CLEAR 1
         ICM   1,1,BLDLAREA+15    GET CONCATENATION NUMBER
         CVD   1,ABSTIME          CONVERT TO DECIMAL
         UNPK  CONCAT+1(3),ABSTIME+6(2) UNPACK
         OI    CONCAT+3,C'Ø'      FORCE X'FØ'
         MVI   CONCAT,C'+'
DSNAMELP DS    ØH
         CH    1,=H'1'            COMPARE WITH H'1'
         BL    DSNFOUND           LOW, FOUND DSNAME
         BCTR  1,Ø                SUBTRACT COUNT BY ONE
         SR    Ø,Ø                CLEAR RØ
         IC    Ø,TIOELNGH         GET TIOE LENGTH
         AR    1Ø,Ø               BUMP UP TO NEXT TIOT ENTRY
         B     DSNAMELP
DSNFOUND DS    ØH
         SR    1,1                CLEAR 1
         ICM   1,7,TIOEJFCB       GET JFCB ADDRESS
         MVC   DSNAME,16(1)       MOVE IN DSNAME
         B     PROCESS1
*
NOMEMBER DS    ØH
         MVC   DSNAME,=CL44'PROGRAM NOT FOUND IN CONCATENATION'
*
PROCESS1 DS    ØH
         L     1,BALSAVE            GET RETURN ADDRESS
         BR    1                    RETURN
*
PROCESS2 DS    ØH
         EXEC CICS SEND TEXT FROM(TEXTOUT) LENGTH(TEXTLEN) ERASE
*
*
*
SENDINIT DS    ØH
         XC    TEXTLEN,TEXTLEN      CLEAR TEXT LENGTH
```

```
         LA    Ø,TEXTOUT            GET ADDRESS OF OUTPUT TEXT
         ST    Ø,TEXTPTR            STORE ADDRESS OF OUTPUT TEXT
         MVI   HEADERA,STFIELD      MOVE IN START FIELD
         MVI   HEADERA+1,DFHBMASK   MOVE IN ASKIP
         MVI   HEADERA+2,DFHSA      MOVE IN SET ATTRIBUTE
         MVI   HEADERA+3,DFHCOLOR   MOVE IN COLOR
         MVI   HEADERA+4,DFHTURQ    MOVE IN COLOR TURQUOISE
         MVC   HEADERC,=CL6'SYSID=' INDICATE SYSID
         EXEC CICS ASSIGN SYSID(HEADERS)
         EXEC CICS ASKTIME ABSTIME(ABSTIME)
         EXEC CICS FORMATTIME ABSTIME(ABSTIME)
               TIME(HEADERT) TIMESEP MMDDYY(HEADERD) DATESEP
         MVI   HEADERNL,NEWLINE     MOVE NEW LINE AFTER LINE1
         MVI   HEADERNL+1,NEWLINE   MOVE NEW LINE AFTER LINE1
         MVC   PGMNAMCA,HEADERA     MOVE IN DEFAULT DISPLAY ATTRIBUTE
         MVC   PGMNAMC,=CL8'PGNNAME:'
  MVI   PGMNAMA,STFIELD      MOVE IN START FIELD
  MVC   PGMNAMA+1(1),=AL1(DFHBMUNP+DFHBMFSE+DFHBMBRY)
  MVI   PGMNAMEA,STFIELD     MOVE IN START FIELD
         MVI   PGMNAMEA+1,DFHBMASK  MOVE IN ASKIP
 LH    1,TEXTLEN            GET TEXT LENGTH
 LA    1,HEADERLE(1)        ADD LENGTH OF HEADER
 STH   1,TEXTLEN            STORE TEXT LENGTH
 L     1,TEXTPTR            GET OUTPUT LOCATION
 MVC   Ø(HEADERLE,1),HEADERA MOVE OUTPUT LINE
 LA    1,HEADERLE(1)        BUMP UP MVC LENGTH
         ST    1,TEXTPTR
SENDTEXT DS    ØH
         EXEC CICS SEND TEXT FROM(TEXTOUT) LENGTH(TEXTLEN)        X
               FREEKB CURSOR(=AL2(171))
RETURNX  DS    ØH
         EXEC CICS RETURN TRANSID(EIBTRNID)
RETURN   DS    ØH
         EXEC CICS SEND CONTROL ERASE FREEKB
         EXEC CICS RETURN
*
*
*
DSNAMES  DC     XL6'151515151515'
         DC    AL1(STFIELD,DFHBMASK,DFHSA,DFHCOLOR,DFHTURQ)
         DC    C' DFHRPL: '
DFHRPLO  DS    CL44
         DC    C'   CONCAT: '
CONCATDO DS    CL4
         DC    XL6'1515'
         DC    AL1(STFIELD,DFHBMASK,DFHSA,DFHCOLOR,DFHTURQ)
         DC    C'STEPLIB: '
STEPLIBO DS    CL44
         DC    C'   CONCAT: '
```

```
CONCATSO DS    CL4
DSNAMEL  EQU   *-DSNAMES
*
         END

/*
//ASM       EXEC PGM=IEV90,
//          REGION=4096K,
//          PARM='NODECK,OBJECT,XREF(SHORT)'
//SYSLIB    DD DSN=CICS.REL41.SDFHMAC,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DISP=SHR,DSN=SYS1.AMODGEN
//SYSUT1    DD UNIT=SYSALLDA,SPACE=(1700,(400,400))
//SYSUT2    DD UNIT=SYSALLDA,SPACE=(1700,(400,400))
//SYSUT3    DD UNIT=SYSALLDA,SPACE=(1700,(400,400))
//SYSLIN    DD DSN=&&LOADSET,
//          UNIT=SYSALLDA,DISP=(,PASS),
//          SPACE=(400,(100,100,1))
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DSN=&&SYSCIN,DISP=(OLD,DELETE)
//COPYLINK  EXEC PGM=IEBGENER,COND=(7,LT,ASM)
//SYSUT1    DD DSN=CICS.REL41.SDFHMAC(DFHEILIA),DISP=SHR
//SYSUT2    DD DSN=&&COPYLINK,DISP=(NEW,PASS),
//          DCB=(LRECL=80,BLKSIZE=400,RECFM=FB),
//          UNIT=SYSALLDA,SPACE=(400,(20,20))
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//LKED      EXEC PGM=IEWL,REGION=4096K,
//          PARM='LIST,XREF',COND=(7,LT,ASM)
//SYSLIB    DD DSN=CICS.REL41.SDFHLOAD,DISP=SHR
//SYSLMOD   DD DISP=SHR,DSN=CICS.TESTR41.LOADLIB(PINQPGM)
//SYSUT1    DD UNIT=SYSALLDA,DCB=BLKSIZE=1024,
//          SPACE=(1024,(200,20))
//SYSPRINT  DD SYSOUT=*
//SYSLIN    DD DSN=&&COPYLINK,DISP=(OLD,DELETE)
//          DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD DDNAME=SYSIN
//*
//*
```

*Chorng S (Jack) Hwang*
*Principal*
*HSA Systems (USA)*

# SAS CICS workload manager

We currently run CICS Version 3 under OS/390 and, like many other CICS users, we have Landmark's 'The Monitor for CICS' product. The Monitor has a special language and we find that using SAS/CPE from MONITOR dump files is more user-friendly. Furthermore, SAS/CPE can be used for auditing not only CICS but also RMF, RACF, JES2, DHSMS, DFHRMM, and VTAM.

One of the main failings of The Monitor is the lack of a report on the workload by hour. A feature of the program presented here is that, thanks to SAS/CPE, it not only does reports, but can also aggregate information by week, by month, or by year.

## CICS WORKLOAD MEASUREMENT

At present, using some of the software products that run on IBM systems, the CICS subsystem can be measured for service objectives such as response time, and for resource use. Figure 1 shows some of the resource measures available. Response time, for example, would be measured as TRCCPU plus TRCWAIT.

The TYPETMON member will process the data records created by The Monitor. Records are created in a VSAM file and then dumped with the TMV608 program supplied by Landmark. Do not use IDCAMS to copy the VSAM file; only TMV608 will work. TMV608 will create either an uncompressed format (which can be read directly with TYPEMONI) or a compressed format (which strips out repeated nulls and thereby reduces the disk requirements for The Monitor's VSAM file).

## RESPONSE AND RESOURCE MEASUREMENTS

The effects of summarization are obvious for time and numerical values, but not so obvious for other data types such as character, time stamps, and flag bytes. SAS/CPE uses summarization by day, by week, by month, and by year.

Of the reports, RJMONI gives the transaction frequency by hour, and

| *Name* | *Resource measured in transaction record* |
|---|---|
| TRCCPU | CPU time as perceived by CICS |
| TRCCPUR | Actual ‘real’ CPU time of transaction |
| TRCWAIT | Wait time as perceived by CICS |
| TRCSIO | Count of start I/Os issued by transaction |
| TRCPGIN | Count of page-ins |
| TRCPGOUT | Count of page-outs |
| TRCMSGIN | Number of input messages |
| TRCMSGOT | Number of output messages |
| TRCMILTH | Total length of input messages |
| TRCMOLTH | Total length of output messages |
| TRCHISTG | Highwater mark of terminal + user address space |
| TRCPCPH | Highwater mark of program address space |

*Figure 1: CICS resource measures*

UPLYON gives the time of each first transaction. EXPSMON, EXPSMON1, EXPSMON2, EXPSMON3, EXPSMON4, EXPSMON5, and EXPSMON6 are useful jobs with which to audit your CICS.

TYPETMON

```
/*  TYPETMON PROCESSES RECORDS WRITTEN BY LANDMARK SYSTEMS CORP     */
/*  PRODUCT «THE MONITOR FOR CICS/ESA» VERSION 1.3 AND LATER.      */
/*  RECORDS FROM THEIR VERSION 1.3 ARE INCOMPATIBLE WITH PRIOR     */
/*  VERSIONS, AND THUS TYPETMON MUST BE USED INSTEAD OF TYPEMON8.  */
/*                                                                 */
```

```
/* Extraction controller from LUname */
data monit / view=monit ;
     set detail.monitas (where=(jour="&HIER" and luname not =:'ØØ'X));
     pu = substr(luname,2,5);
run;

title "Number of transactions per controller and CICS of &hier";
proc chart data=monit ;
     label pu     ='Controleur'
           sysid  ='CICS';
     format cretime tod2.;
     hbar sysid /
        group=pu
        freq
        ;
run;

title "Number of transactions per hour and per CICS of &hier";
proc chart data=detail.monitas(where=(jour="&hier"));
     label cretime='Heure'
           sysid  ='CICS';
     format cretime tod2.;
     hbar SYSID /
        group=cretime
        freq cpercent
        ;
run;

/* Add hour to Monitas  */
data monit / view=monit ;
     set detail.monitas (where=(jour="&HIER"));
     heure=timepart(cretime);
run;

title "Number of transactions per CICS and per hour of &hier";
proc chart data=monit;
     by sysid ;
     label sysid = 'CICS';
     vbar heure   /
          midpoints =
 'ØØ:ØØ't 'Ø1:ØØ't 'Ø2:ØØ't 'Ø3:ØØ't 'Ø4:ØØ't 'Ø5:ØØ't 'Ø6:ØØ't
 'Ø7:ØØ't 'Ø8:ØØ't 'Ø9:ØØ't '1Ø:ØØ't '11:ØØ't '12:ØØ't '13:ØØ't
 '14:ØØ't '15:ØØ't '16:ØØ't '17:ØØ't '18:ØØ't '19:ØØ't '2Ø:ØØ't
 '21:ØØ't '22:ØØ't '23:ØØ't '24:ØØ't
        ;
     format heure  hour2.;
run;
title;
```

```
proc tabulate data=detail.monitas(where=(jour="&hier"))
     format=f12.0;
     class sysid appli ;
     var tacputm filecn ;
     keylabel sum=' ' ;
     table sysid=' ' * ( appli=' ' all='Total CICS')
              all='Total journee',
           (tacputm='CPU'*f=time12.2
            filecn='Acces fichier')*sum n='Nombre'
           / box="Activite CICS du &HIER"
           ;
run;

proc tabulate data=detail.monitas(where=(jour="&hier"));
     class sysid cretime;
     format cretime tod2.;
     table cretime=' '
               *(sysid=' ' all='Total heure')
                all='Total jour' ,
           n='Nb transactions'*f=12.0
           /rts=45 box="D{bit transactionnel du &HIER" ;
run;

title "Distribution of response times for CICS of &hier";

proc chart data=detail.monitas(where=(jour="&hier"));
     by sysid ;
     vbar  reponse
     / midpoints =
 '00:00:00.05't '00:00:00.10't '00:00:00.15't '00:00:00.20't
 '00:00:00.25't '00:00:00.30't '00:00:00.35't '00.00:00.40't
 '00:00:00.45't
     ;
run;
title ;
```

## EXPSMON

```
//EXPSMON JOB EXP10,SYSTEM,CLASS=R,MSGCLASS=T
//*************************************************************
//*   REPORT : CICS ACTIVITIES
//*************************************************************
//DIVMON EXEC SAS,REGION=8M,
//       WORK='200,150',
//         OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL',
//       SORT=10
//REF1     OUTPUT PAGEDEF=CHRP,FORMDEF=CHRP,COPIES=01
//DAY      DD DSN=SAS.BERCY.CICPDB.DAY,DISP=SHR
```

```
//DETAIL   DD DSN=SAS.BERCY.CICPDB.DETAIL,DISP=SHR
//MONTH    DD DSN=SAS.BERCY.CICPDB.MONTH,DISP=SHR
//LIBRARY  DD DSN=SAS.MXG.V1313.FORMATS,DISP=SHR
//REPORT   DD DSN=SAS.LOCAL.REPORTS,DISP=SHR
//         DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//SASLIST DD SYSOUT=L,OUTPUT=*.REF1
//SYSIN  DD *

  OPTIONS PAGESIZE=60 LINESIZE=132 ;

                  /****  DAILY REPORTS *****/

   %INCLUDE REPORT(OPTIONS);
   %INCLUDE REPORT(HIER);
   %INCLUDE REPORT(RJMONI);
/*
```

## EXPSMON1

```
//EXPSMON1 JOB EXP10,SYSTEM,CLASS=R,MSGCLASS=T
//*************************************************************
//*   REPORT : TMON : NUMBER OF TRANSACTIONS PER CONTROLLER AND
//*                   CICS.
//*************************************************************
//DIVMON1 EXEC SAS,REGION=8M,
//       WORK='200,150',
//         OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL',
//       SORT=10
//REF1     OUTPUT PAGEDEF=CHRP,FORMDEF=CHRP,COPIES=01
//DAY      DD DSN=SAS.BERCY.CICPDB.DAY,DISP=SHR
//DETAIL   DD DSN=SAS.BERCY.CICPDB.DETAIL,DISP=SHR
//MONTH    DD DSN=SAS.BERCY.CICPDB.MONTH,DISP=SHR
//LIBRARY  DD DSN=SAS.MXG.V1313.FORMATS,DISP=SHR
//REPORT   DD DSN=SAS.LOCAL.REPORTS,DISP=SHR
//         DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//SASLIST DD SYSOUT=L,OUTPUT=*.REF1
//SYSIN  DD *
  OPTIONS PAGESIZE=60 LINESIZE=132 ;

                  /****  DAILY REPORTS *****/

   %INCLUDE REPORT(OPTIONS);
   %INCLUDE REPORT(HIER);
   %INCLUDE REPORT(RJMONI1);
/*
```

## EXPSMON2

```
//EXPSMON2 JOB EXP1Ø,SYSTEM,CLASS=R,MSGCLASS=T
//*************************************************************
//*   REPORT : TMON : NUMBER OF TRANSACTIONS PER HOUR AND PER
//*                   CICS.
//*************************************************************
//DIVMON2 EXEC SAS,REGION=8M,
//       WORK='2ØØ,15Ø',
//         OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL',
//       SORT=1Ø
//REF1     OUTPUT PAGEDEF=CHRP,FORMDEF=CHRP,COPIES=Ø1
//DAY      DD DSN=SAS.BERCY.CICPDB.DAY,DISP=SHR
//DETAIL   DD DSN=SAS.BERCY.CICPDB.DETAIL,DISP=SHR
//MONTH    DD DSN=SAS.BERCY.CICPDB.MONTH,DISP=SHR
//LIBRARY  DD DSN=SAS.MXG.V1313.FORMATS,DISP=SHR
//REPORT   DD DSN=SAS.LOCAL.REPORTS,DISP=SHR
//         DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//SASLIST DD SYSOUT=L,OUTPUT=*.REF1
//SYSIN  DD *

  OPTIONS PAGESIZE=6Ø LINESIZE=132 ;

                  /****  DAILY REPORTS *****/

   %INCLUDE REPORT(OPTIONS);
   %INCLUDE REPORT(HIER);
   %INCLUDE REPORT(RJMONI2);
/*
```

## EXPSMON3

```
//EXPSMON3 JOB EXP1Ø,SYSTEM,CLASS=R,MSGCLASS=T
//*************************************************************
//*   REPORT : TMON : NUMBER OF TRANSACTIONS PER CICS AND PER
//*                   HOUR
//*************************************************************
//DIVMON3 EXEC SAS,REGION=8M,
//       WORK='2ØØ,15Ø',
//         OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL',
//       SORT=1Ø
//REF1     OUTPUT PAGEDEF=CHRP,FORMDEF=CHRP,COPIES=Ø1
//DAY      DD DSN=SAS.BERCY.CICPDB.DAY,DISP=SHR
//DETAIL   DD DSN=SAS.BERCY.CICPDB.DETAIL,DISP=SHR
//MONTH    DD DSN=SAS.BERCY.CICPDB.MONTH,DISP=SHR
//LIBRARY  DD DSN=SAS.MXG.V1313.FORMATS,DISP=SHR
//REPORT   DD DSN=SAS.LOCAL.REPORTS,DISP=SHR
//         DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//SASLIST DD SYSOUT=L,OUTPUT=*.REF1
```

```
//SYSIN  DD *

  OPTIONS PAGESIZE=60 LINESIZE=132 ;

                  /****  DAILY REPORTS *****/

   %INCLUDE REPORT(OPTIONS);
   %INCLUDE REPORT(HIER);
   %INCLUDE REPORT(RJMONI3);
/*
```

## EXPSMON4

```
//EXPSMON4 JOB EXP10,SYSTEM,CLASS=R,MSGCLASS=T
//*************************************************************
//*   REPORT : TMON : CICS ACTIVITY
//*************************************************************
//DIVMON4 EXEC SAS,REGION=8M,
//       WORK='200,150',
//         OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL',
//       SORT=10
//REF1     OUTPUT PAGEDEF=CHRP,FORMDEF=CHRP,COPIES=01
//DAY      DD DSN=SAS.BERCY.CICPDB.DAY,DISP=SHR
//DETAIL   DD DSN=SAS.BERCY.CICPDB.DETAIL,DISP=SHR
//MONTH    DD DSN=SAS.BERCY.CICPDB.MONTH,DISP=SHR
//LIBRARY  DD DSN=SAS.MXG.V1313.FORMATS,DISP=SHR
//REPORT   DD DSN=SAS.LOCAL.REPORTS,DISP=SHR
//         DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//SASLIST DD SYSOUT=L,OUTPUT=*.REF1
//SYSIN  DD *

  OPTIONS PAGESIZE=60 LINESIZE=132 ;

                  /****  DAILY REPORTS *****/

   %INCLUDE REPORT(OPTIONS);
   %INCLUDE REPORT(HIER);
   %INCLUDE REPORT(RJMONI4);
/*
```

## EXPSMON5

```
//EXPSMON5 JOB EXP10,SYSTEM,CLASS=R,MSGCLASS=T
//*************************************************************
//*   REPORT : TMON : DEBIT TRANSACTIONS
//*************************************************************
//DIVMON5 EXEC SAS,REGION=8M,
//       WORK='200,150',
```

```
//         OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL',
//       SORT=1Ø
//REF1     OUTPUT PAGEDEF=CHRP,FORMDEF=CHRP,COPIES=Ø1
//DAY      DD DSN=SAS.BERCY.CICPDB.DAY,DISP=SHR
//DETAIL   DD DSN=SAS.BERCY.CICPDB.DETAIL,DISP=SHR
//MONTH    DD DSN=SAS.BERCY.CICPDB.MONTH,DISP=SHR
//LIBRARY  DD DSN=SAS.MXG.V1313.FORMATS,DISP=SHR
//REPORT   DD DSN=SAS.LOCAL.REPORTS,DISP=SHR
//         DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//SASLIST DD SYSOUT=L,OUTPUT=*.REF1
//SYSIN  DD *

  OPTIONS PAGESIZE=6Ø LINESIZE=132 ;

                  /****  DAILY REPORTS *****/

   %INCLUDE REPORT(OPTIONS);
   %INCLUDE REPORT(HIER);
   %INCLUDE REPORT(RJMONI5);
/*
```

## EXPSMON6

```
//EXPSMON6 JOB EXP1Ø,SYSTEM,CLASS=R,MSGCLASS=T
//*************************************************************
//*   REPORT : TMON : RESPONSE TIME DISTRIBUTION FOR CICS       *
//*************************************************************
//DIVMON6 EXEC SAS,REGION=8M,
//       WORK='2ØØ,15Ø',
//         OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL',
//       SORT=1Ø
//REF1     OUTPUT PAGEDEF=CHRP,FORMDEF=CHRP,COPIES=Ø1
//DAY      DD DSN=SAS.BERCY.CICPDB.DAY,DISP=SHR
//DETAIL   DD DSN=SAS.BERCY.CICPDB.DETAIL,DISP=SHR
//MONTH    DD DSN=SAS.BERCY.CICPDB.MONTH,DISP=SHR
//LIBRARY  DD DSN=SAS.MXG.V1313.FORMATS,DISP=SHR
//REPORT   DD DSN=SAS.LOCAL.REPORTS,DISP=SHR
//         DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//SASLIST DD SYSOUT=L,OUTPUT=*.REF1
//SYSIN  DD *

  OPTIONS PAGESIZE=6Ø LINESIZE=132 ;

                  /****  DAILY REPORTS *****/

   %INCLUDE REPORT(OPTIONS);
   %INCLUDE REPORT(HIER);
   %INCLUDE REPORT(RJMONI6);
/*
```

## RANCICS

```
/*********************************************************************/
/* Report on CICS transactions                                       */
/*                                                                   */
/* Base: SAS.BERCY.CICPDB.MONTH                                      */
/*                                                                   */
/* Object: Display CICS transactions by year and month.              */
/*                                                                   */
/* Pre-requis: OPTIONS                                               */
/*********************************************************************/
options nocenter nodate ls=132 ps=56 pageno=1;

/* Calcul de l annee precedente                    */
data _null_ ;
     annee=put(mdy(1,1,year(today())) - 1,year4.);
     call symput ("ANNEE",annee);
run;

title ;
title «  Number of transactions per cics for &annee «;
libname month "sas.bercy.cicpdb.month" disp=shr;
proc tabulate data=month.monitas  missing format=commax15.Ø
           order=data;
           where annee = «&annee»;
     class appli sysid annee  ;
     var  filecn_s nbtranss      ;
     keylabel sum=' " ;
     table sysid =' " all="Total'
          , filecn_s='Nb acces fichiers'
            nbtranss='Nb transactions'
            / box=_page_
             rts=35
            ;
run;

title ;
title «  Number of transactions per month per cics for &annee»;
libname month "sas.bercy.cicpdb.month" disp=shr;
proc tabulate data=month.monitas  missing format=commax15.Ø
           order=data;
           where annee = «&annee»;
     class appli sysid mois   ;
     var  filecn_s nbtranss      ;
     keylabel sum=' " ;
     table mois, sysid =' " all="Total'
          , filecn_s='Nb acces fichiers'
            nbtranss='Nb transactions'
            / box=_page_
             rts=35
```

```
            ;
run;

title ;
title « Number of transactions per month per application for &annee»;
proc tabulate data=month.monitas  missing format=commax15.Ø
          order=data;
           where annee = «&annee»;
     class appli sysid mois   ;
     var  filecn_s nbtranss      ;
     keylabel sum=' " ;
     table mois,  appli=' " all="Total'
          , filecn_s='Nb acces fichiers'
            nbtranss='Nb transactions'
            / box=_page_
             rts=35
            ;
run;
```

## SASJTMON

```
//SASJTMON JOB COM,'SASTMON',CLASS=K,MSGCLASS=9,MSGLEVEL=(1,1)
//************************************************
//DELCPTMO EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE EXPL69.CPETMON
 IF MAXCC <= 8 THEN SET MAXCC=Ø
/*
//************************************************
//COPIE    EXEC PGM=ICEGENER,TIME=2Ø
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=TMON.V1R3.TMONØ1.ARCHIVE9,DISP=SHR       TMON3DLS
//SYSUT2   DD DSN=EXPL69.CPETMON,DISP=(,CATLG,DELETE),UNIT=SYSDA,
//         DCB=(RECFM=VB,LRECL=23472,BLKSIZE=23476),
//         SPACE=(CYL,(3ØØ,15Ø),RLSE)
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD DUMMY
/*
//DELETE   EXEC PGM=IDCAMS,COND=(Ø,NE,COPIE)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE TMON.V1R3.TMONØ1.ARCHIVE9.* NONVSAM PURGE
 IF MAXCC <= 8 THEN SET MAXCC=Ø
/*
//LOAD     EXEC SAS,REGION=8M,TIME=144Ø,
//       WORK='15Ø,25',
//         OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL SORT=1Ø'
//* REF1     OUTPUT PAGEDEF=CHRP,FORMDEF=CHRP,COPIES=Ø1
```

```
//MONICICS DD DSN=EXPL69.CPETMON,DISP=SHR
//REPORT   DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//OUTCIC   DD DSN=SAS.BERCY.CICDAY.APPLI,DISP=OLD
//SASLIST  DD SYSOUT=O
//* SASLIST  DD SYSOUT=O,OUTPUT=*.REF1
//SYSIN    DD *

  /* APPLICATION LOADING CPE */

  OPTIONS PAGESIZE=6Ø LINESIZE=132 ;

  %CPSTART(MODE=BATCH,
         SYSTEM=MVS,
           ROOT=SAS.SAS6Ø9.TS45Ø.CPE.,
           PDB=SAS.BERCY.CICPDB.,
           DISP=OLD,
           ROOTSERV=,
           SHARE=N/A,
           MXGSRC=("SAS.BERCY.SOURCLIB" "SAS.MXG.V1313.SOURCLIB"),    ),
           MXGLIB=SAS.MXG.V1313.FORMATS
          ) ;

  /* READING TMON  FILES */
  %INCLUDE SOURCLIB(TYPEMON8);
  RUN;

  /* LOADING BASE PERFORMANCE-LEVEL DETAILS */
  %CMPROCES(,MONITAS,
            COLLECTR=GENERIC,
            TOOLNM=SASDS,
            UNIT=DISK,
            GENLIB=WORK
           );

  /* LOADING BASE HISTORICAL PERFORMANCE DETAILS  */
  %CPREDUCE(MONITAS);

  /* DAILY REPORTS */

     %INCLUDE REPORT(OPTIONS);
     %INCLUDE REPORT(HIER);
     %INCLUDE REPORT(RJMONI);

  /* LOADING DETAILS FOR PC APPLICATIONS */
  %INCLUDE REPORT(HIER);
  %INCLUDE REPORT(AJMONI);
/*
```

```
/*****************************************************/
/*** nom du pg: monitpg              SALVADOR      ***/
/***                                               ***/
/***      MEASURES LOADS TRANSACTIONS              ***/
/***                                               ***/
/***                                               ***/
/*****************************************************/
/*  options + affichage courier new 9                */
options nofmterr ls = 100 ps= 80 pageno = 1;

  data _null_;
/* RETRIEVING TODAY'S DATE                           */
    tjour = «&sysdate»d ;
/* FOR EXAMPLE IF MONDAY REQUESTED                   */
/* FOR SAS, MONDAY IS THE SECOND DAY OF THE WEEK     */
    if weekday(tjour) = 2 then tjour = tjour - 3;
/* IF THE DAY IS A HOLIDAY, THE FOLLOWING DAYS ARE MODIFIED */
    else tjour = tjour - 1;
/* CREATE A MACRO-VARIABLE CONTAINING THE DATE       */
/* OF THE DAY WE'RE DEALING WITH                     */
   call symput(«tjour»,tjour);
  run;
/* INSERT THE RELEVANT DATE IN THE LOG FOR CHECKING  */
   data _null_;
  tjour = &tjour;
  put «REPARTITION DES TRANSACTIONS, TRAITEMENT CONCERNANT LE «
  tjour ddmmyy8.;
   run;

/*****************************************************/
/* MEASURES LOADS TRANSACTIONS                       */

   libname detail «sas.bercy.cicpdb.detail» disp = shr; run;

/* RECOVERING THE DATA FOR THE RELEVANT DAY          */
options nofmterr ls = 100 ps= 80 pageno = 1;
  data monit (keep = mois appli luname program trnsact cretime);
   set detail.monitas;
   where datepart(cretime) =  &tjour;
  run;

/*  SELECTION OF INITIAL LOAD           */
/*  EDIT TABLE AT END OF PROGRAM        */
   proc sort data = monit;
    by mois appli program trnsact cretime;
   run;
   data fmonit;
     set monit;
```

```
     by mois appli program trnsact cretime;
     if first.program and first.trnsact;
     ejour = put(datepart(cretime),ddmmyy8.);
     call symput(«ejour»,ejour);
     heure = timepart(cretime);
   run;

/*****************************************************/
/* CALCULATING FOR THE PREVIOUS 30 DAY PERIOD        */
/* RECOVERING THE DATA FOR THE DAY REQUESTED         */
   data monit2 (keep = mois appli program trnsact jour plage compteur);
     set detail.monitas;
     where datepart(cretime) =  &tjour;
     jour = day(datepart(cretime));
     plage = hour(cretime);
     compteur = 1;
   run;

/* SUMMATION FOR PERIOD TIMETABLE                        */
   proc summary data = monit2 nway ;
   class mois jour plage ;
   var compteur;
   output out = fmonit2 (drop = _freq_ _type_) sum = ;
   run;

/* CREATING THE VARIABLES MDATE AND MDATER FOR ELIMINATION  */
/* AND CREATE THE PERIODS BEFORE EDITTING                   */
   data fmonit2 (drop = i j ii jj mdater);
    set fmonit2;
    mdate = &tjour;
    mdater = &tjour - 3Ø;
    call symput("mdater",mdater);
    length plaghor $5 ii $2 jj $2;
     do i=1 to 24;
       j = i-1;
       if plage = j then do;
           if i < 1Ø then ii = «Ø»||trim(left(i)); else ii= i;
           if j < 1Ø then jj = «Ø»||trim(left(j)); else jj= j;
           plaghor = trim(left(jj))||»-»||trim(left(ii));
        end;
     end;
   run;

/* ARCHIVE THE DATA FOR THE DAY IN THE MONTHLY TABLE  */

   proc append base = sasuser.monit data=fmonit2;
   run;
```

```
   DATA sasuser.monit;
    set sasuser.monit;
    where mdate > &mdater;
   run;
/*  NODUPKEY TO AVOID DUPLICATIONS                    */
   proc sort data=sasuser.monit nodupkey;
   by mdate plaghor;
   run;

/* CREATE DAILY TABLE                                 */
/*  options + affichage                               */
 options nofmterr ls = 100 ps= 80 pageno = 1;

   proc tabulate data= fmonit ;
   class appli program trnsact ;
   var heure ;
   table appli = «APPLICATION»,
         program =»NOM DU PROGRAMME»*
         trnsact = «NOM DE LA TRANSACTION»,
         heure = «HEURE DE LANCEMENT»*sum = « «*f=time10.
   /box = «REFERENCES» ;
   title «MEASURE TIME OF FIRST LOADING OF TRANSACTION &ejour»
   run;

/* CREATE TABLE FOR THE PREVIOUS 30 DAYS              */
/*  options + affichage                               */
 options nofmterr ls = 164 ps= 66 pageno = 1;

   proc tabulate data= sasuser.monit ;
   class mois jour plaghor  ;
   var compteur ;
   table mois = «MOIS»*
         jour =»JOUR»,
         plaghor = «PLAGE HORAIRE»*
         compteur = «NB DE TR»*sum = « «*f=commax8.
   /box = «DATE» rts=20 ;
   title «NUMBER OF TRANSACTIONS LOADED IN THE PREVIOUS 30 DAYS « ;
   title2 « FOR PERIOD TIMETABLE REQUIRED»;
   run;
```

*Claude Dunand (France)* © Xephon 1998

CICS users can benefit from a range of in-house and third-party tools announced by IBM to speed up all stages of the Year 2000 compliance process. The company is expanding its VisualAge 2000 portfolio with Millennium Language Extensions and the new Application Testing Collection, and will also sell Allstate's Millennium Date Compression Tool (MDCT).

MDCT changes customer data while applications are running. Developed by Allstate for its own systems, it is out now for CICS, COBOL, PL/I, and Assembler on OS/390.

Included in the VisualAge 2000 portfolio is Maintenance 2000 Version 2.1, used in the 'find' and 'fix' phase of a Year 2000 project, which supports CICS, DB2, and DL/I applications. The software analyses MVS application programs for two-digit date items.

For further information contact your local IBM representative.

* * *

Neon Systems has announced Affinities Server for CICSplex (ASC), which allows CICS/ESA applications to take true advantage of sysplex parallel processing without making changes to application programs, CICS, or MVS.

Removing affinity dependencies improves processor utilization and transaction response time and throughput. Since transactions can run in any CICS region rather than one specified CICS region, a single point of failure is eliminated.

Eliminating affinities also allows CICS regions to be cloned. With cloning, system management and maintenance are simplified. Additionally, CICS regions can be started and stopped based on workload requirements.

For more information contact:
Neon Systems, 14141 Southwest Freeway, Suite 6200, Sugar Land, TX 77478, USA. Tel: (281) 491 4200.

* * *

IBM has announced CICS Transaction Server for OS/2 Version 4.1 which provides CICS Clients Version 2.0.2 and Lotus Domino Go Webserver, in addition to significant enhancements over the previous version. Enhancements include extended CEDA resource definition, multiple server operation on a single workstation, and host-on-demand connection.

For further information contact your local IBM representative.

* * *