162

CICS update

May 1999

In this issue

CICS 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

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 £175.00 in the UK; $270.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £16.00 ($23.50) each including postage.

Printed in England.

Storing data using VSAM files

When developing on-line applications, special consideration should be taken to minimize the response time to end-user requests in accessing stored data. The data is held in a number of file types, often in index files. Usually, a data element by which data records will be accessed most often is taken as the key field of the file. Sometimes, an entire file may need to be browsed if a data record of another data element is required. This can be avoided by storing data in another index file, with that data element as the key field. However, this can lead to various problems such as data duplication, data integrity, etc.

I have developed a method to store data and to access it with a number of keys in CICS applications. This system uses two files to store data – a VSAM index and a VSAM RRDS file. The user data is held in a RRDS file (data file) and the key value to access the data is stored in an index file (key file). The data records that belong to a key value are linked by a chain of pointers that is stored in both the key file and the data file. Therefore, when retrieving records, only the records that belong to a key will be read and efficient data retrieving can be expected.

The system has several advantages:

- It gives a quick response to end-user requests for data storage and retrieval functions.
- It is easy to use and is cost-effective.
- Programmers need not perform any CICS data manipulation commands – the pointer manipulation is completely hidden from them.
- You can access data records with a number of keys.
- You can access data records that are stored in different files using one key file.
- It is suitable for small- to medium-scale CICS installations, especially where RDBMSs are not available.
- It allows easy maintenance of programs – there is no need to

change user programs when a new key to access records is added.

- Key values can be of varying lengths.
- The key values need not necessarily be part of the record.
- It is possible to hold transaction data, journal data, etc, where data is held for a temporary period.

SYSTEM DETAILS

The data records can be linked to several keys and a key can be linked to different types of record (in different data files). The key and the data record can be either singly or doubly linked. If singly linked, the last record in will be the first one that is read when retrieving records (ie last in, first out). If doubly linked, records will be read in the order of entry (ie first in, first out). The names of the files, along with other parameters, are defined in a table that can consist of multiple entries of key and data files. The table can be set up using simple Assembler. An example of code to define a table follows:

```
         CSECT
         DC    A($JCTDEFS)  Do not change.
*                           Definition of the first Key file
         DC    CL8'CLNFSTX' Name of the Key file
         DC    AL1(2)       Pointer type, 1- Singly 2- Doubly
         DC    AL1(2)       No of data files
*                           Definition of Data files for this Key file
*                           Definition of the first Data file
         DC    CL8'CLNFSTL' Name of the Data file
         DC    AL1(3)       Number of keys to access a data record
                            Definition of the second Data file
         DC    CL8'CLNFSTR'
*        DC    AL(2)
*                           Definition of the second Key file
         DC    CL8'CLNDPT '
         DC    AL1(1)
         DC    AL1(1)
*                           Definition of Data files for this Key file
         DC    CL8'CLNJRN '
         DC    AL1(3)
*                           Table control data
$JCTDEFS EQU   *
         DC    AL1(2)       Number of Key files
         DC    AL1(1Ø)      Length of Key file table entry
         DC    AL1(9)       Length of Data file table entry
         END
```

Note: the ‘number of data files’ counter must be incremented when you add a new datafile to a key file and the ‘number of key files’ counter must be incremented when you add a new key file.

KEY FILE

```
Record length  =  Length of the  +  (Pointer-type  X  Number of   X 4)
                  physical key                        data files
```

The length of the physical key must be either equal to, or greater than, the maximum length of the key the user expects to handle. The key must start at position 0.

For future requirements, I recommend that you make the length of the physical key a little longer than the maximum length of the key you expect to handle. For consistency, pad the key value with either zeros or spaces equal to the length of the physical key.

DATA FILE

```
Record length = Length of the  +  (Number of keys through which   X 4)
                user data area     data records will be accessed
```

The first four bytes of the first record of a data file must contain the value X'00000002'.

DATA MANIPULATION

The application program calls an interface with a set of parameters to perform a data manipulation operation. The first parameter (function code) tells the interface the type of data manipulation operation being requested. The program must check the EIBRCODE and EIBFN of EIB of the program after control is returned to the application to see the status of the operation performed. The following special values are returned in the EIBFN code:

- X'0001' – invalid function code.
- X'0002' – the key file specified is not defined in the file definition table.
- X'0003' – the data file specified is not defined in the file definition table.

OPEN (FUNC-CODE – X'0001')

Before any data manipulation operation, the program must issue an OPEN command and establish communication with the system:

```
CALL 'VBLJIOR' USING FUNC-CODE, SYS-PARMS, FILE-DEFS
Ø1 FUNC-CODE     PIC S9(4)   COMP  VALUE +1.
Ø1 SYS-PARMS.
   Ø2 SYSBLK-ADDR PIC  S9(8)  COMP.
Ø1 FILE-DEFS.
   Ø2 FDT-NAME  PIC X(8) VALUE 'CLNFDT'.  Name of file definition table
   Ø2 IXF-NAME  PIC X(8) VALUE 'CLNFSTX'. Name of the Key file
   Ø2 RF-NAME   PIC X(8) VALUE 'CLNFSTR'. Name of the Data file
```

If the OPEN function is successful then the system will pass a value in SYSBLK-ADDR (System Control Block Address), which must be passed to the interface when doing a data manipulation operation.

CLOSE (FUNC-CODE – X'0002')

After finishing data manipulation operations, the user must perform a CLOSE operation to free the main storage acquired by the interface:

```
     CALL 'VBLJIOR' USING FUNC-CODE, SYSBLK-ADDR
```

WRITE (FUNC-CODE – X'0003')

To write data records:

```
CALL 'VBLJIOR' USING FUNC-CODE, SYS-PARMS, KEY-DEFS,IO-AREA
Ø1  SYS-PARMS.
    Ø2  SYSBLK-ADDR         PIC S9(8) COMP.
    Ø2  REC-RRN             PIC S9(8) COMP.
    Ø2  NEXT-REC-RRN        PIC S9(8) COMP.
Ø1  KEY-DEFS.
    Ø2  KEY-INDEX           PIC  S9(4)  COMP.
    Ø2  KEY-VALUE           PIC (...).
Ø1  IO-AREA                 PIC(...).
```

Where:

- ‘KEY-INDEX’ – a data record can be accessed by a number of keys (this number is defined in the table). The system identifies a key with which records will be accessed, for example, in a banking environment, transaction records will be needed by branch-id, teller-id, etc. In an application to access records in branch-id order, the application can assign 1 for KEY-INDEX when storing data and 2 to access records in teller-id order.

- ‘KEY-VALUE’ – the value of the key the data record belongs to (eg branch-id, teller-id, etc).
- ‘REC-RRN’ – the system will pass the RRN of the data record if the WRITE operation is successful.
- ‘NEXT-REC-RRN’ – the RRN of the next record. This can contain a zero value.
- ‘IO-AREA’ – the user data area to be stored.

After the WRITE operation, the user can link the added record to a new key value. For example, after adding a record in the branch-id sequence, the user can link the same record to access the teller-id sequence. For this, the user has to carry out the WRITE operation again. The KEY-INDEX value that was used for accessing records in the branch-id sequence cannot be used. The KEY-VALUE must contain the new key value (for example, teller-id). REC-RRN must contain the value passed by the interface after the WRITE operation was performed. The IO-AREA is not needed to pass in this function.

READ (FUNC-CODE – X'0004')

To READ records, the KEY-INDEX and KEY-VALUE must have related values pertinent to the record to be read. The REC-RRN will contain the RRN of the READ record and the NEXT-REC-RRN will contain the RRN of the next record in the chain. The user can READ the next record by moving the value in NEXT-REC-RRN to REC-RRN and calling the interface again. In this way the entire chain of records can be read (ie until the value of NEXT-REC-RRN becomes zero):

```
CALL 'VBLJIOR' USING FUNC-CODE, SYS-PARMS, KEY-DEFS IO-AREA
```

If the operation is successful, the IO-AREA will contain the READ data record.

READ WITH UPDATE (FUNC-CODE – X'0005')

This is similar to the READ function. The user can update the READ record using the UPDATE function.

UPDATE (FUNC-CODE – X'0006')

This is to update the record READ with the UPDATE function. The IO-AREA must contain the updated user data:

```
     CALL 'VBLJIOR' USING FUNC-CODE, SYSBLK-ADDR, IO-AREA
```

UNLOCK (FUNC-CODE – X'0007')

This is to unlock the dataset READ with the READ WITH UPDATE function:

```
     CALL 'VBLJIOR' USING FUNC-CODE, SYSBLK-ADDR
```

SYSTEM REQUIREMENTS

Create a PHASE type member from the source code of a file definition table. Create an OBJ type member from the source code of VBJIOR routine and link-edit your program to include it. An entry must be made in the PPT for the file's definition table. The FCT must have entries for all the files defined in the file definition table.

This system has been successfully tested under CICS 2.1 and VSE/ESA 2.1.

SOURCE CODE

```
DFHEISTG DSECT
RPSADDR  DS    F
XPSADDR  DS    F
SAVERRN  DS    F
RKEY     DS    F
ORGADDR  DS    F
DESTADDR DS    F
UIOADDR  DS    F
FLEN     DS    F
SAVLNKRG DS    F
SAVEREG  DS    16F
JTRLEN   DS    ØF
JTRLENX  DS    H
JTRLENR  DS    H
IXPLEN   DS    H
ORGTAG   DS    H
DESTAG   DS    H
FCODE    DS    H
ERRCODE  DS    CL1
SYSFLOWP DSECT
```

```
SCBADR   DS    F
RELKEY   DS    F
NRELKEY  DS    F
KEYD     DSECT
IXNO     DS    H
XKEY     DS    CL1ØØ
         ORG   IXNO
JDTNAME  DS    CL8
XFLNAMEI DS    CL8
RFLNAMEI DS    CL8
IOAREAD  DSECT
IOAREA   DS    CL1ØØØ
SCBDSECT DSECT                          SYSTEM CONTROL BLOCK
SYSBBEG  EQU   *
EIBADDR  DS    F
IOXADDR  DS    F
IORADDR  DS    F
XFLNAME  DS    CL8
XLEN     DS    H
XKLEN    DS    H
PTRTYPE  DS    H
RFLNAME  DS    CL8
PTRNO    DS    H
RLEN     DS    H
NOIXS    DS    H
SYSBEND  EQU   *
VBLJIOR  CSECT
         USING SYSFLOWP,R7
         USING KEYD,R8
         USING IOAREAD,R9
         USING SCBDSECT,R4
         LM    R6,R9,Ø(R1)
         ST    R9,UIOADDR
         MVC   FCODE,Ø(R6)
         EXEC  CICS HANDLE CONDITION                                   X
               NOTFND(ADDIXØ)                                          X
               DUPREC(DUPREC)                                          X
               ERROR(EOJ)
         CLI   FCODE+1,X'ØB'
         BH    INVALFUN
         LH    R5,FCODE
         LTR   R5,R5
         BZ    INVALFUN
         CLI   FCODE+1,X'Ø1'
         BE    OPEN
         L     R4,SCBADR
         MVC   DFHEIBP,EIBADDR         ESTABLISH ADDRESSABILITY
         L     DFHEIBR,DFHEIBP         TO CALLER'S EIB
         SLL   R5,2
         LA    R1Ø,FUNCRADR
         L     R1Ø,Ø(R5,R1Ø)
         BR    R1Ø
FUNCRADR DS    ØF                      FUNCTIONAL ROUTINE ADDRESSES
         DC    A(Ø)
         DC    A(OPEN)
         DC    A(CLOSE)
         DC    A(ADDIX)
         DC    A(READ)
         DC    A(READ)
         DC    A(UPDATE)
         DC    A(UNLOCK)
OPEN     DS    ØH                      OPEN FUNCTIONAL ROUTINE
         BAL   R1Ø,GEIBADR
         EXEC  CICS LOAD PROGRAM(JDTNAME) SET(R5) FLENGTH(FLEN)
         EXEC  CICS GETMAIN LENGTH(JCBLEN) SET(R6) INITIMG(HEXØ)
         ST    R6,SCBADR
         LR    R4,R6
         MVC   EIBADDR,DFHEIBP
         L     R6,Ø(R5)
         XC    JTRLEN,JTRLEN
         XR    R9,R9
         IC    R9,Ø(R6)
         MVC   JTRLENX+1(1),1(R6)
         MVC   JTRLENR+1(1),2(R6)
         LA    R5,4(R5)
NEXTXFLR DS    ØH                      CHECK KEY FILE IS DEFINED
         CLC   Ø(8,R5),XFLNAMEI        IN THE JOURNAL TABLE
         BE    XFLRFND
         XR    R6,R6
         IC    R6,9(R5)
         MH    R6,JTRLENR
         AH    R5,JTRLENX
         AR    R5,R6
         BCT   R9,NEXTXFLR
         MVI   ERRCODE,X'Ø2'           IF SPECIFIED KEY FILE NOT FOUND
*                                      IN THE TABLE
         B     SETERR
XFLRFND  DS    ØH                      CHECK DATA FILE IS DEFINED IN
         MVC   XFLNAME,Ø(R5)           THE JOURNAL TABLE
         MVC   PTRTYPE+1(1),8(R5)
         EXEC  CICS INQUIRE FILE(XFLNAME) RECORDSIZE(SAVEREG)          X
               KEYLENGTH(RKEY)
         MVC   XLEN,SAVEREG+2
         MVC   XKLEN,RKEY+2
         IC    R9,9(R5)
         AH    R5,JTRLENX
         XR    R6,R6
NEXTRFLR DS    ØH
         CLC   Ø(8,R5),RFLNAMEI
         BE    RFLRFND
         AH    R5,JTRLENR
         LA    R6,1(R6)
         BCT   R9,NEXTRFLR
         MVI   ERRCODE,X'Ø3'            IF DATA FILE NOT FOUND IN THE
*                                       TABLE
         B     SETERR
RFLRFND  DS    ØH
         STH   R6,PTRNO
         MVC   RFLNAME,Ø(R5)
         MVC   NOIXS+1(1),8(R5)
         EXEC  CICS INQUIRE FILE(RFLNAME) RECORDSIZE(SAVEREG)
         MVC   RLEN,SAVEREG+2
         B     EOJ
*
CLOSE    DS    ØH                       CLOSE FUNCTIONAL ROUTINE
         EXEC  CICS FREEMAIN DATA(SCBDSECT)
         B     EOJ
ADDIX    DS    ØH                       WRITE FUNCTIONAL ROUTINE
         CLC   RELKEY,FZERO
         BH    UPDATEIX
         LA    R1Ø,ADDIX1
         B     ADDREC
ADDIXØ   DS    ØH                       PROGRAM HALTS PROCESSING
*                                       IF NOT FOUND CONDITION OCCURS
*                                       WHEN PERFORMING FUNCTION OTHER
*                                       THAN WRITE.
         CLI   FCODE+1,X'Ø3'
         BNE   EOJ
ADDIX1   DS    ØH
         EXEC  CICS GETMAIN LENGTH(XLEN) SET(R6) INITIMG(HEXØ)
         ST    R6,IOXADDR
         LH    R5,XKLEN
         BCTR  R5,Ø
         EX    R5,MVCKVIOA
         BAL   R1Ø,GXPSADDR
         L     R6,XPSADDR
         MVC   Ø(4,R6),RELKEY
         CLI   PTRTYPE+1,X'Ø1'
         BE    ADDIX2
         MVC   4(4,R6),RELKEY
ADDIX2   DS    ØH
         L     R9,IOXADDR
         BAL   R1Ø,WRITEX
         EXEC  CICS FREEMAIN DATA(IOAREA)
         B     EOJ
MVCKVIOA MVC   Ø(Ø,R6),XKEY
DUPREC   DS    ØH                       UPDATE POINTER LINKS IF DUPKEY
*                                       CONDITION OCCURS WHEN PERFORMING
*                                       WRITE OPERATION
*
         EXEC  CICS FREEMAIN DATA(IOAREA)
UPDATEIX DS    ØH                       UPDATE POINTER LINKS ROUTINE
         BAL   R1Ø,READXU
         BAL   R1Ø,GXPSADDR
         L     R6,XPSADDR
         CLI   PTRTYPE+1,X'Ø1'
         BE    UPDATIX1
         MVC   RKEY,4(R6)
         MVC   4(4,R6),RELKEY
         CLC   RKEY,FZERO
         BH    UPDLAST
         MVC   Ø(4,R6),RELKEY
         B     UPDATIX3
UPDLAST  DS    ØH
         BAL   R1Ø,READRU
         BAL   R1Ø,GRPSADDR
         L     R6,RPSADDR
         MVC   Ø(4,R6),RELKEY
         B     UPDATIX2
UPDATIX1 DS    ØH
         MVC   SAVERRN,Ø(R6)
         MVC   Ø(4,R6),RELKEY
         CLC   SAVERRN,FZERO
         BNH   UPDATIX3
         MVC   RKEY,RELKEY
         BAL   R1Ø,READRU
         BAL   R1Ø,GRPSADDR
         L     R6,RPSADDR
         MVC   Ø(4,R6),SAVERRN
UPDATIX2 DS    ØH
         L     R9,IORADDR
         BAL   R1Ø,UPDATER
UPDATIX3 DS    ØH
         L     R9,IOXADDR
         LA    R1Ø,EOJ
         B     UPDATEX
ADDREC   DS    ØH                      WRITE DATA RECORD TO DATA FILE
         ST    R1Ø,SAVLNKRG
         MVC   RKEY,FONE
         BAL   R1Ø,READRU
         L     R9,IORADDR
         L     R5,Ø(R9)
         ST    R5,RKEY
         LA    R5,1(R5)
         ST    R5,Ø(R9)
         BAL   R1Ø,UPDATER
         EXEC  CICS GETMAIN LENGTH(RLEN) SET(R6) INITIMG(HEXØ)
         ST    R6,IORADDR
         MVC   DESTADDR,IORADDR
         MVI   DESTAG+1,X'Ø1'
         MVC   ORGADDR,UIOADDR
         XC    ORGTAG,ORGTAG
         BAL   R1Ø,MTAFVIOA
         L     R9,IORADDR
         BAL   R1Ø,WRITER
         MVC   RELKEY,RKEY
         XC    NRELKEY,NRELKEY
         EXEC  CICS FREEMAIN DATA(IOAREA)
         L     R1Ø,SAVLNKRG
         BR    R1Ø
READ     DS    ØH                         READ AND READ WITH UPDATE
*                                         FUNCTIONAL ROUTINE
         CLC   RELKEY,FZERO
         BH    READ1
         BAL   R1Ø,READX
         BAL   R1Ø,GXPSADDR
         L     R6,XPSADDR
         MVC   RELKEY,Ø(R6)
READ1    DS    ØH
         MVC   RKEY,RELKEY
         CLI   FCODE+1,X'Ø5'
         BE    READ2
         LA    R1Ø,READ3
         B     READR
READ2    DS    ØH
         BAL   R1Ø,READRU
READ3    DS    ØH
         BAL   R1Ø,GRPSADDR
         L     R6,RPSADDR
         MVC   NRELKEY,Ø(R6)
         MVC   ORGADDR,IORADDR
         MVI   ORGTAG+1,X'Ø1'
         MVC   DESTADDR,UIOADDR
         XC    DESTAG,DESTAG
         LA    R1Ø,EOJ
         B     MTAFVIOA
UPDATE   DS    ØH                         UPDATE FUNCTIONAL ROUTINE
         ST    R8,UIOADDR
         MVC   DESTADDR,IORADDR
         MVI   DESTAG+1,X'Ø1'
         MVC   ORGADDR,UIOADDR
         XC    ORGTAG,ORGTAG
         BAL   R1Ø,MTAFVIOA
         L     R9,IORADDR
         LA    R1Ø,EOJ
         B     UPDATER
UNLOCK   DS    ØH                         UNLOCK FUNCTIONAL ROUTINE
         EXEC CICS UNLOCK DATASET(RFLNAME)
         B     EOJ
GXPSADDR DS    ØH                         GET POINTER ADDRESS CORR TO
*                                         THE DATA FILE
         STM   RØ,R15,SAVEREG
         L     R6,IOXADDR
         AH    R6,XKLEN
         LH    R5,PTRTYPE
         SLL   R5,2
         MH    R5,PTRNO
         AR    R6,R5
         ST    R6,XPSADDR
         LM    RØ,R15,SAVEREG
         BR    R1Ø
GRPSADDR DS    ØH                      GET POINTER ADDRESSES CORR
*                                      TO THE KEY VALUE
         STM   RØ,R15,SAVEREG
         L     R6,IORADDR
         LH    R5,IXNO
         BCTR  R5,Ø
         SLL   R5,2
         AR    R6,R5
         ST    R6,RPSADDR
         LM    RØ,R15,SAVEREG
         BR    R1Ø
MTAFVIOA DS    ØH                      MOVE DATA RECORD TO AND FROM
*                                      CALLER AREA
         STM   RØ,R15,SAVEREG
         L     R8,ORGADDR
         LTR   R8,R8
         BZ    EMTAFVIO
         L     R1Ø,DESTADDR
         LTR   R1Ø,R1Ø
         BZ    EMTAFVIO
         LH    R9,NOIXS
         SLL   R9,2
         STH   R9,IXPLEN
         LR    R11,R9
         MH    R9,ORGTAG
         MH    R11,DESTAG
         AR    R8,R9
         AR    R1Ø,R11
         LH    R9,RLEN
         SH    R9,IXPLEN
         LR    R11,R9
         MVCL  R1Ø,R8
EMTAFVIO DS    ØH
         LM    RØ,R15,SAVEREG
         BR    R1Ø
*                                      FOLLOWING ARE CICS DATA
*                                      MANIPULATION COMMANDS
READX    DS    ØH
         EXEC  CICS READ DATASET(XFLNAME) SET(R6)                      X
               LENGTH(XLEN)  RIDFLD(XKEY) KEYLENGTH(XKLEN)
         ST    R6,IOXADDR
         BR    R1Ø
READXU   DS    ØH
         EXEC  CICS READ DATASET(XFLNAME) SET(R6)                      X
               LENGTH(XLEN)  RIDFLD(XKEY) KEYLENGTH(XKLEN) UPDATE
         ST    R6,IOXADDR
         BR    R1Ø
WRITEX   DS    ØH
```

```
         EXEC  CICS WRITE DATASET(XFLNAME) FROM(IOAREA)                X
               LENGTH(XLEN)  RIDFLD(XKEY) KEYLENGTH(XKLEN)
         BR    R1Ø
UPDATEX  DS    ØH
         EXEC  CICS REWRITE DATASET(XFLNAME) FROM(IOAREA)              X
               LENGTH(XLEN)
         BR    R1Ø
READR    DS    ØH
         EXEC  CICS READ DATASET(RFLNAME) SET(R6)                      X
               LENGTH(RLEN)  RIDFLD(RKEY) RRN
         ST    R6,IORADDR
         BR    R1Ø
READRU   DS    ØH
         EXEC  CICS READ DATASET(RFLNAME) SET(R6)                      X
               LENGTH(RLEN)  RIDFLD(RKEY) RRN UPDATE
         ST    R6,IORADDR
         BR    R1Ø
WRITER   DS    ØH
         EXEC  CICS WRITE DATASET(RFLNAME) FROM(IOAREA)                X
               LENGTH(RLEN)  RIDFLD(RKEY) RRN
         BR    R1Ø
UPDATER  DS    ØH
         EXEC  CICS REWRITE DATASET(RFLNAME) FROM(IOAREA)              X
               LENGTH(RLEN)
         BR    R1Ø
*                                      EIBRCODE OF THE CALLERS EIB
*                                      IS VALIDATED BEFORE CONTROL IS
*                                      PASSED TO CALLING PROGRAM
INVALFUN DS    ØH                      IF INVALID FUNCTION CODE IS
*                                      PASSED TO INTERFACE
         MVI   ERRCODE,X'Ø1'
SETERR   DS    ØH
         BAL   R1Ø,GEIBADR
         XC    EIBFN,EIBFN
         MVC   EIBRCODE(1),ERRCODE
         B     EOJ
GEIBADR  DS    ØH                      GET EIB ADDRESS OF CALLER
         EXEC  CICS ADDRESS EIB(R5)    AND SAVE IN SYSTEM CONTROL
         ST    R5,DFHEIBP              BLOCK
         L     DFHEIBR,DFHEIBP
         BR    R1Ø
FZERO    DC    F'Ø'
FONE     DC    F'1'
JCBLEN   DC    AL2(SYSBEND-SYSBBEG)
HEXØ     DC    XL1'ØØ'
         COPY  REGSTRS
EOJ      DS    ØH
         END   VBLJIOR
```

Janaka K Dabare
Software Solutions (Sri Lanka)

CICS system generator – part 3

This month we conclude the article to develop a tool that generates CICS regions on different LPARs.

CICSDEFJ

```
//&USERID.J JOB (,EXP),'&USERID',
// NOTIFY=&USERID,
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
// TIME=3
//*
/*ROUTE  XEQ &LPAR
//*
//* THIS JOB DELETES AND RECREATES
//* CICS SYSTEM AND USER JOURNAL DATASETS
//*
//*
//DELETE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE &SYS..DFHJACD
 DELETE &SYS..DFHJØ1A NONVSAM
 DELETE &SYS..DFHJØ1B NONVSAM
 DELETE &SYS..DFHJØ1X NONVSAM
 DELETE &SYS..DFHJØ2A NONVSAM
 SET MAXCC=Ø
/*
//DEFINE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 /*                                  */
 /* DEFINE A JOURNAL ARCHIVE DATASET */
 /*                                  */
 DEFINE CLUSTER(NAME(&SYS..DFHJACD)-
          NUMBERED -
          REC(199)-
          RECORDSIZE(5Ø5 5Ø5)-
          CISZ(512)-
          SHR(4)-
          VOLUMES(&VOL)) -
        DATA(NAME(&SYS..DFHJACD.DATA))
//*
//FRMTJØ1A EXEC PGM=DFHJCJFP
//STEPLIB  DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//JOURNAL  DD DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(4),,CONTIG),
// UNIT=SYSDA,VOL=SER=&VOL,
// DSN=&SYS..DFHJØ1A
//*
//FRMTJØ1B EXEC PGM=DFHJCJFP
//STEPLIB  DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//JOURNAL  DD DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(4),,CONTIG),
// UNIT=SYSDA,VOL=SER=&VOL,
// DSN=&SYS..DFHJØ1B
//*
//FRMTJØ1X EXEC PGM=DFHJCJFP
//STEPLIB  DD DSN=CICS&VRM.SDFHLOAD,DISP=SHR
//JOURNAL  DD DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(4),,CONTIG),
// UNIT=SYSDA,VOL=SER=&VOL,
// DSN=&SYS..DFHJØ1X
//*
//FRMTJØ2A EXEC PGM=DFHJCJFP
//STEPLIB  DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//JOURNAL  DD DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(4),,CONTIG),
// UNIT=SYSDA,VOL=SER=&VOL,
// DSN=&SYS..DFHJØ2A
//*
```

CICSDEFP

```
//&USERID.P JOB    (ACCT#),'INSTALL',CLASS=A,MSGCLASS=X,
//          NOTIFY=&USERID
//*********************************************************************/
//* JOB NAME = CICSPROC                                                */
//*                                                                    */
//* DESCRIPTIVE NAME = INSTALLATION JOB STREAM                         */
//*                                                                    */
//* FUNCTION = MVS MODIFICATIONS                                       */
//*                                                                    */
//* PSEUDOCODE =                                                       */
//*   CICSIPM  STEP      FOR UPDATING THE MVS PROCLIB WITH CICS:       */
//*                      1) STARTUP PROCEDURES                         */
//*                                                                    */
//* NOTES =                                                            */
//*   PLEASE CHECK THIS JOB CAREFULLY TO ENSURE THAT THE SYSTEM        */
//*   LIBRARY NAMES ARE THE CORRECT LIBRARY NAMES FOR YOUR SITE.       */
//*************************************************
//*    ADD CATALOGUED PROCEDURES TO PROCLIB       *
//*************************************************
/*ROUTE  XEQ &LPAR
//*************************************************
//CICSIPM EXEC PGM=IEBUPDTE,PARM=NEW,COND=(4,LT)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT2   DD DISP=SHR,DSN=SYS1.PROCLIB.CICS
//SYSIN    DD DATA
./  ADD NAME=&SYS
```

```
//&SYS     PROC REG=ØM,START=AUTO,NEWSIT=NO
//*
//*---------------------------------
//&SYS    EXEC PGM=DFHSIP,REGION=&EMP.REG,TIME=144Ø,
//   PARM='SIT=&SYSID,START=&EMP.START,NEWSIT=&EMP.NEWSIT,SYSIN'
//*---------------------------------
//*            CICS STEPLIB - CONTAINS MODULES LOADED BY
//*            SUPERVISOR SERVICES
//*---------------------------------
//STEPLIB   DD DSN=CICS&VRM..SDFHAUTH,DISP=SHR
//          DD DSN=CICS&VRM..APFLOAD,DISP=SHR
//          DD DSN=SYS1.COB2CICS,DISP=SHR
//*         DD DSN=SYS1.COB2LIB,DISP=SHR
//*---------------------------------
//*            DFHRPL CONTAINS CICS MANAGEMENT PROGRAMS
//*            AND TABLES THAT ARE LOADED BY CICS/OS/VS
//*            AS WELL AS APPLICATION PROGRAMS
//*---------------------------------
//DFHRPL    DD DSN=&SYS..TABLOAD,DISP=SHR
//          DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//*         DD DSN=DSN.SDSNLOAD,DISP=SHR
//          DD DSN=SYS1.COB2CICS,DISP=SHR
//          DD DSN=SYS1.COB2LIB,DISP=SHR
//          DD DSN=CICS&VRM..SDFHLPA,DISP=SHR
//*---------------------------------
//*            TEMPORARY DATASET
//*---------------------------------
//DFHTEMP   DD DSN=&SYS..DFHTEMP,DISP=SHR
//*---------------------------------
//*            INTRA-PARTITION DATA SET (VSAM)
//*---------------------------------
//DFHINTRA  DD DSN=&SYS..DFHINTRA,DISP=SHR
//*---------------------------------
//*            RESTART  DATA SET (VSAM)
//*---------------------------------
//DFHRSD    DD DSN=&SYS..DFHRSD,DISP=SHR
//*---------------------------------
//*            CICS LOCAL CATALOG DATASET
//*---------------------------------
//DFHLCD    DD DSN=&SYS..DFHLCD,DISP=SHR
//*---------------------------------
//*            CICS GLOBAL CATALOG DATASET
//*---------------------------------
//DFHGCD    DD DSN=&SYS..DFHGCD,DISP=SHR
//*---------------------------------
//*            AUXILIARY TRACE DATA SET
//*---------------------------------
//DFHAUXT   DD DSN=&SYS..DFHAUXT,DISP=SHR
//DFHBUXT   DD DSN=&SYS..DFHBUXT,DISP=SHR
//*---------------------------------
```

```
//*            SYSTEM LOG + JOURNALFILES
//*---------------------------------
//DFHJØ1A   DD DSN=&SYS..DFHJØ1A,DISP=SHR
//DFHJØ1B   DD DSN=&SYS..DFHJØ1B,DISP=SHR
//*---------------------------------
//*            AUTOMATIC JOURNAL ARCHIVING DATASET
//*---------------------------------
//DFHJACD   DD DSN=&SYS..DFHJACD,DISP=SHR
//DFHJPDS   DD DSN=&SYS..DFHJPDS,DISP=SHR
//DFHJOUT   DD SYSOUT=(A,INTRDR)
//*---------------------------------
//*            EXTRAPARTITION DATASETS
//*---------------------------------
//DFHCXRF   DD SYSOUT=X
//LOGUSR    DD SYSOUT=X,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//MSGUSR    DD SYSOUT=X,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//*---------------------------------
//*            CICS DUMP DATASET(S) AND SYSUDUMP
//*---------------------------------
//DFHDMPA   DD DSN=&SYS..DFHDMPA,DISP=SHR
//DFHDMPB   DD DSN=&SYS..DFHDMPB,DISP=SHR
//SYSABEND  DD SYSOUT=X
//TRACEOUT  DD SYSOUT=X
//*---------------------------------
//*            SIT-OVERWRITE PARAMETERS
//*---------------------------------
//SYSIN     DD DSN=&SYS..TABSRC(SITOVER),DISP=SHR
//*---------------------------------
//*            END OF CICS SYSTEM DATASETS
//*---------------------------------
//*---------------------------------
//*            INTERNAL  READERS
//*---------------------------------
./  ENDUP
```

CICSDEFR

```
//&USERID.R JOB (,EXP),'&USERID',
// NOTIFY=&USERID,
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
// TIME=3
//*
/*ROUTE  XEQ &LPAR
//*
//* THIS JOB DELETES AND RECREATES THE
//* CICS RESTART DATASET
//*
//*
//DELETE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
```

```
//SYSIN    DD *
 DELETE &SYS..DFHRSD
 DELETE &SYS..DFHXRCTL
 DELETE &SYS..DFHXRMSG
 SET MAXCC=Ø
/*
//DEFINE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 /*                          */
 /* DEFINE A RESTART DATASET */
 /*                          */
 DEFINE CLUSTER(NAME(&SYS..DFHRSD) -
          KEYS(22 Ø)-
          INDEXED -
          RECORDSIZE(4ØØ 2ØØØ)-
          REC(1ØØ 1ØØ)-
          CISZ(2Ø48)-
          SHR(2)-
          VOLUMES(&VOL)) -
        DATA(NAME(&SYS..DFHRSD.DATA)-
          UNIQUE)-
       INDEX(NAME(&SYS..DFHRSD.INDEX)-
          UNIQUE)
 /*                                  */
 /* DEFINE THE XRF DATASETS          */
 /*                                  */
 DEFINE CLUSTER(NAME(&SYS..DFHXRCTL)-
          TRACKS(1)-
          NIXD -
          SPEED -
          VOLUMES(&VOL)) -
        DATA(NAME(&SYS..DFHXRCTL.DATA)-
          CISZ(4Ø96)-
          SHR(3,3))
 DEFINE CLUSTER(NAME(&SYS..DFHXRMSG)-
          CYLINDERS(2)-
          SPEED -
          NIXD -
          VOLUMES(&VOL)) -
        DATA(NAME(&SYS..DFHXRMSG.DATA)-
          CISZ(4Ø96)-
          SHR(3,3))
/*
//INITRSD   EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT  DD SYSOUT=*
//RSD DD DSN=&SYS..DFHRSD,DISP=SHR
//SYSIN     DD *
 REPRO INFILE(RSDREC) OUTFILE(RSD)
/*
```

```
//*
//RSDREC    DD DSN=CICS&VRM..SDFHINST(DFHINSTØ),DISP=SHR
```

CICSDEFS

```
//&USERID.S JOB (,EXP),'&USERID',
// NOTIFY=&USERID,
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
// TIME=3
//*
/*ROUTE  XEQ &LPAR
//*
//* THIS JOB DELETES AND RECREATES
//* CICS SAMPLE DATA SET
//*
//*
//DELETE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE &SYS..FILEA
 SET MAXCC=Ø
/*
//DEFINE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 /*                               */
 /* DEFINE A FILEA DATASET        */
 /*                               */
 DEFINE CLUSTER(NAME(&SYS..FILEA)-
          REC(8Ø)-
          IXD -
          VOLUMES(&VOL)) -
         DATA(NAME(&SYS..FILEA.DATA)-
         RECSZ(8Ø 8Ø)-
         UNIQUE -
         KEYS(6 1)) -
        INDEX(NAME(&SYS..FILEA.INDEX)-
          UNIQUE -
          SHR(2 3))
//*
//* LOAD VSAM DATA FILE
//*
//LOADFILE EXEC PGM=DFH$LDSP
//STEPLIB  DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//FILEA    DD DISP=SHR,
// DSN=&SYS..FILEA
//SYSOUT   DD SYSOUT=*
//SYSIN    DD DISP=SHR,
// DSN=CICS&VRM..SDFHSAMP(DFH$FAIN)
```

CICSDEFW

```
//&USERID.W JOB (,EXP),'&USERID',
// NOTIFY=&USERID,
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
// TIME=3
//*
/*ROUTE  XEQ &LPAR
//*
//* THIS JOB DELETES AND RECREATES
//* CICS TEMP AND INTRA DATASETS
//*
//*
//DELETE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE &SYS..DFHTEMP
 DELETE &SYS..DFHINTRA
 SET MAXCC=0
/*
//DEFINE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 /*                                  */
 /* DEFINE TEMPORARY STORAGE DATASET */
 /*                                  */
 DEFINE CLUSTER(NAME(&SYS..DFHTEMP)-
          RECORDSIZE(4089,4089)-
          REC(200)-
          NIXD -
          CISZ(4096)-
          SHR(2 3)-
          VOLUMES(&VOL)) -
        DATA(NAME(&SYS..DFHTEMP.DATA)-
          UNIQUE)
 /*                           */
 /* DEFINE AN INTRA-PARTITION */
 /* TRANSIENT DATASET         */
 /*                           */
 DEFINE CLUSTER(NAME(&SYS..DFHINTRA)-
          RECORDSIZE(4089,4089)-
          REC(100)-
          NIXD -
          CISZ(4096)-
          SHR(2 3)-
          VOLUMES(&VOL)) -
        DATA(NAME(&SYS..DFHINTRA.DATA)-
          UNIQUE)
```

CICSIVPB

```
//&USERID.P JOB    (ACCT#),'INSTALL',CLASS=A,MSGCLASS=X,
//          NOTIFY=&USERID
//*********************************************************************/
/*ROUTE  XEQ &LPAR
//*********************************************************************/
//*
//* EXECUTE DUMP UTILITY PROGRAM TO PRINT THE
//* CONTENTS OF THE DUMP DATASET A
//PRTDMPA  EXEC  PGM=DFHDU410,PARM=SINGLE,
// REGION=0M
//STEPLIB  DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//DFHTINDX DD SYSOUT=X
//SYSPRINT DD SYSOUT=X
//DFHPRINT DD SYSOUT=X
//DFHDMPDS DD DISP=SHR,
// DSN=&SYS..DFHDMPA
//SYSIN    DD DUMMY
//*        EXECUTE DUMP UTILITY PROGRAM TO PRINT THE
//*        CONTENTS OF THE DUMP DATASET B
//PRTDMPB  EXEC  PGM=DFHDU410,PARM=SINGLE,
// REGION=0M
//STEPLIB  DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//DFHTINDX DD SYSOUT=X
//SYSPRINT DD SYSOUT=X
//DFHPRINT DD SYSOUT=X
//DFHDMPDS DD DISP=SHR,
// DSN=&SYS..DFHDMPB
//SYSIN    DD DUMMY
//*        EXECUTE TRACE UTILITY PROGRAM TO PRINT THE
//*        CONTENTS OF THE AUXILIARY TRACE DATASET 'A'.
//*        THIS DATASET WILL BE EMPTY UNLESS SIT
//*        PARAMETER AUXTR IS SET TO ON.
//PRTAUXT  EXEC PGM=DFHTU410,REGION=0M
//STEPLIB  DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//DFHAUXT  DD DISP=SHR,
// DSN=&SYS..DFHAUXT
//DFHAXPRT DD SYSOUT=X
//DFHAXPRM DD DUMMY
//*        EXECUTE TRACE UTILITY PROGRAM TO PRINT THE
//*        CONTENTS OF THE AUXILIARY TRACE DATASET 'B'.
//*        THIS DATASET WILL BE EMPTY UNLESS SIT
//*        PARAMETER AUXTR IS SET TO ON.
//PRTBUXT  EXEC PGM=DFHTU410,REGION=0M
//STEPLIB  DD DSN=CICS&VRM..SDFHLOAD,DISP=SHR
//DFHAUXT  DD DISP=SHR,
// DSN=&SYS..DFHBUXT
//DFHAXPRT DD SYSOUT=X
//DFHAXPRM DD DUMMY
//*
```

LOCAL CUSTOMIZATION

Because of some local naming conventions, it is possible that you will have to make some changes in the system generator tool.

You should consider the following:

- Ensure that you have defined all your local library names.
- Some entries in the CSD update jobs reflect our local definitions (typeterms, filenames, etc). You should customize these to your local standards.
- After filling the macro source library, you have to assemble these macro sources.

Paul Jansen
Systems Programmer
Interpay (The Netherlands)

CICS/MVS 2.1.2 to CICS/ESA 4.1 migration

There are a number of shops running critical applications on CICS/MVS Version 2.1.2 or lower. With the Year 2000 just around the corner, these must be upgraded to at least the stable release of CICS/ESA 4.1. The following are some useful hints and tips to use during this upgrade.

RECOMMENDED READING

It would be a good idea to read the manuals shown in Figure 1, which are between Versions 2.1.2 and 4.1, otherwise you will miss features and changes that have been implemented with those middle releases. You should also read the *CICS-supplied transactions guide* to become familiar with (at least) the transactions shown in Figure 2.

If you are using a native CICS sign-on and sign-off, make sure your CICS users are familiar with the change from CSSN/CSSF to CESN/CESF.

One of your very first tasks should be to identify the availability of compliant vendor software. I started with a vanilla CICS region and

Title	IBM order #	IBM CD ROM
		(SK2T-0730-20)
R321 Migration Guide	(GC33-0656)	N/A
R330 Release Guide	(GC33-0792)	DFHPE413
R330 Program Directory	(GC33-0922)	DFHPA904
R330 Installation Guide	(SC33-0663)	DFHPA110
R330 System Definition Guide	(SC33-0664)	DFHPA211
R330 Operations Guide	(SC33-0668)	DFHPA608
R410 Migration Guide	(GC33-1162)	DFHLE512
R410 Release Guide	(GC33-1161)	DFHLE411
R410 Program Directory	(GC33-1200)	DFHLA906
R410 Installation Guide	(SC33-1163)	DFHLA110
R410 System Definition Guide	(SC33-1164)	DFHLA213
R410 System Program Ref	(SC33-1171)	DFHLA812
R410 Operations Guide	(SC33-1167)	DFHLA610
R410 Performance Guide	(SC33-1183)	DFHLT314
R410 CICS-Supplied Trans	(SC33-1168)	DFHLA711

Figure 1: Recommended manuals

then began adding the OEM (vendor) software products, exits, and in-house software code that were upgraded to run with CICS/ESA 4.1.

I turned off features such as transaction isolation and storage protection and added the in-house application loads. Checking the performance and storage, running statistics, then adding tools like TMON, ABEND AID, INTERTEST, and BMS, etc, I installed each product one at a time so that it would be easier to determine which product caused problems during that CICS run. You should document every step of

Transaction-id	Description
CEBR	Temporary-storage browse
CECI	Command level interpreter
CECS	Command level interpreter
CEDA	Resource definition online
CEDB	Resource definition online
CEDC	Resource definition online
CEDF	Execution diagnostic facility
CEMT	Master terminal
CEOT	Terminal status
CESC	Terminal timeout
CESF	Sign off
CESN	Sign on
CEST	Supervisory terminal
CETR	Trace
CHLP	Alias for CMAC
CMAC	Messages and Codes
CMSG	Message switching
CPLT	PLT programs
CRTE	Transaction routing
CSFE	Terminal test, trace, storage
CWTO	Write to console operator

Figure 2: Transactions to be familiar with

the way because you do not want to debug the same problem all over again when you move to the next CICS region.

Tip: you should go to the current maintenance level of CICS/ESA 4.1 to avoid a further upgrade.

MACRO LEVEL

You should plan for dealing with discontinued functions such as macro level DFHTM, DFHIS, DFHTD, SIMODS, and EXEC CICS ADDRESS CSA.

To get an inventory of macro-level usage by type and language, execute DFHMSCAN at the latest CICS level. DFHMSCAN reports the following:

- $SUMMARY lists:
  - All programs and sizes.
  - Their language (not always).
  - The number of macro-level calls.
  - The number of command-level requests.
  - The number of BALR 14,14 instructions (might not be ML call).
  - The number of BALR 14,15 instructions (might not be ML call).
  - Comments (such as ‘possible address CSA’).
- Totals at the end of the report list:
  - The number of modules scanned.
  - The number of macro-level programs.
  - The number of Assembler, COBOL, and PL/I programs.
  - The number that addressed the CSA.

Note: when it comes to performance, bear in mind you will use more CPU cycles simply by going from macro- to command-level instructions.

DFHCSD – RESOURCE DEFINITION

When upgrading your CSD, you must find and remove all IBM-supplied transactions (starting with C) and programs (starting with

DFH) from your personalized groups (non-DFH). The DFHCSDUP upgrade will not modify IBM PPT and PCT entries that are located in your personal groups, so delete them or make sure you get the correct entry after the upgrade from DFHLIST.

I do not recommend having CICS management modules and transactions defined in your own personalized groups, otherwise you will get burnt sooner or later. For example, transaction CLS1 no longer points to DFHLUP, error DFHZC4921C (MRO).

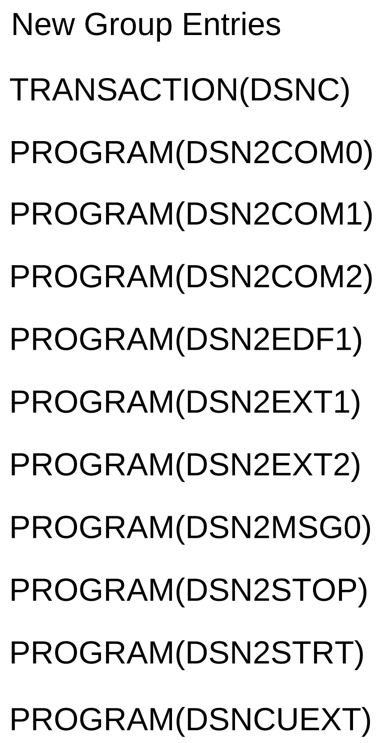
Since CICS has a feature called Progressive Program Compression to help eliminate short on storage conditions, you should review the programs you have defined as resident. There should be no need for them to be resident, based on their high use count. Because of the FCT RDO definitions in your CSD, you should make the size of DFHCSD larger because there is now an optional description field on the entry. If the CSD is to be shared between different CICS versions from 1.7 and up, you have to add groups DFHCOMP1, DFHCOMP2, DFHCOMP3, and DFHCOMP4 compatibility.

DB2 ATTACHMENT FACILITY

If you are running DB2 with CICS/ESA 4.1, then you must make sure that you install the new DFHDB2 CICS-supplied RDO group and remove the old group which contains the programs starting with prefix DSNC**** (see Figure 3).

Program DSNCEXT1 has changed to DSN2EXT1. Any program doing an EXTRACT EXIT with the name DSNCEXT1 will receive an INVEXITREQ error from CICS. In our IBM IMAGING environment, we had some customized code called IBM HOST PRINT running with IODM, which was no longer supported at that level and the source was not available. IBM HOST PRINT (program HPF2000P) needed to be ZAPped with the following in order to get by the error.

When I dumped out HPF2000P, module DSNCEXT1 was an EYECATCHER and only then could I create the following ZAP after dumping the module with proper offsets. The point is that you can ZAP a module you do not have source for by changing one letter from



New Group Entries

TRANSACTION(DSNC)

PROGRAM(DSN2COM0)

PROGRAM(DSN2COM1)

PROGRAM(DSN2COM2)

PROGRAM(DSN2EDF1)

PROGRAM(DSN2EXT1)

PROGRAM(DSN2EXT2)

PROGRAM(DSN2MSG0)

PROGRAM(DSN2STOP)

PROGRAM(DSN2STRT)

PROGRAM(DSNCUEXT)

Figure 3: New group entries

‘C’ to ‘2’ by finding the EYECATCHER DSNCEXT1. The command in HPF2000P is:

```
EXTRACT EXIT PROGRAM('DSNCEXT1') ENTRY ('DSNCSQL')
```

which should be:

```
EXTRACT EXIT PROGRAM('DSN2EXT1') ENTRY ('DSNCSQL')
```

Therefore, this is the ZAP I created to ZAP DSNCEXT1 to DSN2EXT1 within module HPF2000P.

```
//Z JOB USER=*,CLASS=I,MSGCLASS=X
//*
//HOSTPRT  EXEC PGM=IMASPZAP,PARM=(IGNIDRFULL)
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DSN=SYS1.HPF.IODM.LOAD,DISP=SHR
//SYSIN     DD *
NAME HPF2000P HPF2000P
*
VER 0470 D5C3,C5E7
REP  0470 D5F2,C5E7
```

```
*
          Note: change the "C" to "2" in DSNCEXT1.
```

MORE DB2 CONSIDERATIONS

There are further DB2 considerations:

- RCT has a two-byte suffix. The RCT is changed from DSNCRCTx to DSN2CTxx.

  ```
  Parm specified ==>  INITPARM=(DSN2STRT='xx,yyyy')
  ```

  where ‘xx’ is the RCT suffix and ‘yyyy’ is the subsystem-id suffix.

- You should check out the new RCT parameters PLANI, PURGEC, and TXIDSO.
- You can specify SSID override in the DSNC STRT command as well as the RCT suffix.
- There are performance and debugging (traces) enhancements in the new attachment facility.
- SQL programs can run in the PLTPI after DSN2COM0 has run.
- The new attachment facility runs above the 16MB line.
- AEY9 abends can be avoided if you code the INQUIRE EXITPROGRAM within your code and issue a proper message to the users.
- Start-up PLTPI should link to DSN2COM0.
- Shut-down PLTSD should link to DSN2COM2.

Note: specifying DSN2COM1 in the shut-down PLT may result in a storage overwrite and the following message: ‘DFHAP0001 Abend code (0C4/AKEA) has occurred at offset X'2F8' in module DSN2COM1’.

SVC DUMPS

CICS/ESA is now in the SDUMP business and the DUMP tables control the DUMP action. You should code a PLT initialization program to suppress SVC dumps being taken to prevent unnecessary overhead and avoid the flooding of the SDUMP datasets. I have documented mine in *Suppressing CICS SVC dumps, CICS Update,* October 1997. You should also have your MVS people increase the number of SYS1.DUMP datasets and increase their sizes to 100 cylinders.

SECURITY

The CICS Sign-on table is no longer available under CICS/ESA Version 4.1. CICS is out of the security business. To secure your CICS environment, you will have to use an external security manager such as TSS, CA-ACF2, or RACF. We use TSS and found many problems in the area of MRO (performance and functionality). With Release 5.0 of TSS, all problems have been corrected.

PROGRAMMING INTERFACES

EXEC CICS commands are defined as being either Application Programmer (AP) commands or System Programmer (SP) commands (eg INQUIRE, SET, PERFORM), which require the new CICS translator 'SP' option on compile or assembly.

SP commands are normally used by the CICS system programmer, intended to fill the gaps left by the withdrawal of macro-level because the commands access CICS resources and can be security protected. The AP commands are used by the application programmer because they access USER resources and are intended to satisfy basic application needs.

The EXIT Programming Interface (XPI) is a new interface that allows access to CICS internal domains such as DUMP, TRACE, LOADER, and STORAGE MANAGER. The XPI and SPI commands are used within the global user exit program and will cause harmful results if not used properly.

You should review/rewrite your exits and re-assemble/re-compile all user-replaceable modules.

USER-REPLACEABLE MODULES

For user-replaceable modules, you should note the following:

- They must be written as command-level modules.
- They must be 31-bit addressable.
- Use R1, R13, R14, and R15 according to MVS conventions.

The modules are DFHPEP, DFHREST, DFHTEP (TCAM DCB), DFHZNEP (major change for IMAGING), DFHZATDX, DFHZATDY, DFHPGADX, DFHDYP, DFHJXJCO, DFHXJCC, and DFHDBUEX

OBSOLETE USER-REPLACEABLE MODULES

DFHACEE, DFHRTY (replaced by DFHREST,) and DFHUAKP are obsolete user-replaceable modules.

The TRUE interface has withdrawn the addresses of the CSA (UEPCSA) and TCA (UEPTCA). If you try to address these (now) reserved fields through your TRUE program you will get an ASRD abend (also true for GLUEs).

Global user exit	Replaced by
XFCIN/XFCINC	XFCREQ/XFCSREQ (different functions)
XFCOUT/XFCOUTC	XFCREQC/XFCSREQC (different function)
XKCAWT/XKCBWT	XDSAWT/XDSBWT (Dispatcher Domain)
XKCDISP	(DFHKCP obsolete and handled by TC dispatch)
XSCREQ	(DFHSCP obsolete and handled by the Storage Domain)
XTCRDAT	DFHTCP is handled by Terminal Control)
XTDCOUT	XTDEREQ/XTDEREQC

Figure 4: GLUE replacements

GLUEs are activated via the EXEC CICS ENABLE (PLT). All TRUEs and GLUEs must be re-assembled.

Replacement GLUEs are shown in Figure 4. New GLUEs in CICS/ ESA 4.1 (some introduced in CICS/ESA 3.3) are shown below. Xnn***** nn is the two-byte component or Domain identifier:

- XAKUSER – before the end of keypoint write.
- XALCAID – when an AID with data is cancelled.
- XDLIPRE/XDLIPOST – DL/I interface
- XDSAWT/XDSBWT – Dispatcher Domain (after/before an MVS wait).
- XDTAD/XDTLC/XDTRD – Data tables.
- XDUREQ/XDUOUT/XDUCLSE – Dump Domain.
- XFCNREC – File open/close.
- XFCREQ/XFCREQC – File control.
- XICEREQ/XICEREQC – before/after an ICP ATI request.
- XJCWB/XJCWR – Journal control.
- XMEOUT – before a message is sent to a destination.
- XMNOUT – Monitor Domain.
- XPCHAIR – before HANDLE ABEND.
- XPCREQ/XPCREQC – entry/exit to a PC LINK.
- XPCTA – after an ABEND occurs.
- XSNOFF/XSNON – signoff/signon.
- XSRAB – SRT.
- XSTOUT – Statistics Domain.
- XSTERM – during normal shut-down.
- XSTOUT – before writing to SMF.
- XSZARQ/XSZBRQ – after/before FEPI.

- XTDEREQ/XTDEREQC – before/after a TD API request.
- XTSEREQ/XTSEREQC – before/after a TS API request.
- XRMIN/XRMIOUT – RMI command, DL/I or SQL.
- XXDFA/XXDFB/XXDTO – XRF.
- XXMATT – Transaction Manager (transaction attach).
- XXDFA/XXDFB/XXDTO – DBCTL in XRF.
- XZIQUE – allocate request to be queued.

The following is the source code to disable and enable an exit while the CICS region is running:

```
DFHEISTG DSECT                       THIS TRANSACTION'S STORAGE
WORKAREA DS    D                       WORKAREA EYECATCHER
         SPACE 1
         EJECT
*----------------------------------
*        MAIN LINE
*----------------------------------
EXITNEW  CSECT
         B     BEGIN
         DC    C'EXITNEW'
         DC    C'ASSEMBLED  ON  '
         DC    C'&SYSDATE'
         DC    C'  AT  '
         DC    C'&SYSTIME'
         DC    CL2'Ø1'
         SPACE 2
BEGIN    DS    ØH
         MVC   WORKAREA,=CL8'EXITNEW' PUT EYECATCHER ON WORKAREA
         EXEC CICS HANDLE CONDITION INVEXITREQ(INVEXIT)
         EXEC CICS IGNORE CONDITION INVREQ
         EXEC CICS DISABLE PROGRAM('I5ZCATT') ENTRYNAME('I5ZCATT')     X
              EXIT('XZCATT')
*        EXEC CICS DELAY INTERVAL(1Ø)
         EXEC CICS RELEASE PROGRAM('I5ZCATT')
         EXEC CICS SET PROGRAM ('I5ZCATT') PHASEIN ENABLED
         EXEC CICS ENABLE PROGRAM ('I5ZCATT') ENTRYNAME('I5ZCATT')     X
              EXIT('XZCATT') START
         EXEC CICS WRITE OPERATOR                                        X
              TEXT(MSGOUT) TEXTLENGTH(MSGLEN)                          X
              NOHANDLE
         B    RETURN
INVEXIT  EQU  *
         EXEC CICS WRITE OPERATOR                                        X
```

```
                  TEXT(INVMSG) TEXTLENGTH(INVMSGL)
RETURN   EQU  *
         EXEC CICS RETURN
*-----------------------------------
*        CONSTANTS
*-----------------------------------
         DS    ØF
MSGLEN   DC    A(MSGEND-MSGOUT)
MSGOUT   DC    C'EXIT I5ZCATT RELOAD SUCCESSFULLY'
MSGEND   EQU   *
INVMSGL  DC    A(INVEND-INVMSG)
INVMSG   DC    C'ENABLE EXIT I5ZCATT - INVEXITREQ'
INVEND   EQU   *
         END
```

STATISTICS

SMF dataset sizes will have to increase. All CICS shut-down and interval statistics are written to SMF. The CICS Monitor Facility writes out SMF 110 records that contain statistical and performance information. The SMF header records have changed for CICS 4.1.

Any customized programs that read and process these records must be changed to reference the new fields (eg OPID is not in the 110 SMF record). You can set up a batch DFHSTUP job which can execute each day to gather statistics and store them in GDG datasets for analysis.

SVCS

You should copy the new type 3 SVC to LPALIB (do not link-edit) and remove the CICS/MVS 2.1.2 type 2 SVC from the nucleus once you no longer need this release.

Make sure you copy SVCs DFHCSVC (type 3 SVC), DFHHPSVC (type 6 SVC), DFHIRP (MRO), and DFHDSPEX (post exit ‘stub’) from the new supplied LPA library coming from the CICS/ESA 4.1 release even if you are still running CICS/MVS 2.1.2 or lower.

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

Joe DiFranco
Senior Systems Programmer
Workplace Safety and Insurance Board of Ontario (Canada)

Further CICS V3.3 shutdown statistics – part 3

This month we conclude the programs to accumulate statistics for DTB, dumps, ISC/IRC, Task Class, VTAM, and Transient Data Queues.

TASK CLASS STATISTICS PROGRAM

```
         TITLE ' STATTCLS - TASK CLASS STATISTICS'
*
DFHEISTG DSECT
         DS    ØF
RESP     DS    F
CVRTAREA DS    D
ZAPTAREA DS    PL3
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
*
COMMFLDS DSECT
APPLID   DS    CL8
SYSID    DS    CL4
JOBNAME  DS    CL8
DATE     DS    CL8
TIME     DS    CL8
STOKEN   DS    CL8
*                 **  TCLASS GLOBAL STATISTICS **
         COPY  DFHA15DS
*
STATTCLS DFHEIENT CODEREG=(3),DATAREG=(13),EIBREG=11
*
         L    R2,DFHEICAP
         USING COMMFLDS,R2
         BAL   R4,HDNG          PAGE HEADINGS
         BAL   R4,FRSTHEAD       TCLASS STATISTICS HEADINGS
         BAL   R8,TCLS           TCLASS STATISTICS DETAIL
HDNG     EQU   *
         MVC   HJOBNM(8),JOBNAME
         MVC   HAPPLID(8),APPLID
         MVC   HSYSID(4),SYSID
         MVC   HDATE(8),DATE
         MVC   HTIME(8),TIME
         MVC   PRINTLN(133),HEADING
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),UNDRSCOR
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),BLANKS
         BAL   R1Ø,WRITESPL
         ZAP   LNECNT,=P'3'
         BR    R4
**  **
*
****>>>>>>>> **** START PROCESS TCLASS STATISTICS ****** <<<<<<<<*****
FRSTHEAD EQU    *
*        MVI   VTCNTL,C'1'
         MVC   PRINTLN(133),TCLHDT
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),BLANKS
         BAL   R1Ø,WRITESPL
         AP    LNECNT,=P'2'
TCLSHEAD EQU    *
         MVC   PRINTLN(133),TCLHD1
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),TCLHD2
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),TCLHD3
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),UNDRSCOR
         BAL   R1Ø,WRITESPL
         AP    LNECNT,=P'3'
         BR    R4
TCLS     EQU   *
         USING DFHA15DS,R9
         LA     R11,1Ø
         LA     R7,TCLASTBL
MVCTCLAS MVC    TCLASS,Ø(R7)
*
         EXEC CICS COLLECT STATISTICS TCLASS (TCLASS) SET (R9)
*
*   TASK CLASS NUMBER
         L     R6,A15KTCLS
         CVD   R6,CVRTAREA
         ZAP   ZAPTAREA(3),CVRTAREA+5(3)
         OI    ZAPTAREA+2,X'ØF'
         MVC   TCNUM(6),PTRN3
         ED    TCNUM(6),ZAPTAREA
*   MAX NUMBER OF TASKS
         MVC   TCMAX(6),PTRN2
         ED    TCMAX(6),A15MXT
*   TIMES AT MAX TASK
         MVC   TCTMAX(6),PTRN3
         ED    TCTMAX(6),A15MXTM
*   CURRENT TASKS
```

```
         MVC   TCCURR(6),PTRN3
         ED    TCCURR(6),A15MXTC
*   PEAK TASKS
         MVC   TCPEAK(6),PTRN3
         ED    TCPEAK(6),A15MXTR
         MVC   PRINTLN(133),TCLNE1
         BAL   R1Ø,WRITESPL
         LA    R7,4(R7)
         BCT   R11,MVCTCLAS
*
DTBEND   EQU   *
         MVC   PRINTLN(133),UNDRSCOR
         BAL   R1Ø,WRITESPL
*
EXIT     EQU   *
*                        ** RETURN  **
         EXEC  CICS RETURN
*
****>>>>>>>>***** END PROCESS TCLASS STATISTICS  *******<<<<<<<<*******
*                        ** WRITE THE SPOOL RECORD  **
WRITESPL EQU   *
         EXEC CICS SPOOLWRITE TOKEN(STOKEN) FROM(PRINTLN)              -
               FLENGTH(LINELEN) NOHANDLE
         BR    R1Ø
********************************************************************
         DS    ØF
TCLASS   DS    F
TCLASTBL EQU   *
         DC    F'1'
         DC    F'2'
         DC    F'3'
         DC    F'4'
         DC    F'5'
         DC    F'6'
         DC    F'7'
         DC    F'8'
         DC    F'9'
         DC    F'1Ø'
LNECNT   DC    PL2'Ø'
MAXLNE   DC    P'6Ø'
PTRN2    DC    X'4Ø2Ø2Ø212Ø'
PTRN3    DC    X'4Ø2Ø2Ø2Ø212Ø'
OUTCLAS  DC    CL1'Q'
LINELEN  DC    F'133'
PRINTLN  DS    CL133
BLANKS   DS    ØCL133     ** BLANK LINE  **
         DC    CL1'Ø'
         DC    CL132' '
UNDRSCOR DS    ØCL133     ** UNDERSCORE LINE  **
         DC    CL1'Ø'
         DC    132C'_'
*                          ** HEADING LINE DEFINITION **
HEADING  DS    ØCL133
         DC    CL1'1'
         DC    CL8'JOBNAME:'
HJOBNM   DC    CL8' '
         DC    CL1Ø'   APPLID:'
HAPPLID  DC    CL8' '
         DC    CL9'   SYSID:'
HSYSID   DC    CL4' '
         DC    CL8'   DATE:'
HDATE    DC    CL8' '
         DC    CL8'   TIME:'
HTIME    DC    CL8' '
         DC    CL53' '
** **
* ***************** START TCLASS STATISTICS DEFINITIONS ************
TCLNM    DS    CL8
*                          ** TCLASS HEADING LINES **
TCLHDT   DS    ØCL133
TCHCNTLA DC    CL1'Ø'
         DC    CL33'***   TASK CLASS STATISTICS   ***'
         DC    CL99' '
TCLHD1   DS    ØCL133
TCHCNTL  DC    CL1'Ø'
TCLN1H   DC    CL132'             MAXIMUM             NUMBER OF       N-
               UMBER OF       PEAK NO.'
TCLHD2   DS    ØCL133
         DC    CL1'Ø'
TCLN2H   DC    CL132'                  NUMBER OF       TIMES AT       C-
               URRENT         OF TASKS'
*
TCLHD3   DS    ØCL133
         DC    CL1'Ø'
TCLN3H   DC    CL132'                    TASKS        MAX. TASK        -
               TASKS          REACHED '
*                          ** TASK CLASS DETAIL LINE  **
TCLNE1   DS    ØCL133
TCDCNTL  DC    CL1'Ø'
TCID     DC    CL1Ø'TASK CLASS'
TCNUM    DS    CL6
         DC    CL4' '
TCMAX    DS    CL6
         DC    CL6' '
TCTMAX   DS    CL6
         DC    CL9' '
TCCURR   DS    CL6
         DC    CL9' '
TCPEAK   DS    CL6
         DC    CL65' '
```

```
* ****************** END TCLASS STATISTICS DEFINITIONS   ************
**
         LTORG
         DFHEISTG
         DFHEIEND
         END
```

VTAM STATISTICS PROGRAM

```
         TITLE ' STATVTAM - VTAM STATISTICS COLLECTION PROGRAM'
DFHEISTG DSECT
         DS    ØF
RESP     DS    F
VSTATUS  DS    F
FULLWORD DS    F
CVRTAREA DS    D
ZAPTAREA DS    PL3
ZAPTARE2 DS    PL2
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
COMMFLDS DSECT
APPLID   DS    CL8
SYSID    DS    CL4
JOBNAME  DS    CL8
DATE     DS    CL8
TIME     DS    CL8
STOKEN   DS    CL8
*                  **  VTAM GLOBAL STATISTICS  **
         COPY  DFHAØ3DS
*
STATVTAM DFHEIENT CODEREG=(3),DATAREG=(13),EIBREG=11
         L    R2,DFHEICAP
         USING COMMFLDS,R2
         BAL   R4,HDNG          PAGE HEADINGS
         BAL   R4,FRSTHEAD       VTAM STATISTICS HEADINGS
         BAL   R8,VTAMS          VTAM STATISTICS DETAIL
HDNG     EQU   *
         MVC   HJOBNM(8),JOBNAME
         MVC   HAPPLID(8),APPLID
         MVC   HSYSID(4),SYSID
         MVC   HDATE(8),DATE
         MVC   HTIME(8),TIME
         MVC   PRINTLN(133),HEADING
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),UNDRSCOR
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),BLANKS
         BAL   R1Ø,WRITESPL
         ZAP   LNECNT,=P'3'
         BR    R4
**  **
****>>>>>>>> **** START PROCESS VTAM STATISTICS ****** <<<<<<<<*******
FRSTHEAD EQU    *
*        MVI   VTCNTL,C'1'
         MVC   PRINTLN(133),VTAMHDT
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),BLANKS
         BAL   R1Ø,WRITESPL
         AP    LNECNT,=P'2'
VTAMS    EQU   *
         EXEC  CICS INQUIRE VTAM OPENSTATUS(VSTATUS)
         USING DFHAØ3DS,R9
         EXEC CICS COLLECT STATISTICS VTAM SET (R9)
*   MAXIMUM NUMBER OF RPLS POSTED
         MVC   VTLN1D+4(4),PTRN2
         ED    VTLN1D+4(4),AØ3RPLX
         MVC   PRINTLN(133),VTAMHD1
         BAL   R1Ø,WRITESPL
*   NUMBER OF TIMES REACHED MAXIMUM
         MVC   VTLN2D(8),PTRN4
         ED    VTLN2D(8),AØ3RPLXT
         MVC   PRINTLN(133),VTAMHD2
         BAL   R1Ø,WRITESPL
*   VTAM SHORT ON STORAGE
         LH    R6,AØ3VTSOS
         CVD   R6,CVRTAREA
         ZAP   ZAPTAREA(3),CVRTAREA+5(3)
         OI    ZAPTAREA+2,X'ØF'
         MVC   VTLN3D+2(6),PTRN3
         ED    VTLN3D+2(6),ZAPTAREA
         MVC   PRINTLN(133),VTAMHD3
         BAL   R1Ø,WRITESPL
*   VTAM DYNAMIC OPEN COUNT
         LH    R6,AØ3DOC
         CVD   R6,CVRTAREA
         ZAP   ZAPTAREA(3),CVRTAREA+5(3)
         OI    ZAPTAREA+2,X'ØF'
         MVC   VTLN4D+2(6),PTRN3
         ED    VTLN4D+2(6),ZAPTAREA
         MVC   PRINTLN(133),VTAMHD4
         BAL   R1Ø,WRITESPL
VTAMEND  EQU   *
         MVC   PRINTLN(133),UNDRSCOR
         BAL   R1Ø,WRITESPL
```

```
EXIT     EQU   *
*                        ** RETURN  **
         EXEC  CICS RETURN
****>>>>>>>>***** END PROCESS VTAM STATISTICS  *******<<<<<<<<*******
*                        ** WRITE THE SPOOL RECORD  **
WRITESPL EQU   *
         EXEC CICS SPOOLWRITE TOKEN(STOKEN) FROM(PRINTLN)              -
               FLENGTH(LINELEN) NOHANDLE
         BR    R1Ø
********************************************************************
LNECNT   DC    PL2'Ø'
MAXLNE   DC    P'6Ø'
PTRN2    DC    X'4Ø2Ø212Ø'
PTRN3    DC    X'4Ø2Ø2Ø2Ø212Ø'
PTRN4    DC    X'4Ø2Ø2Ø2Ø2Ø2Ø212Ø'
OUTCLAS  DC    CL1'Q'
LINELEN  DC    F'133'
PRINTLN  DS    CL133
BLANKS   DS    ØCL133     ** BLANK LINE  **
         DC    CL1'Ø'
         DC    CL132' '
UNDRSCOR DS    ØCL133     ** UNDERSCORE LINE  **
         DC    CL1'Ø'
         DC    132C'_'
*                         ** HEADING LINE DEFINITION **
HEADING  DS    ØCL133
         DC    CL1'1'
         DC    CL8'JOBNAME:'
HJOBNM   DC    CL8' '
         DC    CL1Ø'   APPLID:'
HAPPLID  DC    CL8' '
         DC    CL9'   SYSID:'
HSYSID   DC    CL4' '
         DC    CL8'   DATE:'
HDATE    DC    CL8' '
         DC    CL8'   TIME:'
HTIME    DC    CL8' '
         DC    CL53' '
** **
* ***************** START VTAM STATISTICS DEFINITIONS  ************ **
VTAMNM   DS    CL8
*                         ** VTAM HEADING LINES **
VTAMHDT  DS    ØCL133
VTHCNTLA DC    CL1'Ø'
         DC    CL38'***   VTAM STATISTICS   ***'
         DC    CL94' '
VTAMHD1  DS    ØCL133
VTHCNTL  DC    CL1'Ø'
VTLN1H   DC    CL38' MAXIMUM NUMBER OF RPLS POSTED'
VTLN1D   DC    CL94' '
VTAMHD2  DS    ØCL133
         DC    CL1'Ø'
```

```
VTLN2H   DC    CL38' NUMBER OF TIMES REACHED MAXIMUM'
VTLN2D   DC    CL94' '
VTAMHD3  DS    ØCL133
VTHCNT3  DC    CL1'Ø'
VTLN3H   DC    CL38' VTAM SHORT ON STORAGE '
VTLN3D   DC    CL94' '
VTAMHD4  DS    ØCL133
         DC    CL1'Ø'
VTLN4H   DC    CL38' VTAM DYNAMIC OPEN COUNT'
VTLN4D   DC    CL94' '
*
* ******************* END VTAM STATISTICS DEFINITIONS  ************ **
*
         LTORG
         DFHEISTG
         DFHEIEND
         END
```

TRANSIENT DATA STATISTICS PROGRAM

```
         TITLE ' STATTDQS - TRANSIENT DATA STATISTICS'
DFHEISTG DSECT
         DS    ØF
RESP     DS    F
CVRTAREA DS    D
ZAPTAREA DS    PL3
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
COMMFLDS DSECT
APPLID   DS    CL8
SYSID    DS    CL4
JOBNAME  DS    CL8
DATE     DS    CL8
TIME     DS    CL8
STOKEN   DS    CL8
*                         **  TRANSIENT DATA DEFINITION DSECT **
         COPY  DFHA1ØDS
STATTDQS DFHEIENT CODEREG=(3),DATAREG=(13),EIBREG=11
         L    R2,DFHEICAP
         USING COMMFLDS,R2
         BAL   R4,HDNG          PAGE HEADINGS
         BAL   R4,FRSTHEAD      TDQ HEADINGS
         BAL   R8,TDQUES        TRANSIENT DATA STATISTICS DETAIL
         B     EXIT
HDNG     EQU   *
         MVC   HJOBNM(8),JOBNAME
         MVC   HAPPLID(8),APPLID
         MVC   HSYSID(4),SYSID
         MVC   HDATE(8),DATE
         MVC   HTIME(8),TIME
         MVC   PRINTLN(133),HEADING
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),UNDRSCOR
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),BLANKS
         BAL   R1Ø,WRITESPL
         ZAP   LNECNT,=P'3'
         BR    R4
**  **
FRSTHEAD EQU    *
         MVC   PRINTLN(133),BLANKS
*        MVI   TDCNTL,C'1'
         MVC   PRINTLN(133),TDQUEHD
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),BLANKS
         BAL   R1Ø,WRITESPL
         AP    LNECNT,=P'2'
         MVC   PRINTLN(133),BLANKS
         MVC   PRINTLN(133),TDQUEHD1
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),BLANKS
         MVC   PRINTLN(133),TDQUEHD2
         BAL   R1Ø,WRITESPL
         MVC   PRINTLN(133),BLANKS
         BAL   R1Ø,WRITESPL
         AP    LNECNT,=P'3'
         MVC   PRINTLN(133),BLANKS
         BR    R4
*
****>>>>>>>> **** START PROCESS TRANSIENT DATA STATISTICS ** <<<<<<<<*
*
TDQUES   EQU   *
         USING DFHA1ØDS,R9
         EXEC CICS INQUIRE TDQUEUE START
         EXEC CICS INQUIRE TDQUEUE (TDQUEUE) NEXT
         BAL  R7,TDQSCK
NXTTDQ   EQU   *
         EXEC CICS INQUIRE TDQUEUE (TDQUEUE) NEXT RESP(RESP)
         CLC   RESP(4),DFHRESP(END)
         BE    TDQEND
         BAL   R7,TDQSCK
         B     NXTTDQ
```

```
TDQEND   EQU   *
         EXEC CICS INQUIRE TDQUEUE END
         B     MVULINE
TDQSCK   EQU   *
         EXEC CICS COLLECT STATISTICS TDQUEUE (TDQUEUE) SET (R9)
         MVC   PRINTLN(133),BLANKS
         MVC   TDLNE1(133),BLANKS
         MVC   TDQID(4),A1ØDEST    TRANSIENT DATA QUEUE ID
         CLI   A1ØTYPE,X'Ø1'       DETERMINE TYPE OF QUEUE:
         BE    TDEXTRA              X'Ø1' = EXTRAPARTITION
         CLI   A1ØTYPE,X'Ø2'        X'Ø2' = INTRAPARTITION
         BE    TDINTRA              X'Ø3' = INDIRECT
         CLI   A1ØTYPE,X'Ø3'        X'Ø4' = REMOTE
         BE    TDINDIR             AND GO TO APPROPRIATE
         CLI   A1ØTYPE,X'Ø4'       ROUTINE.
         BE    TDREMOT
         B     BR7
TDEXTRA  EQU   *
         L     R6,A1ØEO            EXTRAPARTITION OUTPUT
         CVD   R6,CVRTAREA
         ZAP   ZAPTAREA(3),CVRTAREA+5(3)
         OI    ZAPTAREA+2,X'ØF'
         CP    ZAPTAREA,=P'ØØØ'
         BE    BR7
         MVC   EXTROUT(6),PTRN3
         ED    EXTROUT(6),ZAPTAREA
         B     MVLINE
TDINTRA  EQU   *
         L     R6,A1ØIO            INTRAPARTITION OUTPUT
         CVD   R6,CVRTAREA
         ZAP   ZAPTAREA(3),CVRTAREA+5(3)
         OI    ZAPTAREA+2,X'ØF'
         CP    ZAPTAREA,=P'ØØØ'
         BE    BR7
         MVC   INTROUT(6),PTRN3
         ED    INTROUT(6),ZAPTAREA
         B     MVLINE
TDINDIR  EQU   *
         L     R6,A1ØIR            INDIRECT REQUESTS
         CVD   R6,CVRTAREA
         ZAP   ZAPTAREA(3),CVRTAREA+5(3)
         OI    ZAPTAREA+2,X'ØF'
         CP    ZAPTAREA,=P'ØØØ'
         BE    BR7
         MVC   INDIRCT(6),PTRN3
         ED    INDIRCT(6),ZAPTAREA
         B     MVLINE
TDREMOT  EQU   *
         L     R6,A1ØRR            REMOTE REQUESTS
         CVD   R6,CVRTAREA
         ZAP   ZAPTAREA(3),CVRTAREA+5(3)
         OI    ZAPTAREA+2,X'ØF'
         CP    ZAPTAREA,=P'ØØØ'
         BE    BR7
         MVC   REMOTE(6),PTRN3
         ED    REMOTE(6),ZAPTAREA
         B     MVLINE
MVLINE   EQU   *
         MVC   PRINTLN(133),BLANKS
         MVC   PRINTLN(133),TDLNE1
         BAL   R1Ø,WRITESPL
BR7      MVC   PRINTLN(133),BLANKS
         BR    R7
*
MVULINE  MVC   PRINTLN(133),UNDRSCOR
         BAL   R1Ø,WRITESPL
         BR    R8
EXIT     EQU   *
*                        ** RETURN  **
         EXEC  CICS RETURN
*
****>>>>>>>>***** END PROCESS TRANSIENT DATA STATISTICS ***<<<<<<<<*
*                        ** WRITE THE SPOOL RECORD  **
WRITESPL EQU   *
         EXEC CICS SPOOLWRITE TOKEN(STOKEN) FROM(PRINTLN)              -
               FLENGTH(LINELEN) NOHANDLE
         BR    R1Ø
********************************************************************
*
         DS    ØF
TDQUEUE  DS    CL4
LNECNT   DC    PL2'Ø'
MAXLNE   DC    P'6Ø'
PTRN3    DC    X'4Ø2Ø2Ø2Ø212Ø'
OUTCLAS  DC    CL1'Q'
LINELEN  DC    F'133'
PRINTLN  DS    CL133
BLANKS   DS    ØCL133     ** BLANK LINE  **
         DC    CL1'Ø'
         DC    CL132' '
UNDRSCOR DS    ØCL133     ** UNDERSCORE LINE  **
         DC    CL1'Ø'
         DC    132C'_'
*                         ** HEADING LINE DEFINITION **
HEADING  DS    ØCL133
         DC    CL1'1'
         DC    CL8'JOBNAME:'
HJOBNM   DC    CL8' '
         DC    CL1Ø'   APPLID:'
HAPPLID  DC    CL8' '
```

```
         DC    CL9'   SYSID:'
HSYSID   DC    CL4' '
         DC    CL8'   DATE:'
HDATE    DC    CL8' '
         DC    CL8'   TIME:'
HTIME    DC    CL8' '
         DC    CL53' '
** **
* ***************** START TRANSIENT DATA STATISTICS DEFINITIONS **
*                         ** TRANSIENT DATA HEADING LINES **
TDQUEHD  DS    ØCL133
TDHCNTLA DC    CL1'Ø'
         DC    CL37'***   TRANSIENT DATA STATISTICS   ***'
         DC    CL95' '
TDQUEHD1 DS    ØCL133
TDHCNTL  DC    CL1'Ø'
TDLN1H   DC    CL64' DESTINATION  EXTRAPARTITION  INTRAPARTITION  INDIR-
               ECT    REMOTE'
TDLN1D   DC    CL72' '
TDQUEHD2 DS    ØCL133
         DC    CL1'Ø'
TDLN2B1  DC    CL3' '
TDLN2H   DC    CL64'IDENT        REQUESTS        OUTPUTS       REQUESTS-
                   REQUESTS '
TDLN2D   DC    CL55' '
*
* ***************** START TRANSIENT DATA STATISTICS DEFINITIONS ***
*                         ** TASK CLASS DETAIL LINE  **
TDLNE1   DS    ØCL133
TDDCNTL  DC    CL1'Ø'
         DC    CL4' '
TDQID    DS    CL4
         DC    CL8' '
EXTROUT  DS    CL6
         DC    CL8' '
INTROUT  DS    CL6
         DC    CL8' '
INDIRCT  DS    CL6
         DC    CL8' '
REMOTE   DS    CL6
         DC    CL68' '
* ******************* END TRANSIENT DATA STATISTICS DEFINITIONS ****
*
         LTORG
         DFHEISTG
         DFHEIEND
         END
```

Jim Smith
System Programmer
Onondaga County Data Processing (USA)

CICS news

Blue Lobster Software has released Mako 2.0, enabling Java developers to generate Legacy Business Objects (LBOs), which are JavaBeans that map CICS transactions between COBOL and Java. LBOs are incorporated into new applications using drag-and-drop programming in third-party Java IDEs.

With the integration of Security Integration's Security Bridge product, Mako 2.0 also provides mainframe security integration for Internet-enabled CICS legacy applications. Mako 2.0 uses certified mainframe security services and extends data delivery options, eliminating the need to maintain security information for users on multiple platforms.

Specific new features include an LBO Builder that enables mainframe transactions for Java and the Web through automatic generation of JavaBeans from COBOL transactions. An enhanced server component maps application requests for CICS transactions between COBOL and Java.

For further information contact:
Blue Lobster, 2005 Hamilton Avenue, Suite 270, San Jose, CA 95125, USA.
Tel: (408) 371 5300.
URL: http://www.bluelobster.com.

* * *

IBM has launched REXX for CICS development toolset, enabling CICS/ESA Version 4 users to use REXX programs to write and execute in a CICS region. The programs have access to most EXEC CICS commands, the CICS CEDA and CEMT transaction programs, and DB2 databases via the EXEC SQL interface.

The tools include features for CICS system programmers, and CICS and DB2 administrators, programmers, developers, support staff, and users.

Features include a general-purpose, CICS-based text editor, a file system for storing text files and EXECs, a file list utility that provides a full-screen interface to the file system, and run-time facilities.

For further information contact your local IBM representative.

* * *

CICS users can benefit from Peritus Software Services' RQE Workbench tool for independent verification and validation (IV and V) of C and COBOL code which has been remediated for Y2K compliance.

Peritus has also expanded its IV and V services to support CICS, C, IDMS, and DB2, as well as RPG, PL/I, Assembler, Easytrieve, and Focus programs.

For further information contact:
Peritus Software Services, 2 Federal Street, Billerica, MA 01821-3540, USA.
Tel: (978) 670 0800.
URL: http://www.peritus.com.

* * *