159

CICS update

February 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

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.

Anchoring a WSA address using TRUE

This article explains how a Task Related User Exit (TRUE) can be used to anchor the address of a working storage area GETMAINed, once only, at the start of a transaction by a program, and then used throughout the life of the task by further calls to the same program, or other programs running under the same task.

This technique was designed to cut down on the code path of transactions that typically execute the same program separately but use the same working storage area each time the program is executed. The working storage area contained mainly static data and so it seemed pointless setting it up from scratch every time the program was executed. Also the ‘main’ program that used the working storage area would normally be executed hundreds of thousands of times in the life of one CICS AOR – so the CPU savings were not insignificant.

TECHNIQUE

Typically, the first call to the program in a task will GETMAIN a system working storage area, populate it, and then call the TRUE to store the address for later use by other programs in the task. When a program that utilizes the working storage area is executed again, it will first of all check to see whether the address of the working storage area has been stored away by the TRUE. If it has, the program will map and use the working storage area already obtained.

Note: when it was designed, this technique had to work for any transaction in the system that executed programs making use of this special working storage area. Therefore, it was not possible to utilize the IBM supplied TWA as an anchor point because some transactions were already using the TWA for their own purposes.

COMPONENTS

There are three components to put this technique into place:

1 A Task Related User Exit (TRUE).

2 A PLT program to enable the TRUE.

3 Application code to GETMAIN, populate, and reference the system working storage area.

CICS SOFTWARE RELEASE

This code was written specifically for CICS 4.1.0 and no attempt has been made to use it under any lower releases.

SOURCE CODE

The TRUE is coded as follows:

```
************************************************************
*   PROGRAM NAME:  TRUEXIT
*   DESCRIPTION:   TASK RELATED USER EXIT
*
*                  THIS PROGRAM IS USED TO EITHER RETRIEVE
*                  OR STORE THE ADDRESS OF A WORK AREA
*                  FROM/IN THE TASK WORK AREA.
************************************************************
*         REGISTER EQUATES
**************************
          REQU
***********************************
*         TASK PARAMETER LIST DSECT
***********************************
          DFHUEXIT TYPE=RM
**************************
*         CODE STARTS HERE
**************************
TRUEXIT   CSECT
TRUEXIT   AMODE 31                      SET AMODE 31.
TRUEXIT   RMODE ANY                     SET RMODE ANY.
**************************
*         ENTRY POINT
**************************
          STM   R14,R12,12(R13)         SAVE AWAY REGGIES
          LR    R11,R15                 LOAD ENTRY POINT
          USING TRUEXIT,R11             AND MAP
          B     MAINSØØØ                BRANCH AROUND 'EYE'
*************************
*         EYE CATCHER
*************************
          DC    C'.',C'CICS 4.1.Ø'      SYSTEM-ID.
          DC    C'.',C'TRUEXIT'         PROGRAM-ID.
          DC    C'.',C'V=Ø1, ML=ØØ'     PROGRAM VERSION.
          DC    C'.',C'TASK RELATED USER EXIT'
          DC    C'.',C'&SYSDATE'        DATE ASSEMBLED.
          DC    C'.',C'&SYSTIME'        TIME ASSEMBLED.
          DC    C'.'                    END OF PROGRAM-ID.
************************************************************
MAINSØØØ  DS    ØH
          USING DFHUEPAR,R1         MAP PASSED PARAMETERS
************************************************************
*         PROCESS PASSED PARAMETERS
*         NB TO SIMPLIFY THINGS THE ADDRESS OF THE
*         CALLERS WORK AREA WILL BE STORED IN R9.
*************************************************
MAINSØ4Ø  DS    ØH
          L     R1Ø,UEPTAA          ADDR OF LOCAL WORK AREA
          L     R8,UEPHMSA          GET CALLERS SAVE AREA
          L     R7,56(R8)           THEN CALLERS R9 INTO R7
          ICM   R9,15,Ø(R1Ø)        ANY WORK AREA STORED YET?
          BZ    MAINSØ8Ø            NO  STORE AWAY NEW ADDR
          ST    R9,56(R8)           YES STORE TA ADDR INTO R9
          B     MAINS999                AND EXIT
MAINSØ8Ø  DS    ØH
          ST    R7,Ø(R1Ø)           STORE R9 ADDR FROM SA
**************************
*         RETURN TO CALLER
**************************
MAINS999  DS    ØH
          LM    R14,R12,12(R13)     RESTORE REGS
          BR    R14                 AND RETURN TO CALLER
*************************
          LTORG
          END
```

The PLT program is as follows:

```
************************************************************
*   PROGRAM NAME:    TRUEENAB
*
*   DESCRIPTION:     MAINLINE CODE THAT RUNS AT CICS
*                    INITIALIZATION TO ENABLE THE TRUE.
************************************************************
**************************
*         REGISTER EQUATES
**************************
          REQU
*************************************
*         WORKING STORAGE DEFINITIONS
*************************************
DFHEISTG  DSECT
WSMESS    DS    CL5Ø                 CSMT MESSAGE FIELD
```

```
************************************************************
*         MAINLINE CODE
************************************************************
TRUEENAB  DFHEIENT CODEREG=(11),DATAREG=(1Ø),EIBREG=9
          B     MAINSØØØ                     BRANCH TO MAINLINE
          DC    C'.',C'CICS 4.1'             SYSTEM-ID.
          DC    C'.',C'TRUEENAB'             PROGRAM SOURCE NAME
          DC    C'.',C'V=Ø1,SML=Ø1'
          DC    C'.',C'PLT TRUE ENABLER'
          DC    C'.',C'&SYSDATE'             DATE ASSEMBLED.
          DC    C'.',C'&SYSTIME'             TIME ASSEMBLED.
MAINSØØØ  DS    ØH
          EXEC CICS HANDLE CONDITION INVEXITREQ(MAINS1ØØ)
          EXEC CICS ENABLE PROGRAM('TRUEXIT') TALENGTH(4) START
MAINSØ4Ø  DS    ØH
          MVC   WSMESS(5Ø),WDCMESS1
          EXEC CICS WRITEQ TD QUEUE('CSMT') FROM(WSMESS)
          B     MAINS999
MAINS1ØØ  DS    ØH
          MVC   WSMESS(5Ø),WDCMESS2
          EXEC CICS WRITEQ TD QUEUE('CSMT') FROM(WSMESS)
MAINS999  DS    ØH
          EXEC CICS RETURN
***************************************
*         CONSTANTS USED IN THIS PROGRAM
***************************************
WDCMESS1  DC    CL5Ø'TRUEENAB-IØ1 - TRUE EXIT NOW ENABLED'
WDCMESS2  DC    CL5Ø'TRUEENAB-EØ1 - ERROR ENABLING TRUE EXIT'
          LTORG
          END
```

Application code to call the TRUE exit will look something like the following:

```
***********************************************************************
*   PROGRAM NAME:   TRUECALL
*
*   DESCRIPTION:    SAMPLE PROGRAM TO CALL THE TRUE EXIT
***********************************************************************
*         REGISTER EQUATES
**************************
          REQU
***********************************
*         TASK PARAMETER LIST DSECT
***********************************
          DFHUEXIT TYPE=RM
***************************************
*         SPECIAL WORKING STORAGE DSECT
***************************************
WSAREA    DSECT
```

```
WSFLDØ1   DS    CLn
WSFLDØ2   DS    CLn
........
........
........
WSFLDnn   DS    CLn
WSAREAL   EQU   *-WSAREA

*************************************
*         WORKING STORAGE DEFINITIONS
*************************************
DFHEISTG  DSECT
..............
..............WORKING STORAGE FIELDS GO HERE
..............
************************************************************
*         MAINLINE CODE
************************************************************
TRUECALL  DFHEIENT CODEREG=(11),DATAREG=(1Ø),EIBREG=9
          B     MAINSØØØ                BRANCH TO MAINLINE
          DC    C'.',C'CICS 4.1'             SYSTEM-ID.
          DC    C'.',C'TRUECALL'        PROGRAM SOURCE NAME
          DC    C'.',C'V=Ø1,SML=Ø1'
          DC    C'.',C'TRUE CALLER'
          DC    C'.',C'&SYSDATE'        DATE ASSEMBLED.
          DC    C'.',C'&SYSTIME'        TIME ASSEMBLED.
*********************************************************
*         CALL TRUE TO SEE IF WE HAVE AN ANCHOR POINT YET
*********************************************************
MAINSØØØ  DS   ØH
          XR        R9,R9               ZEROIZE WORK AREA ADDR
          LA        R14,MAINSØ1Ø        LOAD RETURN ADDR FROM MACRO CALL
          DFHRMCAL TO=TRUEXIT,DSECTS=NO
MAINSØ1Ø  DS   ØH
          LTR       R9,R9               ANY WORK AREA ADDR RETURNED
          BNZ       MAINS1ØØ            YES - GO AND PROCESS
*********************************************************************
*         GET STORAGE FOR WORK AREA AND USE THE MACRO CALL TO
*         'TRUEXIT' TO STORE THE ADDRESS OF THE WORK AREA IN THE
*         TASK RELATED USER AREA.
*********************************************************************
MAINSØ2Ø  DS   ØH
          EXEC  CICS GETMAIN SET(R9) LENGTH(=Y(WSAREAL))  INITIMG(ZERO)
          LA        R14,MAINS1ØØ       LOAD RETURN ADDR FROM MACRO CALL
          DFHRMCAL TO=TRUEXIT,DSECTS=NO
********************************************
*         MAP THE WORKING STORAGE AREA
********************************************
MAINS1ØØ  DS   ØH
          USING WSAREA,R9
```

```
******************************************************
*         REST OF THE MAINLINE CODE GOES HERE
******************************************************
................

................

................

................

MAINS999  DS   ØH
          EXEC CICS RETURN
***************************************
*         CONSTANTS USED IN THIS PROGRAM
***************************************
          LTORG
          END
```

Simon Higgins
Blackbox Design Services (UK) © Xephon 1999

Analysing abended transactions – part 3

This month we continue the article that describes how to store and analyse abends that occur in a CICS region, as well as obtaining an immediate description using the CICS file DFHMAC.

```
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
*
* Program Dercode
*
         TITLE  'MACRO DEFINITIONS'
         MACRO                     MACRO HEADER
         CXID  &MEMBER,&R=          PROTYPE STATEMENT
         AGO   .PGNAME
.* G.RALLO TP 16/4/8Ø
.PGNAME  ANOP
.*
.*       THIS VARIABLE FOR TIME AND DATE STAMPING
         LCLC  &VMTMDT             TIME/DATE STAMP
         LCLC  &RELEASE            VERSION
.*
.*
        AIF  (T'&R NE 'O').SETR
```

```
&RELEASE     SETC 'Ø1Ø1'
        AGO  .DROP
.SETR   ANOP
&RELEASE     SETC '&R'
         SPACE 1
.DROP   ANOP
         PUSH PRINT
         PRINT GEN
************************************************************************
         DC    C'*',C' '
         DC    C'PROGRAM NAME:'
         DC    CL8'&MEMBER'  NAME
         DC    C' ',C'*',C' '
         DC    C'PROGRAM VERSION:'
         DC    CL4'&RELEASE'
         DC    C' '
         DC    C'*',C' '
         SPACE
         DC    C'ASSEMBLY TIME(HH.MM):'
&VMTMDT  SETC  '&SYSTIME'
         DC    C'&VMTMDT'          ASSEMBLY TIME (HH.MM) AND
         DC    C' '
         DC    C'ASSEMBLY DATE(MM/DD/YY):'
&VMTMDT  SETC  '&SYSDATE'
         DC    C'&VMTMDT'          DATE (MM/DD/YY) SAME AS LISTING
************************************************************************
         POP   PRINT
         MEXIT
         MEND
*=================================================================
         MACRO
*
*
*
         CSNAME &NAME
         GBLC  &CSECT
         AIF ('&NAME' EQ '').NONAME
&CSECT   SETC  '&NAME'
         AGO   .SC
.NONAME  ANOP
&CSECT   SETC  '&SYSECT'
.SC      ANOP
         PUSH  PRINT
         PRINT GEN
*========================================*
*                                        *
*                                        *
CSNAME   DC    CL8'&CSECT'
*                                        *
*                                         *
*=========================================*
         POP   PRINT
         MEND
         TITLE 'CICS ERROR HANDLER'
*
*
         SPACE
*
DFHEISTG DSECT
*
SAVE14   DS    A
RESP     DS    F
TDNAME   DS    CL4
RS       DS    CL4
RESX     DS    CL8
SWF      DS    X
SWTD     DS    X
*
STDAREA  DS    ØCL8Ø
TRANSID  DS    CL4              TRANSACTION IDENTIFIER
         DS    CL1
RESCØ    DS    CL2 P:
PGMNAME  DS    CL8              CALLING PROGRAM
         DS    CL1
TIME     DS    CL1Ø             TIME HH.MM.SS
         DS    CL1
TKN      DS    CL5              TASK NUMBER
         DS    CL1
SYID     DS    CL4,CL2          SYSTEM-ID
         DS    CL1
STCODE   DS    CL2
         DS    CL1
FNEIB    DS    CL4              FUNCTION CODE
         DS    CL1
RCODEEIB DS    CL12             ERROR CODE
         ORG   STDAREA
         DS    CL8Ø
*
STDAREA1 DS    ØCL8Ø
         DS    CL4              TRANSACTION IDENTIFIER
         DS    CL1
RESC     DS    CL2 R:
SRCE     DS    CL8              RESOURCE
         DS    CL1
         DS    CL1Ø             TIME HH.MM.SS
         DS    CL1
         DS    CL5              TASK NUMBER
         DS    CL1
         DS    CL4,CL2          SYSTEM-ID
         DS    CL1
         DS    CL2
         DS    CL1
FNC      DS    CL2Ø             FUNCTION CODE DECODIFIED
         DS    CL1
ERC      DS    CL15             ERROR CODE DECODIFIED
         DS    CL1
         ORG   STDAREA1
         DS    CL8Ø
         ORG   *-3Ø
ERMSG    DS    ØCL3Ø
FTDAREA  EQU   *
         PRINT NOGEN
DERCODE  DFHEIENT CODEREG=(4,5,6),DATAREG=(13),EIBREG=(12)
DERCODE  AMODE ANY
DERCODE  RMODE ANY
RØ       EQU   Ø
R1       EQU   1
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
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
COMPTR   EQU   R11
RWKR1    EQU   R1
RWKR2    EQU   R2
RWKR14   EQU   R14
RWKR15   EQU   R15
         B     ACXID
         CXID  DERCODE,R=ØØØ1
ACXID    DS    ØH
*
         EXEC  CICS IGNORE CONDITION ERROR
*
* CLEAR MESSAGE AREA
*
         LA    RWKR1,STDAREA
         LA    RWKR2,FTDAREA-STDAREA
LOOPBLK  DS    ØH
         MVI   Ø(RWKR1),C' '
         LA    RWKR1,1(RWKR1)
         BCT   RWKR2,LOOPBLK
         MVC   STDAREA1,STDAREA
         MVC   RESX,STDAREA
*
         MVC   TDNAME,QNAME         SET DEFAULT TD NAME
*
         USING DERCODED,COMPTR
         L     COMPTR,DFHEICAP
         CLC   EIBCALEN,=Y(DEERRØAF-DEERRØAI) VERIFY COMMAREA LEN
         BL    COMER
         MVC   ERMSGS,STDAREA       MOVE BLANK
         CLI   ERRESNAM,C' '
         BE    NORESX
         CLI   ERRESNAM,X'Ø'
         BE    NORESX
         MVC   RESX,ERRESNAM        MOVE RESOURCE NAME
NORESX   DS    ØH
         CLI   ERTDQNAM,X'FF'       MSG REQUIRED ?
         BE    NORSØ                ..NO
         CLI   ERTDQNAM,X'Ø'        DEFAULT TD QUEUE
         BE    NORS1                ...YES
         CLI   ERTDQNAM,C' '        DEFAULT TD QUEUE
         BE    NORS1                ...YES
         MVC   TDNAME,ERTDQNAM      MOVE REQUIRED QUEUE
*
         EXEC  CICS INQUIRE TDQUEUE(TDNAME) RESP(RESP)
*
         CLC   RESP,DFHRESP(NORMAL) TD QUEUE ERROR?
         BE    NORS1                ... NO
         MVC   TDNAME,QNAME         SET DEFAULT TD NAME
         B     NORS1                ... NO
NORSØ    DS    ØH
         MVI   SWTD,X'FF'           NO MSG REQUIRED
NORS1    DS    ØH
*
         EXEC  CICS ASSIGN SYSID(SYID) STARTCODE(STCODE)
*
         EXEC  CICS INQUIRE TERMINAL(EIBTRMID) REMOTESYSTEM(RS)
*
         MVC   SYID+L'SYID(3),=CL3'-L-' LOCAL TERMINAL
         CLI   RS,C' '
         BE    LRS
         CLI   RS,X'Ø'
         BE    LRS
         MVC   SYID+L'SYID(3),=CL3'-R-' REMOTE TERMINAL
         MVC   SYID,RS
LRS      DS    ØH
         MVC   TRANSID,EIBTRNID
         MVC   PGMNAME,ERPGMCAL
         MVC   TIME,=XL1Ø'FØ2Ø2Ø2Ø4B2Ø2Ø4B2Ø2Ø'
         ED    TIME,EIBTIME
         MVI   TIME,C' '
         MVI   TIME+1,C'-'
         UNPK  TKN,EIBTASKN
         OI    TKN+L'TKN-1,X'FØ'
         MVI   TKN+L'TKN,C'-'
         UNPK  FNEIB(L'FNEIB+1),ERFUNCOD(3)
         TR    FNEIB(L'FNEIB+1),TABEX-24Ø
         MVI   FNEIB+L'FNEIB,C'-'
         UNPK  RCODEEIB(L'RCODEEIB+1),ERFUNCOD+2(L'EIBRCODE+1)
         TR    RCODEEIB(L'RCODEEIB+1),TABEX-24Ø
         MVI   RCODEEIB+L'RCODEEIB,C'-'
         L     RWKR2,=A(NFN)
         LH    RWKR2,Ø(RWKR2)
         L     RWKR1,=A(TABFN)
LOOPFN   DS    ØH
         CLC   ERFUNCOD(2),Ø(RWKR1)
         BE    FFN
         LA    RWKR1,L'TABFN(RWKR1)
         BCT   RWKR2,LOOPFN
         MVC   FNC,=CL2Ø'INVALID FUNCTION'
         B     AFN
FFN      DS    ØH
         MVC   FNC,2(RWKR1)
AFN      DS    ØH
         L     RWKR2,=A(NTABEC)
         LH    RWKR2,Ø(RWKR2)
         L     RWKR1,=A(TABEC)
LOOPFN2  DS    ØH
         CLC   ERFUNCOD(1),Ø(RWKR1)
         BE    TESTERC
RLOOPFN2 DS    ØH
         MVI   SWF,X'ØØ'
         LA    RWKR1,L'TABEC(RWKR1)
         BCT   RWKR2,LOOPFN2
         MVC   ERC,=CL15'INVALID ER/CODE'
         B     WRITETD
TESTERC  DS    ØH
         LA    RWKR15,1(RWKR1)
         LA    RWKR14,4
LOOPERC  DS    ØH
         CLI   Ø(RWKR15),X'Ø'
         BNE   TESTB
RLOOPERC DS    ØH
         LA    RWKR15,1(RWKR15)
         BCT   RWKR14,LOOPERC
         CLI   SWF,X'ØØ'
         BE    RLOOPFN2
         MVC   ERC,5(RWKR1)
         B     WRITETD
```

```
TESTB    DS    ØH
         ST    RWKR14,SAVE14
         SH    RWKR14,=H'4'
         LPR   RWKR14,RWKR14
         LA    RWKR14,ERFUNCOD+2(RWKR14)
         CLC   Ø(1,RWKR14),Ø(RWKR15)
         L     RWKR14,SAVE14
         BNE   RLOOPFN2
         MVI   SWF,X'FF'
         B     RLOOPERC
COMER    DS    ØH
         MVC   ERMSG,=CL3Ø'COMMAREA LENGTH ERROR'
         B     WRITETDA
WRITETD  DS    ØH
         CLC   EIBCALEN,=Y(L'ERFUNCOD+L'ERERRCOD)
         BE    WRITETDA
         MVC   ERMSGS,FNC
WRITETDA DS    ØH
*
         CLI   SWTD,X'FF'
         BE    RETURN
         MVC   RESCØ,=CL2'P:' CALLING PROGRAM
*
         EXEC  CICS WRITEQ TD QUEUE(TDNAME) FROM(STDAREA)              *
               LENGTH(=Y(L'STDAREA))
*
         MVC   STDAREA1(FNC-STDAREA1),STDAREA
         MVC   RESC,=CL2'R:' RESOURCE
         MVC   SRCE,RESX
*
         EXEC  CICS WRITEQ TD QUEUE(TDNAME) FROM(STDAREA1)             *
               LENGTH(=Y(L'STDAREA1))
*
RETURN   DS    ØH
*
         EXEC  CICS RETURN
*
TABEX    DC    C'Ø123456789ABCDEF'
*
         LTORG *
*
         CSNAME
*
QNAME    DC    CL4'CSMT' DEFAULT TRANSIENT DATA QUEUE
*
         COPY  EIBCODE
*
DERCODED DSECT
*=====================================================================*
*=====================================================================*
```

```
DEERRØAI DS    ØH
ERFUNCOD DS    CL2  FUNCTION CODE
ERERRCOD DS    CL6  ERROR CODE
ERRESNAM DS    CL8  RESOURCE NAME
ERTDQNAM DS    CL4  TD NAME
*              X'ØØØØØØØØ' DEFAULT TD QUEUE (CSMT)
*              CL4' '      DEFAULT TD QUEUE (CSMT)
*              X'FFFFFFFF' NO SEND MSG
*              CL4'....'   TD QUEUE NAME SPECIFIED BY CALLER
ERPGMCAL DS    CL8  CALLING PROGRAM
ERMSGS   DS    CL36 ERROR MSG
*
DEERRØAF EQU   *
         ORG   DEERRØAI
DEERRØAG DS    CL(DEERRØAF-DEERRØAI)
DEERRØAL EQU   L'DEERRØAG
************************************************************ END ERR **
*
         END   DERCODE
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
*
* COPY EIBCODE
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                   *
*              EIB FUNCTION CODES & ERROR CODES                     *
*                  updated to CICS Version 3.3                      *
*                                                                   *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
NFN      DC    Y((ENDTBFN-TABFN)/L'TABFN)
TABFN    DS    ØXL22 FUNCTION CODES
         DC    XL2'Ø2Ø2',CL2Ø'ADDRESS'
         DC    XL2'Ø2Ø4',CL2Ø'HANDLE CONDITION'
         DC    XL2'Ø2Ø6',CL2Ø'HANDLE AID'
         DC    XL2'Ø2Ø8',CL2Ø'ASSIGN'
         DC    XL2'Ø2ØA',CL2Ø'IGNORE CONDITION'
         DC    XL2'Ø2ØC',CL2Ø'PUSH'
         DC    XL2'Ø2ØE',CL2Ø'POP'
         DC    XL2'Ø21Ø',CL2Ø'ADDRESS SET'
         DC    XL2'Ø4Ø2',CL2Ø'RECEIVE'
         DC    XL2'Ø4Ø4',CL2Ø'SEND'
         DC    XL2'Ø4Ø6',CL2Ø'CONVERSE'
         DC    XL2'Ø4Ø8',CL2Ø'ISSUE EODS'
         DC    XL2'Ø4ØA',CL2Ø'ISSUE COPY'
         DC    XL2'Ø4ØC',CL2Ø'WAIT TERMINAL'
         DC    XL2'Ø4ØE',CL2Ø'ISSUE LOAD'
         DC    XL2'Ø41Ø',CL2Ø'WAIT SIGNAL'
         DC    XL2'Ø412',CL2Ø'ISSUE RESET'
         DC    XL2'Ø414',CL2Ø'ISSUE DISCONNECT'
         DC    XL2'Ø416',CL2Ø'ISSUE ENDOUTPUT'
         DC    XL2'Ø418',CL2Ø'ISSUE ERASEUP'
         DC    XL2'Ø41A',CL2Ø'ISSUE ENDFILE'
         DC    XL2'Ø41C',CL2Ø'ISSUE PRINT'
         DC    XL2'Ø41E',CL2Ø'ISSUE SIGNAL'
         DC    XL2'Ø42Ø',CL2Ø'ALLOCATE'
         DC    XL2'Ø422',CL2Ø'FREE'
         DC    XL2'Ø424',CL2Ø'POINT'
         DC    XL2'Ø426',CL2Ø'BUILD ATTACH'
         DC    XL2'Ø428',CL2Ø'EXTRACT ATTACH'
         DC    XL2'Ø42A',CL2Ø'EXTRACT TCT'
         DC    XL2'Ø42C',CL2Ø'WAIT CONVID'
         DC    XL2'Ø42E',CL2Ø'EXTRACT PROCESS'
         DC    XL2'Ø43Ø',CL2Ø'ISSUE ABEND'
         DC    XL2'Ø432',CL2Ø'CONNECT PROCESS'
         DC    XL2'Ø434',CL2Ø'ISSUE CONFIRMATION'
         DC    XL2'Ø436',CL2Ø'ISSUE ERROR'
         DC    XL2'Ø438',CL2Ø'ISSUE PREPARE'
         DC    XL2'Ø43A',CL2Ø'ISSUE PASS'
         DC    XL2'Ø43C',CL2Ø'EXTRACT LOGONMSG'
         DC    XL2'Ø43E',CL2Ø'EXTRACT ATTRIBUTES'
         DC    XL2'5E32',CL2Ø'WAITCICS'
         DC    XL2'Ø6Ø2',CL2Ø'READ'
         DC    XL2'Ø6Ø4',CL2Ø'WRITE'
         DC    XL2'Ø6Ø6',CL2Ø'REWRITE'
         DC    XL2'Ø6Ø8',CL2Ø'DELETE'
         DC    XL2'Ø6ØA',CL2Ø'UNLOCK'
         DC    XL2'Ø6ØC',CL2Ø'STARTBR'
         DC    XL2'Ø6ØE',CL2Ø'READNEXT'
         DC    XL2'Ø61Ø',CL2Ø'READPREV'
         DC    XL2'Ø612',CL2Ø'ENDBR'
         DC    XL2'Ø614',CL2Ø'RESETBR'
         DC    XL2'Ø8Ø2',CL2Ø'WRITEQ TD'
         DC    XL2'Ø8Ø4',CL2Ø'READQ TD'
         DC    XL2'Ø8Ø6',CL2Ø'DELETEQ TD'
         DC    XL2'ØAØ2',CL2Ø'WRITEQ TS'
         DC    XL2'ØAØ4',CL2Ø'READQ TS'
         DC    XL2'ØAØ6',CL2Ø'DELETEQ TS'
         DC    XL2'ØCØ2',CL2Ø'GETMAIN'
         DC    XL2'ØCØ4',CL2Ø'FREEMAIN'
         DC    XL2'ØEØ2',CL2Ø'LINK'
         DC    XL2'ØEØ4',CL2Ø'XCTL'
         DC    XL2'ØEØ6',CL2Ø'LOAD'
         DC    XL2'ØEØ8',CL2Ø'RETURN'
         DC    XL2'ØEØA',CL2Ø'RELEASE'
         DC    XL2'ØEØC',CL2Ø'ABEND'
         DC    XL2'ØEØE',CL2Ø'HANDLE ABEND'
         DC    XL2'1ØØ2',CL2Ø'ASKTIME'
         DC    XL2'1ØØ4',CL2Ø'DELAY'
         DC    XL2'1ØØ6',CL2Ø'POST'
         DC    XL2'1ØØ8',CL2Ø'START'
```

```
         DC    XL2'100A',CL20'RETRIEVE'
         DC    XL2'100C',CL20'CANCEL'
         DC    XL2'1202',CL20'WAIT EVENT'
         DC    XL2'1204',CL20'ENQ'
         DC    XL2'1206',CL20'DEQ'
         DC    XL2'1208',CL20'SUSPEND'
         DC    XL2'1402',CL20'WRITE JOURNAL'
         DC    XL2'1404',CL20'WAIT JOURNAL'
         DC    XL2'1602',CL20'SYNCPOINT'
         DC    XL2'1604',CL20'RESYNC'
         DC    XL2'1802',CL20'RECEIVE MAP'
         DC    XL2'1804',CL20'SEND MAP'
         DC    XL2'1806',CL20'SEND TEXT'
         DC    XL2'1808',CL20'SEND PAGE'
         DC    XL2'180A',CL20'PURGE MESSAGE'
         DC    XL2'180C',CL20'ROUTE'
         DC    XL2'180E',CL20'RECEIVE PARTN'
         DC    XL2'1810',CL20'SEND PARTNSET'
         DC    XL2'1812',CL20'SEND CONTROL'
         DC    XL2'1A02',CL20'TRACE ON/OFF'
         DC    XL2'1A04',CL20'ENTER'
         DC    XL2'1C02',CL20'DUMP'
         DC    XL2'1E02',CL20'ISSUE ADD'
         DC    XL2'1E04',CL20'ISSUE ERASE'
         DC    XL2'1E06',CL20'ISSUE REPLACE'
         DC    XL2'1E08',CL20'ISSUE ABORT'
         DC    XL2'1E0A',CL20'ISSUE QUERY'
         DC    XL2'1E0C',CL20'ISSUE END'
         DC    XL2'1E0E',CL20'ISSUE RECEIVE'
         DC    XL2'1E10',CL20'ISSUE NOTE'
         DC    XL2'1E12',CL20'ISSUE WAIT'
         DC    XL2'1E14',CL20'ISSUE SEND'
         DC    XL2'2002',CL20'BIF DEEDIT'
         DC    XL2'2202',CL20'EXIT ENABLE'
         DC    XL2'2204',CL20'EXIT DISABLE'
         DC    XL2'2206',CL20'EXIT EXTRACT'
         DC    XL2'4802',CL20'ENTER TRACENUM'
         DC    XL2'4804',CL20'MONITOR'
         DC    XL2'4A02',CL20'ASKTIME ABSTIME'
         DC    XL2'4A04',CL20'FORMATTIME'
         DC    XL2'4C02',CL20'INQUIRE FILE'
         DC    XL2'4C04',CL20'SET FILE'
         DC    XL2'4E02',CL20'INQUIRE PROGRAM'
         DC    XL2'4E04',CL20'SET PROGRAM'
         DC    XL2'5002',CL20'INQUIRE TRANSACTION'
         DC    XL2'5004',CL20'SET TRANSACTION'
         DC    XL2'5202',CL20'INQUIRE TERMINAL'
         DC    XL2'5204',CL20'SET TERMINAL'
         DC    XL2'5206',CL20'INQUIRE NETNAME'
         DC    XL2'5402',CL20'INQUIRE SYSTEM'
         DC    XL2'54Ø4',CL2Ø'SET SYSTEM'
         DC    XL2'56Ø2',CL2Ø'SPOOLOPEN'
         DC    XL2'56Ø4',CL2Ø'SPOOLREAD'
         DC    XL2'56Ø6',CL2Ø'SPOOLWRITE'
         DC    XL2'561Ø',CL2Ø'SPOOLCLOSE'
         DC    XL2'58Ø2',CL2Ø'INQUIRE CONNECTION'
         DC    XL2'58Ø4',CL2Ø'SET CONNECTION'
         DC    XL2'5AØ2',CL2Ø'INQUIRE MODENAME'
         DC    XL2'5AØ4',CL2Ø'SET MODENAME'
         DC    XL2'5EØ6',CL2Ø'CHANGE TASK'
         DC    XL2'5E22',CL2Ø'WAIT EXTERNAL'
         DC    XL2'6614',CL2Ø'SET TRANDUMPCODE'
         DC    XL2'6AØ2',CL2Ø'QUERY SECURITY'
         DC    XL2'6CØ2',CL2Ø'WRITE OPERATOR'
         DC    XL2'6C12',CL2Ø'ISSUE DFHWTO'
         DC    XL2'74Ø2',CL2Ø'SIGN ON'
         DC    XL2'74Ø4',CL2Ø'SIGN OFF'
         DC    XL2'7EØ2',CL2Ø'DUMP TRANSACTION'
         DC    XL2'7EØ4',CL2Ø'DUMP SYSTEM'
         DC    XL2'82ØE',CL2Ø'AP NOOP'
         DC    XL2'821Ø',CL2Ø'ALLOCATE'
         DC    XL2'8212',CL2Ø'CONVERSE FORMATTED'
         DC    XL2'8214',CL2Ø'CONVERSE DATASTREAM'
         DC    XL2'8216',CL2Ø'EXTRACT CONV'
         DC    XL2'8218',CL2Ø'EXTRACT FIELDS'
         DC    XL2'821A',CL2Ø'EXTRACT STSN'
         DC    XL2'821C',CL2Ø'FREE'
         DC    XL2'821E',CL2Ø'ISSUE'
         DC    XL2'822Ø',CL2Ø'RECEIVE FORMATTED'
         DC    XL2'8222',CL2Ø'RECEIVE DATASTREAM'
         DC    XL2'8224',CL2Ø'SEND FORMATTED'
         DC    XL2'8226',CL2Ø'SEND DATASTREAM'
         DC    XL2'8228',CL2Ø'START'
         DC    XL2'84Ø2',CL2Ø'CICS NORMAL SHUTDOWN'
         DC    XL2'84Ø4',CL2Ø'CICS IMMED. SHUTDOWN'
         DC    XL2'84Ø6',CL2Ø'CICS FORCED SHUTDOWN'
         DC    XL2'84Ø8',CL2Ø'CICS END-OF-TASK'
         DC    XL2'84ØE',CL2Ø'SP NOOP'
         DC    XL2'8422',CL2Ø'INQUIRE PROPERTYSET'
         DC    XL2'8428',CL2Ø'INSTALL PROPERTYSET'
         DC    XL2'843Ø',CL2Ø'DISCARD PROPERTYSET'
         DC    XL2'8442',CL2Ø'INQUIRE NODE'
         DC    XL2'8444',CL2Ø'SET NODE'
         DC    XL2'8448',CL2Ø'INSTALL NODELIST'
         DC    XL2'844A',CL2Ø'ADD POOL'
         DC    XL2'844C',CL2Ø'DELETE POOL'
         DC    XL2'845Ø',CL2Ø'DISCARD NODELIST'
         DC    XL2'8462',CL2Ø'INQUIRE POOL'
         DC    XL2'8464',CL2Ø'SET POOL'
         DC    XL2'8468',CL2Ø'INSTALL POOL'
         DC    XL2'847Ø',CL2Ø'DISCARD POOL'
         DC    XL2'8482',CL2Ø'INQUIRE TARGET'
         DC    XL2'8484',CL2Ø'SET TARGET'
         DC    XL2'8488',CL2Ø'INSTALL TARGETLIST'
         DC    XL2'849Ø',CL2Ø'DISCARD TARGETLIST'
         DC    XL2'84A2',CL2Ø'INQUIRE CONNECTION'
         DC    XL2'84A4',CL2Ø'SET CONNECTION'
ENDTBFN  EQU   *
NTABEC   DC    Y((ENDTABEC-TABEC)/L'TABEC)
* EIBFN(BYTE Ø) EIBRCODE(BYTES Ø-3)
TABEC    DS    ØXL2Ø
         DC    X'Ø2',XL4'EØØØØØØØ',CL15'INVREQ'
         DC    X'Ø4',XL4'Ø4ØØØØØØ',CL15'EOF'
         DC    X'Ø4',XL4'1ØØØØØØØ',CL15'EODS'
         DC    X'Ø4',XL4'C1ØØØØØØ',CL15'EOF'
         DC    X'Ø4',XL4'C2ØØØØØØ',CL15'ENDINPT'
         DC    X'Ø4',XL4'DØØØØØØØ',CL15'SYSIDERR'
         DC    X'Ø4',XL4'DØØ4ØØØØ',CL15'REQ.FUN.NOT VAL'
         DC    X'Ø4',XL4'DØØ4Ø4ØØ',CL15'NO SESSION AVLB'
         DC    X'Ø4',XL4'DØØ4Ø8ØØ',CL15'MODENAME NOT FO'
         DC    X'Ø4',XL4'DØØ4ØCØØ',CL15'MODENAME NOT VA'
         DC    X'Ø4',XL4'DØØ41ØØØ',CL15'TASK CANCELLED'
         DC    X'Ø4',XL4'DØØ414ØØ',CL15'MODE GROUP OUT'
         DC    X'Ø4',XL4'DØØ418ØØ',CL15'CLOSE DRAIN=ALL'
         DC    X'Ø4',XL4'DØØ8ØØØØ',CL15'SYSID OUT OF SE'
         DC    X'Ø4',XL4'DØØCØØØØ',CL15'NAME NOT= TCTSE'
         DC    X'Ø4',XL4'DØØCØ4ØØ',CL15'NAME NOT= REMOT'
         DC    X'Ø4',XL4'DØØCØ8ØØ',CL15'MODE NAME NOTFO'
         DC    X'Ø4',XL4'DØØCØCØØ',CL15'PROFILE NOT FOU'
         DC    X'Ø4',XL4'DØØ8ØØØØ',CL15'LINK OUT OF SRVC'
         DC    X'Ø4',XL4'DØØCØØØØ',CL15'NAME UNKNOWN'
         DC    X'Ø4',XL4'DØØCØ4ØØ',CL15'NOT NAME OF SE'
         DC    X'Ø4',XL4'DØØCØ8ØØ',CL15'MODENAME NOTFND'
         DC    X'Ø4',XL4'DØØCØCØØ',CL15'PROFILE NOTFND'
         DC    X'Ø4',XL4'D2ØØØØØØ',CL15'SESSIONERR'
         DC    X'Ø4',XL4'D2Ø4ØØØØ',CL15'NOT NAME OF S.E.'
         DC    X'Ø4',XL4'D2Ø8ØØØØ',CL15'LINK OUT OF SRVC'
         DC    X'Ø4',XL4'D2ØCØØØØ',CL15'NAME UNKNOWN'
         DC    X'Ø4',XL4'D2ØCØØØØ',CL15'PROFILE UNKNOWN'
         DC    X'Ø4',XL4'D3ØØØØØØ',CL15'SYSBUSY'
         DC    X'Ø4',XL4'D3ØØØØØ1',CL15'SYSBUSY (TOR)'
         DC    X'Ø4',XL4'D3ØØØØØ2',CL15'SYSBUSY (TOR)'
         DC    X'Ø4',XL4'D4ØØØØØØ',CL15'SESSBUSY'
         DC    X'Ø4',XL4'D5ØØØØØØ',CL15'NOTALLOC'
         DC    X'Ø4',XL4'EØØØØØØØ',CL15'INVREQ'
         DC    X'Ø4',XL4'EØØØØØØ4',CL15'TE ALRDY ALCT'
         DC    X'Ø4',XL4'EØØØØØØ8',CL15'TE WRONG STATE'
         DC    X'Ø4',XL4'EØØØØØØC',CL15'SYNCL2 NOT SUP.'
         DC    X'Ø4',XL4'EØØØØØ1Ø',CL15'INVALID DATA'
         DC    X'Ø4',XL4'EØØØØØ14',CL15'CONF.NOT SYNCL2'
         DC    X'Ø4',XL4'EØØØØØ18',CL15'INVALID NETNAME'
         DC    X'Ø4',XL4'EØØØØØ1C',CL15'CMD CONFLICT'
         DC    X'Ø4',XL4'EØØØØØ2Ø',CL15'CMD CONFLICT'
         DC    X'Ø4',XL4'EØØØØØ28',CL15'GTMN FAILURE'
         DC    X'Ø4',XL4'E1ØØØØØØ',CL15'LENGERR'
         DC    X'Ø4',XL4'E1Ø4ØØØØ',CL15'OUTPUT LENGERR'
         DC    X'Ø4',XL4'E1Ø8ØØØØ',CL15'INPUT LENGERR'
         DC    X'Ø4',XL4'E1ØCØØØØ',CL15'LENGERR'
         DC    X'Ø4',XL4'E3ØØØØØØ',CL15'WRBRK'
         DC    X'Ø4',XL4'E4ØØØØØØ',CL15'RDATT'
         DC    X'Ø4',XL4'E5ØØØØØØ',CL15'SIGNAL'
         DC    X'Ø4',XL4'E6ØØØØØØ',CL15'TERMIDERR'
         DC    X'Ø4',XL4'E7ØØØØØØ',CL15'NOPASSBKRD'
         DC    X'Ø4',XL4'E8ØØØØØØ',CL15'NOPASSBKWR'
         DC    X'Ø4',XL4'EAØØØØØØ',CL15'IGREQCD'
         DC    X'Ø4',XL4'EBØØØØØØ',CL15'CBIDERR'
         DC    X'Ø4',XL4'F1ØØØØØØ',CL15'TERMERR'
         DC    X'Ø4',XL4'ØØ2ØØØØØ',CL15'EOC'
         DC    X'Ø4',XL4'ØØ4ØØØØØ',CL15'IMBFMH'
         DC    X'Ø4',XL4'ØØØØØØF6',CL15'NOSTART'
         DC    X'Ø4',XL4'ØØØØØØF7',CL15'NONVAL'
         DC    X'Ø4',XL4'ØØ2ØØØØØ',CL15'EOC'
         DC    X'Ø4',XL4'ØØ4ØØØØØ',CL15'INBFMH'
         DC    X'Ø4',XL4'ØØØØØØF6',CL15'NOSTART'
         DC    X'Ø4',XL4'ØØØØØØF7',CL15'NONVAL'
         DC    X'Ø6',XL4'Ø1ØØØØØØ',CL15'DSIDERR'
         DC    X'Ø6',XL4'Ø2ØØØØØØ',CL15'ILLOGIC'
         DC    X'Ø6',XL4'Ø4ØØØØØØ',CL15'SEGIDERR'
         DC    X'Ø6',XL4'Ø8ØØØØØØ',CL15'INVREQ'
         DC    X'Ø6',XL4'ØCØØØØØØ',CL15'NOTOPEN'
         DC    X'Ø6',XL4'ØDØØØØØØ',CL15'DISABLED'
         DC    X'Ø6',XL4'ØFØØØØØØ',CL15'ENDFILE'
         DC    X'Ø6',XL4'8ØØØØØØØ',CL15'IOERR'
         DC    X'Ø6',XL4'81ØØØØØØ',CL15'NOTFND'
         DC    X'Ø6',XL4'82ØØØØØØ',CL15'DUPREC'
         DC    X'Ø6',XL4'83ØØØØØØ',CL15'NOSPACE'
         DC    X'Ø6',XL4'84ØØØØØØ',CL15'DUPKEY'
         DC    X'Ø6',XL4'85ØØØØØØ',CL15'SUPPRESSED'
         DC    X'Ø6',XL4'86ØØØØØØ',CL15'LOADING'
         DC    X'Ø6',XL4'DØØØØØØØ',CL15'SYSIDERR'
         DC    X'Ø6',XL4'DØØ4ØØØØ',CL15'NOT NAME OF S.E.'
         DC    X'Ø6',XL4'DØØ4Ø4ØØ',CL15'NO SESSION AVLB'
         DC    X'Ø6',XL4'DØØ4Ø8ØØ',CL15'MODENAME NOT FO'
         DC    X'Ø6',XL4'DØØ4ØCØØ',CL15'MODENAME NOT VA'
         DC    X'Ø6',XL4'DØØ41ØØØ',CL15'TASK CANCELLED'
         DC    X'Ø6',XL4'DØØ414ØØ',CL15'MODE GROUP OUT'
         DC    X'Ø6',XL4'DØØ418ØØ',CL15'CLOSE DRAIN=ALL'
```

```
         DC    X'Ø6',XL4'DØØ8ØØØØ',CL15'LINK OUT OF SRVC'
         DC    X'Ø6',XL4'DØØCØØØØ',CL15'NAME UNKNOWN'
         DC    X'Ø6',XL4'DØØCØ4ØØ',CL15'NAME NOT= REMOT'
         DC    X'Ø6',XL4'DØØCØ8ØØ',CL15'MODE NAME NOTFO'
         DC    X'Ø6',XL4'DØØCØCØØ',CL15'PROFILE NOT FOU'
         DC    X'Ø6',XL4'DØØ8ØØØØ',CL15'LINK OUT OF SRVC'
         DC    X'Ø6',XL4'DØØCØØØØ',CL15'NAME UNKNOWN'
         DC    X'Ø6',XL4'DØØCØ4ØØ',CL15'NOT NAME OF SE'
         DC    X'Ø6',XL4'DØØCØ8ØØ',CL15'MODENAME NOTFND'
         DC    X'Ø6',XL4'DØØCØCØØ',CL15'PROFILE NOTFND'
         DC    X'Ø6',XL4'D1ØØØØØØ',CL15'ISCINVREQ'
         DC    X'Ø6',XL4'D6ØØØØØØ',CL15'NOTAUTH'
         DC    X'Ø6',XL4'E1ØØØØØØ',CL15'LENGERR'
         DC    X'Ø8',XL4'Ø1ØØØØØØ',CL15'QZERO'
         DC    X'Ø8',XL4'Ø2ØØØØØØ',CL15'QIDERR'
         DC    X'Ø8',XL4'Ø4ØØØØØØ',CL15'IOERR'
         DC    X'Ø8',XL4'Ø8ØØØØØØ',CL15'NOTOPEN'
         DC    X'Ø8',XL4'1ØØØØØØØ',CL15'NOSPACE'
         DC    X'Ø8',XL4'CØØØØØØØ',CL15'QBUSY'
         DC    X'Ø8',XL4'DØØØØØØØ',CL15'SYSIDERR'
         DC    X'Ø8',XL4'DØØ4ØØØØ',CL15'NOT NAME OF S.E.'
         DC    X'Ø8',XL4'DØØ4Ø4ØØ',CL15'NO SESSION AVLB'
         DC    X'Ø8',XL4'DØØ4Ø8ØØ',CL15'MODENAME NOT FO'
         DC    X'Ø8',XL4'DØØ4ØCØØ',CL15'MODENAME NOT VA'
         DC    X'Ø8',XL4'DØØ41ØØØ',CL15'TASK CANCELLED'
         DC    X'Ø8',XL4'DØØ414ØØ',CL15'MODE GROUP OUT'
         DC    X'Ø8',XL4'DØØ418ØØ',CL15'CLOSE DRAIN=ALL'
         DC    X'Ø8',XL4'DØØ8ØØØØ',CL15'LINK OUT OF SRVC'
         DC    X'Ø8',XL4'DØØCØØØØ',CL15'NAME UNKNOWN'
         DC    X'Ø8',XL4'DØØCØ4ØØ',CL15'NAME NOT= REMOT'
         DC    X'Ø8',XL4'DØØCØ8ØØ',CL15'MODE NAME NOTFO'
         DC    X'Ø8',XL4'DØØCØCØØ',CL15'PROFILE NOT FOU'
         DC    X'Ø8',XL4'DØØ8ØØØØ',CL15'LINK OUT OF SRVC'
         DC    X'Ø8',XL4'DØØCØØØØ',CL15'NAME UNKNOWN'
         DC    X'Ø8',XL4'DØØCØ4ØØ',CL15'NOT NAME OF SE'
         DC    X'Ø8',XL4'DØØCØ8ØØ',CL15'MODENAME NOTFND'
         DC    X'Ø8',XL4'DØØCØCØØ',CL15'PROFILE NOTFND'
         DC    X'Ø8',XL4'D1ØØØØØØ',CL15'ISCINVREQ'
         DC    X'Ø8',XL4'D6ØØØØØØ',CL15'NOTAUTH'
         DC    X'Ø8',XL4'D7ØØØØØØ',CL15'DISABLED'
         DC    X'Ø8',XL4'EØØØØØØØ',CL15'INVREQ'
         DC    X'Ø8',XL4'E1ØØØØØØ',CL15'LENGERR'
         DC    X'ØA',XL4'Ø1ØØØØØØ',CL15'ITEMERR'
         DC    X'ØA',XL4'Ø2ØØØØØØ',CL15'QIDERR'
         DC    X'ØA',XL4'Ø4ØØØØØØ',CL15'IOERR'
         DC    X'ØA',XL4'Ø8ØØØØØØ',CL15'NOSPACE'
         DC    X'ØA',XL4'2ØØØØØØØ',CL15'INVREQ'
         DC    X'ØA',XL4'DØØØØØØØ',CL15'SYSIDERR'
         DC    X'ØA',XL4'DØØ4ØØØØ',CL15'NOT NAME OF S.E.'
         DC    X'ØA',XL4'DØØ4Ø4ØØ',CL15'NO SESSION AVLB'
```

```
DC    X'ØA',XL4'DØØ4Ø8ØØ',CL15'MODENAME NOT FO'
DC    X'ØA',XL4'DØØ4ØCØØ',CL15'MODENAME NOT VA'
DC    X'ØA',XL4'DØØ41ØØØ',CL15'TASK CANCELLED'
DC    X'ØA',XL4'DØØ414ØØ',CL15'MODE GROUP OUT'
DC    X'ØA',XL4'DØØ418ØØ',CL15'CLOSE DRAIN=ALL'
DC    X'ØA',XL4'DØØ8ØØØØ',CL15'LINK OUT OF SRVC'
DC    X'ØA',XL4'DØØCØØØØ',CL15'NAME UNKNOWN'
DC    X'ØA',XL4'DØØCØ4ØØ',CL15'NAME NOT= REMOT'
DC    X'ØA',XL4'DØØCØ8ØØ',CL15'MODE NAME NOTFO'
DC    X'ØA',XL4'DØØCØCØØ',CL15'PROFILE NOT FOU'
DC    X'ØA',XL4'DØØ8ØØØØ',CL15'LINK OUT OF SRVC'
DC    X'ØA',XL4'DØØCØØØØ',CL15'NAME UNKNOWN'
DC    X'ØA',XL4'DØØCØ4ØØ',CL15'NOT NAME OF SE'
DC    X'ØA',XL4'DØØCØ8ØØ',CL15'MODENAME NOTFND'
DC    X'ØA',XL4'DØØCØCØØ',CL15'PROFILE NOTFND'
DC    X'ØA',XL4'D1ØØØØØØ',CL15'ISCINVREQ'
DC    X'ØA',XL4'D6ØØØØØØ',CL15'NOTAUTH'
DC    X'ØA',XL4'E1ØØØØØØ',CL15'LENGERR'
DC    X'ØC',XL4'E1ØØØØØØ',CL15'LENGERR'
DC    X'ØC',XL4'E2ØØØØØØ',CL15'NOSTG'
DC    X'ØE',XL4'Ø1ØØØØØØ',CL15'PGMIDERR'
DC    X'ØE',XL4'D6ØØØØØØ',CL15'NOTAUTH'
DC    X'ØE',XL4'EØØØØØØØ',CL15'INVREQ'
DC    X'ØE',XL4'DØØ8ØØØØ',CL15'SYSIDERR'
DC    X'ØE',XL4'DØØ414ØØ',CL15'SYSIDERR'
DC    X'ØE',XL4'F1ØØØØØØ',CL15'TERMERR'
DC    X'1Ø',XL4'Ø1ØØØØØØ',CL15'ENDDATA'
DC    X'1Ø',XL4'Ø4ØØØØØØ',CL15'IOERR'
DC    X'1Ø',XL4'11ØØØØØØ',CL15'TRANSIDERR'
DC    X'1Ø',XL4'12ØØØØØØ',CL15'TERMIDERR'
DC    X'1Ø',XL4'14ØØØØØØ',CL15'INVTSREQ'
DC    X'1Ø',XL4'2ØØØØØØØ',CL15'EXPIRED'
DC    X'1Ø',XL4'81ØØØØØØ',CL15'NOTFND'
DC    X'1Ø',XL4'DØØØØØØØ',CL15'SYSIDERR'
DC    X'1Ø',XL4'D1ØØØØØØ',CL15'ISCINVREQ'
DC    X'1Ø',XL4'D6ØØØØØØ',CL15'NOTAUTH'
DC    X'1Ø',XL4'E1ØØØØØØ',CL15'LENGERR'
DC    X'1Ø',XL4'E9ØØØØØØ',CL15'ENVDEFERR'
DC    X'1Ø',XL4'FFØØØØØØ',CL15'INVREQ'
DC    X'12',XL4'32ØØØØØØ',CL15'ENQBUSY'
DC    X'12',XL4'EØØØØØØØ',CL15'INVREQ'
DC    X'14',XL4'Ø1ØØØØØØ',CL15'JIDERR'
DC    X'14',XL4'Ø2ØØØØØØ',CL15'INVREQ'
DC    X'14',XL4'Ø5ØØØØØØ',CL15'NOTOPEN'
DC    X'14',XL4'Ø6ØØØØØØ',CL15'LENGERR'
DC    X'14',XL4'Ø7ØØØØØØ',CL15'IOERR'
DC    X'14',XL4'Ø9ØØØØØØ',CL15'NOJBUFSP'
DC    X'14',XL4'D6ØØØØØØ',CL15'NOTAUTH'
DC    X'16',XL4'Ø1ØØØØØØ',CL15'ROLLEDBACK'
```

```
         DC    X'18',XL4'01000000',CL15'INVREQ'
         DC    X'18',XL4'02000000',CL15'RETPAGE'
         DC    X'18',XL4'04000000',CL15'MAPFAIL'
         DC    X'18',XL4'08000000',CL15'INVMPSZ'
         DC    X'18',XL4'20000000',CL15'INVERRTERM'
         DC    X'18',XL4'40000000',CL15'RTESOME'
         DC    X'18',XL4'80000000',CL15'RTEFAIL'
         DC    X'18',XL4'E1000000',CL15'LENGERR'
         DC    X'18',XL4'E3000000',CL15'WRBRK'
         DC    X'18',XL4'E4000000',CL15'RDATT'
         DC    X'18',XL4'00020000',CL15'PARTNFAIL'
         DC    X'18',XL4'00040000',CL15'INVPARTN'
         DC    X'18',XL4'00080000',CL15'INVPARTNSET'
         DC    X'18',XL4'00100000',CL15'INVLDC'
         DC    X'18',XL4'00200000',CL15'UNEXPIN'
         DC    X'18',XL4'00400000',CL15'IGREQCD'
         DC    X'18',XL4'00800000',CL15'TSIOERR'
         DC    X'18',XL4'00000100',CL15'OVERFLOW'
         DC    X'18',XL4'00000400',CL15'EODS'
         DC    X'18',XL4'00000800',CL15'EOC'
         DC    X'18',XL4'00001000',CL15'IGREQID'
         DC    X'1A',XL4'E0000000',CL15'INVREQ'
         DC    X'1A',XL4'00002000',CL15'INVREQ'
         DC    X'1E',XL4'04000000',CL15'DSSTAT'
         DC    X'1E',XL4'08000000',CL15'FUNCERR'
         DC    X'1E',XL4'0C000000',CL15'SELNERR'
         DC    X'1E',XL4'10000000',CL15'UNEXPIN'
         DC    X'1E',XL4'E1000000',CL15'LENGERR'
         DC    X'1E',XL4'00110000',CL15'EODS'
         DC    X'1E',XL4'00150000',CL15'NODATARECD'
         DC    X'1E',XL4'002B0000',CL15'IGREQCD'
         DC    X'1E',XL4'00002000',CL15'EOC'
         DC    X'22',XL4'80000000',CL15'PGMIDERR'
         DC    X'22',XL4'40000000',CL15'EXIT ID INVALID'
         DC    X'22',XL4'20000000',CL15'PGM ALR.ENABLED'
         DC    X'22',XL4'10000000',CL15'PGM ALR.ACTIVE'
         DC    X'22',XL4'08000000',CL15'PGM NOT ENABLED'
         DC    X'22',XL4'04000000',CL15'PGM NOT OWN WK'
         DC    X'22',XL4'02000000',CL15'PGM NOT ENABLED'
         DC    X'22',XL4'01000000',CL15'PGM-EXITID INV.'
         DC    X'22',XL4'00800000',CL15'PGM-BUSY'
         DC    X'22',XL4'00400000',CL15'UEINT NOT INIT.'
         DC    X'4A',XL4'00000001',CL15'ERROR'
         DC    X'4C',XL4'0000000C',CL15'DSIDERR'
         DC    X'4C',XL4'00000010',CL15'INVREQ'
         DC    X'4C',XL4'00000011',CL15'IOERR'
         DC    X'4C',XL4'00000015',CL15'ILLOGIC'
         DC    X'4C',XL4'00000046',CL15'NOTAUTH'
         DC    X'4C',XL4'00000053',CL15'END'
         DC    X'4E',XL4'00000001',CL15'ERROR'
         DC    X'4E',XL4'00000010',CL15'INVREQ'
         DC    X'4E',XL4'00000015',CL15'ILLOGIC'
         DC    X'4E',XL4'0000001B',CL15'PGMIDERR'
         DC    X'4E',XL4'00000046',CL15'NOTAUTH'
         DC    X'4E',XL4'00000053',CL15'END'
         DC    X'50',XL4'00000010',CL15'INVREQ'
         DC    X'50',XL4'00000015',CL15'ILLOGIC'
         DC    X'50',XL4'0000001C',CL15'TRANSIDERR'
         DC    X'50',XL4'00000046',CL15'NOTAUTH'
         DC    X'50',XL4'00000053',CL15'END'
         DC    X'52',XL4'00000001',CL15'ERROR'
         DC    X'52',XL4'0000000B',CL15'TERMIDERR'
         DC    X'52',XL4'00000010',CL15'INVREQ'
         DC    X'52',XL4'00000015',CL15'ILLOGIC'
         DC    X'52',XL4'00000053',CL15'END'
         DC    X'54',XL4'00000010',CL15'INVREQ'
         DC    X'56',XL4'0000000D',CL15'NOTFND'
         DC    X'56',XL4'00000010',CL15'INVREQ'
         DC    X'56',XL4'00000013',CL15'NOTOPEN'
         DC    X'56',XL4'00000014',CL15'ENDFILE'
         DC    X'56',XL4'00000015',CL15'ILLOGIC'
         DC    X'56',XL4'00000016',CL15'LENGERR'
         DC    X'56',XL4'0000002A',CL15'NOSTG'
         DC    X'56',XL4'00000046',CL15'NOTAUTH'
         DC    X'56',XL4'00000050',CL15'NOSPOOL'
         DC    X'56',XL4'00000055',CL15'ALLOCERR'
         DC    X'56',XL4'00000056',CL15'STRELERR'
         DC    X'56',XL4'00000057',CL15'OPENERR'
         DC    X'56',XL4'00000058',CL15'SPOLBUSY'
         DC    X'56',XL4'00000059',CL15'SPOLERR'
         DC    X'56',XL4'0000005A',CL15'NODEIDERR'
         DC    X'58',XL4'00000010',CL15'INVREQ'
         DC    X'58',XL4'00000015',CL15'ILLOGIC'
         DC    X'58',XL4'00000035',CL15'SYSIDERR'
         DC    X'58',XL4'00000053',CL15'END'
         DC    X'5A',XL4'00000010',CL15'INVREQ'
         DC    X'5A',XL4'00000015',CL15'ILLOGIC'
         DC    X'5A',XL4'00000035',CL15'SYSIDERR'
         DC    X'5A',XL4'00000053',CL15'END'
         DC    X'66',XL4'0000000E',CL15'DUPREC'
         DC    X'66',XL4'00000010',CL15'INVREQ'
ENDTABEC EQU   *
```

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

Giuseppe Rallo
Senior Technical Analyst
Sicilcassa (Italy)

Exploiting EXCI to manage CICS files from batch

SUMMARY

Before the introduction of the External CICS Interface (EXCI), managing CICS files from batch often meant acquiring and maintaining third-party software. Now, by using the code presented here, or by writing your own, you can administer the open and enabled status of CICS files, along with their read and update attributes, from a batch job or jobstep.

DETAILS

EXCI was introduced in CICS/ESA 4.1 as a means for a non-CICS client program running in MVS to invoke a server program running in CICS and to pass and receive data through a communications area. One implication of this development is that the powers of Distributed Program Link (DPL), formerly available only on CICS platforms, are now at our disposal in MVS batch.

Two programs are included with this article:

- EXCIFILB is the batch client program that reads an input file of requests to administer CICS files and links to the server program.
- EXCIFILC is the CICS server program that carries out the file requests.

A communications area is used for the client to pass requests to the server and for the server to pass return codes back to the client. EXCIFILB also produces a report on the status of each file request issued. Both programs are written in COBOL/2.

Following the program source code is the required PROC and two samples of execution JCL to run the batch client program. The first sample illustrates how to close eight files allocated to PRODCICS, run a batch procedure against those files, and re-open the files to PRODCICS. The second sample shows how to place three files allocated to PRODCICS in read-only mode.

For information on establishing EXCI connections and compiling EXCI programs, please consult IBM publication DFHLTF08, *CICS/ESA Version 4 Release 1 External CICS Interface*.

EXCI CLIENT PROGRAM

```
 CBL XOPTS(EXCI,COBOL2)
    IDENTIFICATION DIVISION.
    PROGRAM-ID.  EXCIFILB.

    ENVIRONMENT DIVISION.

    INPUT-OUTPUT SECTION.

    FILE-CONTROL.
      SELECT PRINTER ASSIGN TO SYSPRINT.
      SELECT REQUEST ASSIGN TO SYSIN.

    DATA DIVISION.

    FILE SECTION.

    FD PRINTER BLOCK CONTAINS 128 CHARACTERS
      RECORDING MODE S
      LABEL RECORDS OMITTED.

    Ø1 OUTPUT-RECORD   PIC X(128).

    FD REQUEST BLOCK CONTAINS 8Ø CHARACTERS
      RECORDING MODE F
      LABEL RECORDS OMITTED.

    Ø1 INPUT-RECORD    PIC X(8Ø).

    WORKING-STORAGE SECTION.

    COPY DFHXCPLO.

    Ø1 OUTPUT-RETAREA.
      Ø5 FILLER           PIC X(2Ø)  VALUE SPACE.
      Ø5 O-RESP           PIC 9(8).
      Ø5 FILLER           PIC XX     VALUE SPACE.
      Ø5 O-RESP2          PIC 9(8).
      Ø5 FILLER           PIC XX     VALUE SPACE.
      Ø5 OEXCI-SUB-REASON1    PIC 9(8).
      Ø5 O-ABCODE-LINE    REDEFINES OEXCI-SUB-REASON1.
        1Ø O-ABCODE       PIC X(4).
        1Ø OPAD-ABCODE    PIC X(4).
```

```
  Ø5 FILLER           PIC X(8Ø)   VALUE SPACE.

Ø1 SUB                PIC S9(8)   COMP.

Ø1 OUT-REC.
  Ø5 OUT-REC-ELEM     PIC X OCCURS 128.

Ø1 TARGET-PROGRAM     PIC X(8)    VALUE 'EXCIFILC'.

Ø1 TARGET-TRANSID     PIC X(4)    VALUE 'EXCI'.

Ø1 TARGET-SYSTEM.
  Ø5 TARGET-SYS-ELEM  PIC X OCCURS 8.

Ø1 COMMAREA.
  Ø5 CA-RC            PIC S9(8)   COMP VALUE ZERO.
    88 CA-RC-GOLDEN            VALUE ZERO.
    88 CA-RC-NOFILE            VALUE +8.
    88 CA-RC-NOTSET            VALUE +12.
    88 CA-RC-FAILED            VALUE +16.
  Ø5 CA-FILE          PIC X(Ø8)   VALUE SPACE.
  Ø5 CA-OPE           PIC X(Ø3)   VALUE SPACE.
  Ø5 CA-ENA           PIC X(Ø3)   VALUE SPACE.
  Ø5 CA-FLAGS         PIC X(Ø5)   VALUE SPACE.
  Ø5 CA-FILL          PIC X(Ø1)   VALUE SPACE.

Ø1 INPUT-REQUEST.
  Ø5 IR-FILE          PIC X(Ø8)   VALUE SPACE.
  Ø5 IR-OPE           PIC X(Ø3)   VALUE SPACE.
  Ø5 IR-ENA           PIC X(Ø3)   VALUE SPACE.
  Ø5 IR-FLAGS         PIC X(Ø5)   VALUE SPACE.
  Ø5 FILLER           PIC X(61)   VALUE SPACE.

Ø1 OUTPUT-STATUS.
  Ø5 OS-REGION        PIC X(Ø9)   VALUE SPACE.
  Ø5 F                PIC X(13)   VALUE 'FILE REQUEST'.
  Ø5 OS-REQNO         PIC ZZ9     VALUE ZERO.
  Ø5 F                PIC X(Ø2)   VALUE ' ('.
  Ø5 OS-FILE          PIC X(Ø9)   VALUE SPACE.
  Ø5 OS-OPE           PIC X(Ø4)   VALUE SPACE.
  Ø5 OS-ENA           PIC X(Ø4)   VALUE SPACE.
  Ø5 OS-FLAGS         PIC X(Ø5)   VALUE SPACE.
  Ø5 F                PIC X(Ø2)   VALUE ')'.
  Ø5 OS-STATUS        PIC X(16)   VALUE SPACE.
  Ø5 F                PIC X(Ø6)   VALUE ' RC ='.
  Ø5 OS-RC            PIC 99      VALUE ZERO.
  Ø5 F                PIC X(53)   VALUE SPACE.

Ø1 STATUS-LITERALS.
  Ø5 SL-SUCCESSFUL    PIC X(16)   VALUE 'WAS SUCCESSFUL,'.
```

```
   Ø5 SL-NOT-FOUND     PIC X(16)   VALUE 'FILE NOT FOUND,'.
   Ø5 SL-INVALID       PIC X(16)   VALUE 'INVALID REQUEST,'.
   Ø5 SL-SERIOUS       PIC X(16)   VALUE 'SERIOUS ERROR,'.

 Ø1 MISC.
   Ø5 READ-CNT         PIC S9(4)   COMP VALUE ZERO.
   Ø5 REQ-EOF-SW       PIC X(Ø1)   VALUE 'N'.
     88 REQ-EOF                    VALUE 'Y'.
   Ø5 FATAL-ERR-SW     PIC X(Ø1)   VALUE 'N'.
     88 FATAL-ERR                  VALUE 'Y'.
   Ø5 RC-HIGHEST       PIC S9(8)   COMP VALUE ZERO.

 Ø1 COMM-LENGTH        PIC S9(8)   COMP VALUE 98.
 Ø1 DATA-LENGTH        PIC S9(8)   COMP VALUE 18.
 Ø1 LINK-COM-LEN       PIC S9(4)   COMP VALUE 24.
 Ø1 LINK-DAT-LEN       PIC S9(4)   COMP VALUE 24.

 Ø1 PROGRAM-MESSAGES.
  Ø5 MSGØ1 PIC X(128) VALUE '*
-   '                       *'.
  Ø5 MSGØ2 PIC X(128) VALUE '*  The Link Request has failed.
-   ' Return codes are:              *'.
  Ø5 MSGØ3 PIC X(128) VALUE '*  A message was received from t
-   'he target CICS system:            *'.
  Ø5 MSGØ4 PIC X(128) VALUE '*  >>>> Aborting further process
-   'ing <<<<                   *'.

 LINKAGE SECTION.

 Ø1 NULL-PTR           USAGE POINTER.

 Ø1 CALL-LEVEL-MSG.
   Ø5 CALL-LEVEL-MSG-LEN    PIC S9(4) COMP.
   Ø5 FILLER                PIC S9(4) COMP.
   Ø5 CALL-LEVEL-MSG-TEXT   PIC X OCCURS 128.

 Ø1 EXEC-LEVEL-MSG.
   Ø5 EXEC-LEVEL-MSG-TEXT   PIC X OCCURS 128.

 Ø1 PARM-DATA.
   Ø5 PARM-STRING-LENGTH    PIC 9(4) COMP.
   Ø5 PARM-STRING           PIC X OCCURS 8.

 PROCEDURE DIVISION USING PARM-DATA.

 ØØØ-MAINLINE.

   IF PARM-STRING-LENGTH > Ø
     MOVE SPACES TO TARGET-SYSTEM
     PERFORM TEST BEFORE
```

```
          VARYING SUB FROM 1 BY 1
          UNTIL SUB > PARM-STRING-LENGTH OR SUB > 8
            MOVE PARM-STRING (SUB) TO TARGET-SYS-ELEM(SUB)
        END-PERFORM
      ELSE
        MOVE 'DBDCCICS' TO TARGET-SYSTEM
      END-IF.

      OPEN OUTPUT PRINTER
        INPUT REQUEST.

      PERFORM 1ØØ-DRIVER THRU 1ØØ-EXIT
        UNTIL REQ-EOF OR FATAL-ERR.

      CLOSE PRINTER
         REQUEST.

      MOVE RC-HIGHEST TO RETURN-CODE.

      STOP RUN.

    1ØØ-DRIVER.

      PERFORM 11Ø-READ-REQUEST THRU 11Ø-EXIT.

      IF REQ-EOF
        GO TO 1ØØ-EXIT
      ELSE
        MOVE IR-FILE TO CA-FILE
        MOVE IR-OPE  TO CA-OPE
        MOVE IR-ENA  TO CA-ENA
        MOVE IR-FLAGS TO CA-FLAGS
        PERFORM 12Ø-LINK-CICS-PGM THRU 12Ø-EXIT
        PERFORM 13Ø-REPORT-STATUS THRU 13Ø-EXIT
      END-IF.

    1ØØ-EXIT.
      EXIT.

    11Ø-READ-REQUEST.

      INITIALIZE INPUT-REQUEST.

      READ REQUEST
        INTO INPUT-REQUEST
      AT END
        SET REQ-EOF TO TRUE.

      IF REQ-EOF
        GO TO 11Ø-EXIT
```

```
  ELSE
    ADD +1 TO READ-CNT
  END-IF.

110-EXIT.
  EXIT.

120-LINK-CICS-PGM.

  EXEC CICS LINK
    PROGRAM(TARGET-PROGRAM)
    TRANSID(TARGET-TRANSID)
    APPLID(TARGET-SYSTEM)
    COMMAREA(COMMAREA)
    LENGTH(LINK-COM-LEN)
    DATALENGTH(LINK-DAT-LEN)
     RETCODE(EXCI-EXEC-RETURN-CODE)
           SYNCONRETURN
       END-EXEC.

   120-EXIT.
       EXIT.

   130-REPORT-STATUS.

       IF EXEC-RESP = ZERO
           MOVE TARGET-SYSTEM TO OS-REGION
           MOVE READ-CNT      TO OS-REQNO
           MOVE CA-FILE       TO OS-FILE
           MOVE CA-OPE        TO OS-OPE
           MOVE CA-ENA        TO OS-ENA
           MOVE CA-FLAGS      TO OS-FLAGS
           MOVE CA-RC         TO OS-RC
           EVALUATE TRUE
               WHEN CA-RC-GOLDEN
                   MOVE SL-SUCCESSFUL TO OS-STATUS
               WHEN CA-RC-NOFILE
                   MOVE SL-NOT-FOUND  TO OS-STATUS
               WHEN CA-RC-NOTSET
                   MOVE SL-INVALID    TO OS-STATUS
               WHEN CA-RC-FAILED
                   MOVE SL-SERIOUS    TO OS-STATUS
                   SET FATAL-ERR TO TRUE
           END-EVALUATE
           IF CA-RC > RC-HIGHEST
               MOVE CA-RC TO RC-HIGHEST
           END-IF
           WRITE OUTPUT-RECORD FROM OUTPUT-STATUS
       ELSE
           SET FATAL-ERR TO TRUE
```

```
               WRITE OUTPUT-RECORD FROM MSGØ2
               MOVE  EXEC-RESP TO O-RESP
               MOVE  EXEC-RESP2 TO O-RESP2
               MOVE  SPACES TO OPAD-ABCODE
               MOVE  SPACES TO OPAD-ABCODE
               MOVE  EXEC-ABCODE TO O-ABCODE
               WRITE OUTPUT-RECORD FROM OUTPUT-RETAREA
               IF EXEC-MSG-PTR = NULLS THEN
                   MOVE +2Ø TO RC-HIGHEST
               ELSE
                   MOVE +24 TO RC-HIGHEST
                   WRITE OUTPUT-RECORD FROM MSGØ3
                   WRITE OUTPUT-RECORD FROM MSGØ1
                   SET ADDRESS OF EXEC-LEVEL-MSG TO EXEC-MSG-PTR
                   MOVE SPACES TO OUT-REC
                   PERFORM TEST BEFORE
                   VARYING SUB FROM 1 BY 1
                   UNTIL SUB > EXEC-MSG-LEN
                   MOVE EXEC-LEVEL-MSG-TEXT (SUB) TO OUT-REC-ELEM (SUB)
                   END-PERFORM
                   WRITE OUTPUT-RECORD FROM OUT-REC
                   WRITE OUTPUT-RECORD FROM MSGØ1
               END-IF
               WRITE OUTPUT-RECORD FROM MSGØ4
           END-IF.

       13Ø-EXIT.
           EXIT.
```

EXCI SERVER PROGRAM

```
 CBL XOPTS(SP)
       IDENTIFICATION DIVISION.
       PROGRAM-ID.    EXCIFILC.

       ENVIRONMENT DIVISION.

       CONFIGURATION SECTION.
       SOURCE-COMPUTER. IBM-3Ø9Ø.
       OBJECT-COMPUTER. IBM-3Ø9Ø.

       DATA DIVISION.

       WORKING-STORAGE SECTION.

       Ø1  COMMAREA.
           Ø5  CA-RC                   PIC S9(8) COMP  VALUE ZERO.
               88  CA-RC-GOLDEN                        VALUE ZERO.
               88  CA-RC-NOFILE                        VALUE +8.
```

```
          88  CA-RC-NOTSET                          VALUE +12.
          88  CA-RC-FAILED                          VALUE +16.
      Ø5  CA-FILE               PIC X(Ø8)       VALUE SPACE.
      Ø5  CA-OPE-STATUS         PIC X(Ø3)       VALUE SPACE.
      Ø5  CA-ENA-STATUS         PIC X(Ø3)       VALUE SPACE.
      Ø5  CA-FLAGS.
          1Ø  CA-BRO-STATUS     PIC X(Ø1)       VALUE SPACE.
          1Ø  CA-REA-STATUS     PIC X(Ø1)       VALUE SPACE.
          1Ø  CA-ADD-STATUS     PIC X(Ø1)       VALUE SPACE.
          1Ø  CA-DEL-STATUS     PIC X(Ø1)       VALUE SPACE.
          1Ø  CA-UPD-STATUS     PIC X(Ø1)       VALUE SPACE.
      Ø5  F                     PIC X(Ø1)       VALUE SPACE.

  Ø1  MISC.
      Ø5  CMD-RESP              PIC S9(8)       COMP.

  Ø1  CVDA-VALUES.
      Ø5  CVDA-INQ-OPE          PIC S9(8)       COMP.
      Ø5  CVDA-INQ-ENA          PIC S9(8)       COMP.
      Ø5  CVDA-INQ-VALUES.
          1Ø  CVDA-INQ-BRO      PIC S9(8)       COMP.
          1Ø  CVDA-INQ-REA      PIC S9(8)       COMP.
          1Ø  CVDA-INQ-ADD      PIC S9(8)       COMP.
          1Ø  CVDA-INQ-DEL      PIC S9(8)       COMP.
          1Ø  CVDA-INQ-UPD      PIC S9(8)       COMP.
      Ø5  CVDA-SET-OPE          PIC S9(8)       COMP.
      Ø5  CVDA-SET-ENA          PIC S9(8)       COMP.
      Ø5  CVDA-SET-VALUES.
          1Ø  CVDA-SET-BRO      PIC S9(8)       COMP.
          1Ø  CVDA-SET-REA      PIC S9(8)       COMP.
          1Ø  CVDA-SET-ADD      PIC S9(8)       COMP.
          1Ø  CVDA-SET-DEL      PIC S9(8)       COMP.
          1Ø  CVDA-SET-UPD      PIC S9(8)       COMP.
      Ø5  CVDA-CLO-OPE          PIC S9(8)       COMP.
      Ø5  CVDA-DIS-ENA          PIC S9(8)       COMP.

  LINKAGE SECTION.

  Ø1  DFHCOMMAREA           PIC X(24).

  PROCEDURE DIVISION.

  ØØØ-MAINLINE.

      EXEC CICS HANDLE CONDITION ERROR(9ØØ-ERRORS) END-EXEC.

      IF EIBCALEN = +24
          MOVE DFHCOMMAREA TO COMMAREA
      ELSE
          EXEC CICS ABEND
```

```
            ABCODE('NOCA')
        END-EXEC
    END-IF.

    PERFORM 1ØØ-INQ-FILE THRU 1ØØ-EXIT.

    EVALUATE CMD-RESP
        WHEN DFHRESP(NORMAL)
            CONTINUE
        WHEN DFHRESP(FILENOTFOUND)
            SET CA-RC-NOFILE TO TRUE
            GO TO ØØØ-EXIT
        WHEN OTHER
            SET CA-RC-FAILED TO TRUE
            GO TO ØØØ-EXIT
    END-EVALUATE.

    PERFORM 2ØØ-SET-REQ-CVDAS THRU 2ØØ-EXIT.

    IF CA-RC-GOLDEN
        CONTINUE
    ELSE
        GO TO ØØØ-EXIT
    END-IF.

    PERFORM 3ØØ-SET-FILE-INITIAL THRU 3ØØ-EXIT.

    EVALUATE CMD-RESP
        WHEN DFHRESP(NORMAL)
            CONTINUE
        WHEN DFHRESP(INVREQ)
            SET CA-RC-NOTSET TO TRUE
            GO TO ØØØ-EXIT
        WHEN OTHER
            SET CA-RC-FAILED TO TRUE
            GO TO ØØØ-EXIT
    END-EVALUATE.

    IF CVDA-SET-OPE = CVDA-CLO-OPE
            AND CVDA-SET-ENA = CVDA-DIS-ENA
        GO TO ØØØ-EXIT
    END-IF.

    PERFORM 4ØØ-SET-FILE-FINAL THRU 4ØØ-EXIT.

    EVALUATE CMD-RESP
        WHEN DFHRESP(NORMAL)
            SET CA-RC-GOLDEN TO TRUE
        WHEN DFHRESP(INVREQ)
            SET CA-RC-NOTSET TO TRUE
        WHEN OTHER
            SET CA-RC-FAILED TO TRUE
    END-EVALUATE.

ØØØ-EXIT.
    MOVE COMMAREA TO DFHCOMMAREA.
    EXEC CICS RETURN END-EXEC.
    GOBACK.

1ØØ-INQ-FILE.

    EXEC CICS INQUIRE
        FILE(CA-FILE)
        OPENSTATUS(CVDA-INQ-OPE)
        ENABLESTATUS(CVDA-INQ-ENA)
        BROWSE(CVDA-INQ-BRO)
        READ(CVDA-INQ-REA)
        ADD(CVDA-INQ-ADD)
        DELETE(CVDA-INQ-DEL)
        UPDATE(CVDA-INQ-UPD)
        RESP(CMD-RESP)
    END-EXEC.

1ØØ-EXIT.
    EXIT.

2ØØ-SET-REQ-CVDAS.

    EVALUATE CA-OPE-STATUS
        WHEN 'OPE'
            MOVE DFHVALUE(OPEN)           TO CVDA-SET-OPE
        WHEN 'CLO'
            MOVE DFHVALUE(CLOSED)         TO CVDA-SET-OPE
        WHEN SPACE
            MOVE CVDA-INQ-OPE             TO CVDA-SET-OPE
        WHEN OTHER
            SET CA-RC-NOTSET TO TRUE
            GO TO 2ØØ-EXIT
    END-EVALUATE.

    EVALUATE CA-ENA-STATUS
        WHEN 'ENA'
            MOVE DFHVALUE(ENABLED)        TO CVDA-SET-ENA
        WHEN 'DIS'
            MOVE DFHVALUE(DISABLED)       TO CVDA-SET-ENA
        WHEN 'UNE'
            MOVE DFHVALUE(UNENABLED)      TO CVDA-SET-ENA
        WHEN SPACE
            MOVE CVDA-INQ-ENA             TO CVDA-SET-ENA
        WHEN OTHER
            SET CA-RC-NOTSET TO TRUE
            GO TO 2ØØ-EXIT
    END-EVALUATE.

    EVALUATE CA-BRO-STATUS
        WHEN 'Y'
            MOVE DFHVALUE(BROWSABLE)     TO CVDA-SET-BRO
        WHEN 'N'
            MOVE DFHVALUE(NOTBROWSABLE)  TO CVDA-SET-BRO
        WHEN SPACE
            MOVE CVDA-INQ-BRO            TO CVDA-SET-BRO
        WHEN OTHER
            SET CA-RC-NOTSET TO TRUE
            GO TO 2ØØ-EXIT
    END-EVALUATE.

    EVALUATE CA-REA-STATUS
        WHEN 'Y'
            MOVE DFHVALUE(READABLE)      TO CVDA-SET-REA
        WHEN 'N'
            MOVE DFHVALUE(NOTREADABLE)   TO CVDA-SET-REA
        WHEN SPACE
            MOVE CVDA-INQ-REA            TO CVDA-SET-REA
        WHEN OTHER
            SET CA-RC-NOTSET TO TRUE
            GO TO 2ØØ-EXIT
    END-EVALUATE.

    EVALUATE CA-ADD-STATUS
        WHEN 'Y'
            MOVE DFHVALUE(ADDABLE)       TO CVDA-SET-ADD
        WHEN 'N'
            MOVE DFHVALUE(NOTADDABLE)    TO CVDA-SET-ADD
        WHEN SPACE
            MOVE CVDA-INQ-ADD            TO CVDA-SET-ADD
        WHEN OTHER
            SET CA-RC-NOTSET TO TRUE
            GO TO 2ØØ-EXIT
    END-EVALUATE.

    EVALUATE CA-DEL-STATUS
        WHEN 'Y'
            MOVE DFHVALUE(DELETABLE)     TO CVDA-SET-DEL
        WHEN 'N'
            MOVE DFHVALUE(NOTDELETABLE)  TO CVDA-SET-DEL
        WHEN SPACE
            MOVE CVDA-INQ-DEL            TO CVDA-SET-DEL
        WHEN OTHER
            SET CA-RC-NOTSET TO TRUE
            GO TO 2ØØ-EXIT
    END-EVALUATE.

    EVALUATE CA-UPD-STATUS
        WHEN 'Y'
            MOVE DFHVALUE(UPDATABLE)     TO CVDA-SET-UPD
        WHEN 'N'
            MOVE DFHVALUE(NOTUPDATABLE)  TO CVDA-SET-UPD
        WHEN SPACE
            MOVE CVDA-INQ-UPD            TO CVDA-SET-UPD
        WHEN OTHER
            SET CA-RC-NOTSET TO TRUE
            GO TO 2ØØ-EXIT
    END-EVALUATE.

2ØØ-EXIT.
    EXIT.

3ØØ-SET-FILE-INITIAL.

    MOVE DFHVALUE(CLOSED)   TO CVDA-CLO-OPE.
    MOVE DFHVALUE(DISABLED) TO CVDA-DIS-ENA.

    EXEC CICS SET
        FILE(CA-FILE)
        OPENSTATUS(CVDA-CLO-OPE)
        ENABLESTATUS(CVDA-DIS-ENA)
        BROWSE(CVDA-SET-BRO)
        READ(CVDA-SET-REA)
        ADD(CVDA-SET-ADD)
        DELETE(CVDA-SET-DEL)
        UPDATE(CVDA-SET-UPD)
        RESP(CMD-RESP)
    END-EXEC.

3ØØ-EXIT.
    EXIT.

4ØØ-SET-FILE-FINAL.

    EXEC CICS SET
        FILE(CA-FILE)
        OPENSTATUS(CVDA-SET-OPE)
        ENABLESTATUS(CVDA-SET-ENA)
        RESP(CMD-RESP)
    END-EXEC.

4ØØ-EXIT.
    EXIT.

9ØØ-ERRORS.
```

```
           SET CA-RC-FAILED TO TRUE.
           MOVE COMMAREA TO DFHCOMMAREA.
           EXEC CICS RETURN END-EXEC.

       9ØØ-EXIT.
           EXIT.
```

JCL – PROC

```
//*
//* PROC TO MANAGE CICS FILES THROUGH EXCI
//*
//* SYSIN RECORD:
//* Cols    Description      Values (b = space)
//* ——      ——————           —————————
//* Ø1-Ø8   FILE DD NAME     as defined to CICS
//* Ø9-11   OPEN STATUS      'OPE' 'CLO' 'bbb'
//* 12-14   ENABLED STATUS   'ENA' 'DIS' 'UNE'  'bbb'
//* 15      BROWSE STATUS    'Y' 'N' 'b'
//* 16      READ STATUS      'Y' 'N' 'b'
//* 17      ADD STATUS       'Y' 'N' 'b'
//* 18      DELETE STATUS    'Y' 'N' 'b'
//* 19      UPDATE STATUS    'Y' 'N' 'b'
//* 2Ø-8Ø   FILL
//*
//* Note: blank in status retains current state
//*
//EXCIFILE EXEC PGM=EXCIFILB,PARM='CICSREGN'
//STEPLIB   DD  DSN=CICSESA.SDFHEXCI,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  DUMMY
```

JCL – SAMPLE EXECUTION 1

```
//EXCICLO  EXEC EXCIFILE,PARM='PRODCICS'
//SYSIN     DD  *
ANYFILE1CLODIS
ANYFILE2CLODIS
ANYFILE3CLODIS
ANYFILE4CLODIS
ANYFILE5CLODIS
ANYFILE6CLODIS
ANYFILE7CLODIS
ANYFILE8CLODIS
/*
//*
//PROCESS  EXEC BATCHJOB,  batch process with exclusive file control
```

```
//             COND=(Ø,NE)
//*
//EXCIOPE  EXEC EXCIFILE,PARM='PRODCICS'
//SYSIN    DD *
ANYFILE1OPEENA
ANYFILE2OPEENA
ANYFILE3OPEENA
ANYFILE4OPEENA
ANYFILE5OPEENA
ANYFILE6OPEENA
ANYFILE7OPEENA
ANYFILE8OPEENA
/*
```

JCL – SAMPLE EXECUTION 2

```
//EXCIRDO  EXEC EXCIFILE,PARM='PRODCICS'
//SYSIN     DD  *
ANYFILE1OPEENAYYNNN
ANYFILE2OPEENAYYNNN
ANYFILE3OPEENAYYNNN
```

Russell Hunt
Senior Systems Programmer (USA)

Using the CICS 4.1 CREATE command

THE PROBLEM

In our organization, we do not use either auto-mailing products or an intranet. In the past, whenever CICS application programmers wanted to define a new resource for CICS, they filled in a form and sent it to the CICS system programmer.

Occasionally, the application programmer's handwriting was difficult to read, and sometimes the definitions in CICS appeared to be wrong. In addition, the CICS system programmer was often unavailaible and the application programmers had to wait.

We looked for a way to solve this problem, without allowing the application programmers full access to the CICS RDO with the CEDA transaction. (They are allowed to use CEDB/CEDC to allow them to see the resources, but not to update them.)

THE SOLUTION

We have developed a small COBOL program that reads CICS extra partition transient data that is a member in a PDS. This PDS is in the ‘world writeable’ library.

Whenever an application programmer wants to use a new resource in CICS he simply appends lines to this member, each line for a single definition in the CICS.

We use the following format:

- The first letter is either ‘T’ for transaction, ‘P’ for program, ‘M’ for mapset, or ‘*’ for remark (we decided to support only mapsets, transactions, and programs because we use auto-install for terminals and VSAM file definitions are rarely required).
- The following column is blank.
- The following four columns are:
  - The transaction name (for transaction).
  - The program language (for a program, which can be C for C, COB for COBOL, or ASS for Assembler).
  - ‘xxxx’ for a map.
- The following column is blank.
- The following eight columns are the program that the transaction starts, the mapset name, or the program name.
- The following column is blank,

We also keep one column for transaction security indication, a blank, and another two columns for the TWA. We never use more then twenty bytes. The programmer may start a line with a ‘*’ to write his or her own remark or to write an example.

The structure of the member is simple and intuitive. We write a simple

CICS COBOL program that is started every hour. The program closes and re-opens the external TD and so reads all the resources in the member. This has some performance cost, but has the advantage that programmers may self-correct errors by re-editing the line.

Because the CICS created resources are erased during a CICS cold start, we have added three steps to the CICS job which check whether it is CICS cold start (by analysing the sysin dataset) and build RDO definitions from the user member. This is done before CICS is started. The resources are defined in a temporary RDO group from where they can be moved by the CICS system programmer to the final destination.

TECHNICAL REMARKS

You should note the following:

- In our organization, we use job and not started task, and the start=cold/auto parameter is always in the sysin. In a way it is better to analyse the 'real' way in which CICS will come up, but I believe that in most installations auto/cold indication is in the 'JCL override'.

  Since the 'member definitions' should be clean, we have written some simple JCL to nullify it.

- If each CICS region has its own VSAM CSD file, then the two IEBGENER nullifying steps may be added after the cold start steps, and before CICS is started. In this case it might be useful to check the return code from the DFHCSDUP step.

- The solution will work well in an environment where many (test) regions share the same CSD file. In CICS Version 4, where many lists may be concatenated in the GRPLIST parameter, it makes sense to use the same CSD VSAM file for many test regions. If more than one region uses the same CSD file, then it is impossible to nullify the 'member definitions' – a daily/weekly job must be run to nullify the member definitions.

- In our organization, TCLASS is very uncommon. Transactions are defined below the line, and apart from the TWA and the spurge + dtimout parameter, transactions are defined with the IBM defaults. It is easy to change the COBOL program code and the REXX code to read those parameters (and others) from the member and to create the resource according to the programmers' wishes.

- The COBOL program is started from the PLT and from a transaction, but it will run every hour. This is important so that application programmers know when the resource will be defined to CICS.
- We write a note to the SDSF (via another extra-partition TD) for any resource that CICS creates. CICS will write the created resources to the CSMT.
- The SPI CREATE command is discussed in *System Programming Reference*. This book is useful when analysing the response codes from the CREATE command, and also to extend the 'solution' if required.
- You will need CICS/ESA 4.1 or above to use this solution – the CREATE SPI was first introduced in this release. Apart from this limitation, the code could be used, with minor changes, at any CICS/MVS site.

We developed this solution for the test+verification environment, where performance is not a big issue, but where CICS system programmer response time is an issue.

SYSDEFR

```
 IDENTIFICATION      DIVISION.
******************************
 PROGRAM-ID. SYSDEFR.
 AUTHOR.  URICO
      ************************************************************
      *  THIS PROGRAM READS TDQ WHICH CONTAINS DEFINITIONS FOR   *
      *  CICS. AFTER SYNTAX CHECKS IT USES CREATE TO DEFINE THEM *
      ************************************************************
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
       Ø1  MY-DATA                       .
           Ø3  OPTI1        PIC X(1).
           Ø3  FILLER1      PIC X(1).
```

```
    Ø3  LANGTRAN    PIC X(4).
    Ø3  FILLER2     PIC X(1).
    Ø3  PROGNAME    PIC X(8).
    Ø3  FILLER3     PIC X(1).
    Ø3  SECTRAN     PIC X(1).
    Ø3  FILLER4     PIC X(1).
    Ø3  TWASIZE     PIC X(2).
    Ø3  FILLER5     PIC X(1).
    Ø3  IFBELOW     PIC X(1).
    Ø3  FILLER5     PIC X(58).
Ø1  TDNAME        PIC X(4)  VALUE 'DEFR'.
Ø1  TDOUT         PIC X(4)  VALUE 'DCPM'.
Ø1  TXT-LEN       PIC 9(4)  COMP  VALUE 9.
Ø1  TXT-MSG                                  .
    Ø3  TXT-PROG      PIC X(1Ø) VALUE 'SYSDEFR:  '       .
    Ø3  TXT-SAY       PIC X(3Ø)  .
    Ø3  TXT-VAR2      PIC X(1Ø) .
Ø1  STAT1         PIC S9(8) COMP.
Ø1  SWITCH        PIC S9(1) COMP VALUE Ø.
Ø1  HOWSTART      PIC X(2)            .
Ø1  UTIME         PIC S9(15) COMP-3   .
Ø1  ATIME         PIC X(8).
Ø1  FILLME REDEFINES ATIME .
    Ø3  CURR-HH     PIC 99  .
    Ø3  FILL1       PIC X(1).
    Ø3  CURR-MM     PIC 99  .
    Ø3  FILL2       PIC X(1).
    Ø3  CURR-SS     PIC 99  .
Ø1  CURR-HH-F       PIC S9(8) COMP.
Ø1  CRE-PROG                .
    Ø3  OPTION      PIC X(8) VALUE 'LANGUAGE'.
    Ø3  FILLER1     PIC X(1) VALUE '('      .
    Ø3  LANG        PIC X(8).
    Ø3  FILLER2     PIC X(1) VALUE ')'      .
Ø1  CRE-TRAN                 .
    Ø3  OPTION      PIC X(8)  VALUE 'PROGRAM '.
    Ø3  FILLER1     PIC X(1) VALUE '('      .
    Ø3  PROGCRE     PIC X(8)                .
    Ø3  FILLER2     PIC X(1)  VALUE ')'      .
    Ø3  FILLER3     PIC X(22) VALUE 'SPURGE(YES) DTIMOUT('.
    Ø3  DTIME       PIC X(3)  VALUE '1ØØ'.
    Ø3  FILLER4     PIC X(11) VALUE ') TWASIZE('.
    Ø3  ATWA        PIC X(2)  VALUE 'ØØ'.
    Ø3  FILLER5     PIC X(9)  VALUE ') RESSEC('.
    Ø3  YESNO       PIC X(3)                 .
    Ø3  FILLER6     PIC X(1)  VALUE ')'       .
Ø1  CRE-MAP                  .
    Ø3  FILLER1     PIC X(1) VALUE ' '      .
Ø1  MYREQ           PIC X(8) VALUE 'DEFCRQST'.
PROCEDURE DIVISION.
```

```
           EXEC CICS ASSIGN  STARTCODE(HOWSTART) END-EXEC.
           MOVE  HOWSTART     TO TXT-VAR2 .
           MOVE  'START OF PROGRAM IS ' TO TXT-SAY.
           EXEC CICS WRITEQ TD QUEUE(TDOUT)
           FROM(TXT-MSG) END-EXEC          .
           EXEC CICS IGNORE CONDITION NOTFND END-EXEC.
           EXEC CICS HANDLE CONDITION QZERO(LOOP-SOFF)
           NOTOPEN(OPEN-ERR) IOERR(IO-ERR) END-EXEC.
           IF HOWSTART = 'S '
      *    THE TRANSACTION WAS STARTED AUTOMATICALY
           EXEC CICS START TRANSID(EIBTRNID) INTERVAL(Ø1ØØØØ)
            REQID(MYREQ) END-EXEC
           ELSE
      *    THE TRANSACTION WAS STARTED MANUALLY OR FROM PLT
             EXEC CICS CANCEL  REQID(MYREQ) END-EXEC
             EXEC CICS ASKTIME ABSTIME(UTIME) END-EXEC
             EXEC CICS FORMATTIME ABSTIME(UTIME) DATESEP('-')
                TIME(ATIME) TIMESEP END-EXEC
             ADD 1 TO CURR-HH
             IF CURR-HH > 24
               SUBTRACT 24 FROM CURR-HH
             END-IF
             MOVE CURR-HH TO CURR-HH-F
             EXEC CICS START TRANSID(EIBTRNID) REQID(MYREQ) AT
             HOURS(CURR-HH-F)  MINUTES(Ø) SECONDS(Ø) END-EXEC
           END-IF
           MOVE DFHVALUE(CLOSED) TO STAT1
           EXEC CICS SET TDQUEUE(TDNAME) OPENSTATUS(STAT1)
           END-EXEC.
           MOVE DFHVALUE(OPEN) TO STAT1
           EXEC CICS SET TDQUEUE(TDNAME) OPENSTATUS(STAT1)
           END-EXEC.
      *            IF RESOURCE IS ACTIVE CREATE WILL FAIL
      *            IGNORE THE FAILURE AND CONTINUE LOOPING
           EXEC CICS IGNORE CONDITION INVREQ END-EXEC.
      *            LOOP UNTIL QUEUE IS EMPTY
           PERFORM  UNTIL SWITCH = 1
           EXEC CICS READQ TD QUEUE(TDNAME) INTO(MY-DATA)
           END-EXEC
           EVALUATE OPTI1
      *            IT IS A PROGRAM
           WHEN 'P'
             MOVE  'PROG DEFINITION' TO TXT-SAY
             MOVE   PROGNAME    TO  TXT-VAR2
             EVALUATE LANGTRAN
                 WHEN 'COB '
                  MOVE 'COBOL   ' TO LANG
                  EXEC CICS CREATE PROGRAM(PROGNAME) ATTRIBUTES
                  (CRE-PROG) ATTRLEN(LENGTH OF CRE-PROG) END-EXEC
                 WHEN 'C   '
```

```
                  MOVE 'C       ' TO LANG
                  EXEC CICS CREATE PROGRAM(PROGNAME) ATTRIBUTES
                  (CRE-PROG) ATTRLEN(LENGTH OF CRE-PROG) END-EXEC
                 WHEN 'ASS '
                  MOVE 'ASSEM   ' TO LANG
                  EXEC CICS CREATE PROGRAM(PROGNAME) ATTRIBUTES
                  (CRE-PROG) ATTRLEN(LENGTH OF CRE-PROG) END-EXEC
                 WHEN OTHER
                   MOVE 'INVALID LANG IN PROG' TO TXT-SAY
                   MOVE  PROGNAME          TO TXT-VAR2
                   GO TO LOOP-SOFF
```

CICSJOB

```
//CICSJOB  JOB .............(JOBCARD)
//*****************************************************
//**  STEP 1 :  CHECK WHETHER CICS START UP IS COLD
//**  STEP 2 :  IF IT IS THEN REBUILD THE JOB TO ADD THOSE
//**            DEFINITIONS TO THE CSD
//**  STEP 3 :  RUN THE DFHCSDUP UTILITY
//*************************************************
//STEP1  EXEC PGM=IKJEFTØ1,DYNAMNBR=1ØØ
//SYSOUT   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSPROC  DD   DISP=SHR,DSN=SYS2.PROCLIB
//         DD   DISP=SHR,DSN=SYS1.PROCLIB
//         DD   DISP=SHR,DSN=SYS1.CPAC.PROCLIB
//         DD   DISP=SHR,DSN=SYS1.CPAC.PROCLIB
//SYSTSPRT DD   SYSOUT=*
//IN       DD   DSN=CICS.JCL.OVERRIDE(CICSTEST),DISP=SHR
//SYSTSIN  DD   *
  PROFILE NOPREFIX
  EX 'MYREXX.LIB.EXEC(IFCOLD)'
//******IS IT A COLD START ***************************
//KUKU   IF (STEP1.RC LE Ø) THEN
//STEP2  EXEC PGM=IKJEFTØ1,DYNAMNBR=1ØØ
//SYSOUT   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSPROC  DD   DISP=SHR,DSN=SYS2.PROCLIB
//         DD   DISP=SHR,DSN=SYS1.PROCLIB
//         DD   DISP=SHR,DSN=SYS1.CPAC.PROCLIB
//         DD   DISP=SHR,DSN=SYS1.CPAC.PROCLIB
//IN       DD   DISP=SHR,DSN=GLOBAL.ACCESS.LIB(DEFRCICS)
//* IN MUST BE THE SAME HERE AS IN CICSJOB EXEC DEFRSRC   CARD
//OUT      DD   DISP=SHR,DSN=GLOBAL.ACCESS.LIB(DEFOCICS)
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
  PROFILE NOPREFIX
  EX 'MYREXX.LIB.EXEC(DEFCICS2)'
```

```
//KUKU   ENDIF
//************************************************************
//KUKU   IF (STEP1.RC LE Ø) AND (STEP2.RC LE 5) THEN
//STEP3 EXEC PGM=DFHCSDUP,REGION=4ØØK
//STEPLIB DD DSN=CICS41Ø.SDFHLOAD,DISP=SHR
//DFHCSD  DD DSN=CICS41Ø.DFHCSD,DISP=SHR
//* DFHCSD MUST BE THE SAME HERE AS IN CICSJOB EXEC DFHCSD DD CARD
//SYSOUT  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=GLOBAL.ACCESS.LIB(DEFOCICS),DISP=SHR
//KUKU   ENDIF
//**************************************************************
//CICSJOB  EXEC PGM=DFHSIP,PARM=('SYSIN')
//*
//STEPLIB  DD DSN=............................
//         .................................
//DFHRPL    DD DSN=...........................
//....DFHRPL CONTINUES ............
//SYSIN    DD   DSN=CICS.JCL.OVERRIDE(CICSTEST),DISP=SHR
//................................
//DFHCSD   DD DISP=SHR,DSN=CICS41Ø.DFHCSD
//...........................
//DEFRSRC DD DISP=SHR,DSN=GLOBAL.ACCESS.LIB(DEFRCICS)
```

DCT DEFINITIONS

```
DEFRSRC  DFHDCT TYPE=SDSCI,          CICS  JOB QUEUE                  *
               DSCNAME=DEFRSRC,                                        *
               TYPEFLE=INPUT
*
DEFR     DFHDCT TYPE=EXTRA,           CICS JOB QUEUE                   *
               DESTID=DEFR,                                            *
               DSCNAME=DEFRSRC,                                        *
               OPEN=DEFERRED
```

IFCOLD

```
/* REXX - PREPARE THE CSD JOB                        */
/* DO NOT START THE FIX PART ..........              */
TRACE ALL
/*ADDRESS TSO 'ALLOC FILE(IN) DA(SYSP.CICS41Ø.SYSIN(AØ1CICSU)) SHR'*/
"EXECIO * DISKR IN (FINIS STEM ROWBASE"
IF RC > Ø THEN DO
            SAY "ERROR READING DATASET :" DSNAME
            SIGNAL OUT
END
/* SO FAR WRITING TO FIX PAR IS COMPLETE */
CODE = 2Ø
```

```
DO I = 1 TO ROWBASEØ
ROW = VALUE('ROWBASE'||I)
 IF SUBSTR(ROW,1,5) = 'START' THEN HOW = SUBSTR(ROW,7,4)
END /* DO */
IF HOW='COLD'  THEN CODE = Ø
IF HOW='AUTO'  THEN CODE = 5
"EXECIO Ø DISKR IN (FINIS"
ADDRESS TSO "FREE F(IN)"
RETURN(CODE)
OUT:
  "EXECIO Ø DISKR IN (FINIS"
  ADDRESS TSO "FREE F(IN)"
  CODE = 16
  RETURN (CODE)
  EXIT
```

DEFCICS2

```
/* REXX - PREPARE THE SYSIN PART OF THE CSD JOB    */
/* BY PARSING INPUT LINES    ...........           */
TRACE ALL
"EXECIO * DISKR IN (FINIS STEM ROWBASE"
IF RC > Ø THEN DO
            SAY "ERROR READING DATASET :" DSNAME
            SIGNAL OUT
END
DO I = 1 TO ROWBASEØ
ROW = VALUE('ROWBASE'||I)
IF SUBSTR(ROW,1,1) = 'P' THEN DO
  OUTREC1 = 'DEFINE PROGRAM(' || SUBSTR(ROW,8,8) || ') LANG('
  OUTREC2 =  SUBSTR(ROW,3,4) || ') GR(SYSTEMP)'
END
IF SUBSTR(ROW,1,1) = 'M' THEN  DO
  OUTREC1 = 'DEFINE MAPSET(' || SUBSTR(ROW,8,8)
  OUTREC2 = ') GR(SYSTEMP)'
END
IF SUBSTR(ROW,1,1) = 'T' THEN DO
  SEC='NO'
  IF SUBSTR(ROW,17,1) = 'Y' THEN SEC='YES'
  IF SUBSTR(ROW,19,1) = ' ' THEN TTWA='ØØ'
     ELSE  TTWA = SUBSTR(ROW,19,2)
  OUTREC1 = 'DEFINE TRANSACTION(' || SUBSTR(ROW,3,4) ||  ') PROGRAM('
  OUTREC2 =  SUBSTR(ROW,8,8) || ') DTIMOUT(1ØØ) SPURGE(YES)'
  OUTREC3 = 'TWA(' || TTWA || ') RESSEC(' || SEC || ,
            ') GROUP(SYSTEMP)'
END
OUT1 = OUTREC1 || OUTREC2
IF SUBSTR(ROW,1,1) ¬= '*' THEN   DO
   PUSH OUT1
  "EXECIO 1 DISKW OUT "
END /* IF DO */
IF SUBSTR(ROW,1,1) = 'T' THEN   DO
   PUSH OUTREC3
  "EXECIO 1 DISKW OUT "
END /* IF DO */
END /* LOOP DO */
ADDRESS TSO "FREE F(IN)"
"EXECIO Ø DISKW OUT (FINIS"
OUT:
  "EXECIO Ø DISKR IN (FINIS"
  ADDRESS TSO "FREE F(IN)"
  "EXECIO Ø DISKW OUT (FINIS"
  ADDRESS TSO "FREE F(OUT)"
  EXIT
```

NULLIFYING JCL

```
//SØØ4JOB JOB  (SSØ4,A1,1Ø),URIC,MSGCLASS=T,NOTIFY=SØØ4
/*JOBPARM S=SYS1
//*
//* *------------------------------------------------------------*
//* *                                                            *
//* *  NULLIFYING DAILY CUMULATIVE DATASET                       *
//* *                                                            *
//* *------------------------------------------------------------*
//S1       EXEC PGM=IEBGENER
//SYSIN     DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=GLOBAL.ACCESS.LIB(DEFRCICS),DISP=SHR
//SYSUT2    DD  DSN=GLOBAL.ACCESS.LIB(ALLRCICS),DISP=MOD
//*
//* *------------------------------------------------------------*
//* *                                                            *
//* *  NULLIFYING DAILY CUMULATIVE DATASET                       *
//* *                                                            *
//* *------------------------------------------------------------*
//S2       EXEC PGM=IEBGENER
//SYSIN     DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSUT1    DD  DSN=NULLFILE,DISP=SHR,DCB=GLOBAL.ACCESS.LIB
//SYSUT2    DD  DSN=GLOBAL.ACCESS.LIB(DEFRCICS),DISP=OLD
//*
```

Uri Cohen
CICS System Programmer (Israel) © Xephon 1999

CICS news

CICS users can benefit from IBM's VisualAge for Java, Enterprise Edition for OS/390. The optional compiler feature can be used in conjunction with the run-time feature to develop compiled and bound Java programs. The run-time feature is required to execute fully-bound Java programs.

The compiler/binder statically compiles Java bytecodes directly into native object code and also binds the code into an executable or DLL that can be run in the OS/390 shell or under the CICS Transaction Server for OS/390.

With export and remote bind, class files can be sent from the workstation to OS/390 for final compilation and binding. On the OS/390, debug options include interpreted programs running in the JVM and compiled and bound Java programs running natively on the OS/390, either in the OS/390 Unix environment or under CICS.

The jport utility identifies the Java code that won't execute in the target OS/390 Unix and CICS environments, which don't support some parts of the JDK. Hence jport reads Java bytecode files and generates HTML files that list any unsupported packages, classes, methods, and fields.

For further information contact your local IBM representative.

* * *

CICS users can benefit from Version 4.1 of Neon Systems' ShadowDirect integration middleware for System/390. This incorporates CICS, DB2, IMS/DB, IMS/TM, ADABAS, VSAM, and all other sources into ODBC, application server, and common development tool execution environments.

Version 4.1 includes added support for IBM's Work Load Manager, DB2 stored procedure access, dynamic load-balancing, ADABAS access, and support for Microsoft Transaction Server, as well as access to OS/390 and MVS for Forte and BEA Tuxedo/M3 users.

For further information contact:
Neon Systems, 14141 Southwest Freeway, Suite 6200, Sugar Land, TX 77478, USA.
Tel: (281) 491 4200.
URL: http://www.neonsys.com.

* * *

IBM has announced enhancements to DataInterchange MVS/CICS and MVS. These translation components of IBM EDI services run on System/390 to provide MVS/CICS real-time processing and MVS batch processing respectively. Enhancements to Version 3.1 include the extraction of SAP records during translation; MQSeries message queueing; updating of the DataInterchange Client for 31-bit architecture; an expanded EDI control number assignment option; and event log conversion into a DB2 table.

For further information contact your local IBM representative.

* * *