February 1999

In this issue

3 Anchoring a WSA address using TRUE
8 Analysing abended transactions – part 3
25 Exploiting EXCI to manage CICS files from batch
38 Using the CICS 4.1 CREATE command
48 CICS news

© Xephon plc 1999
CICS Update

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.

© Xephon plc 1999. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs $36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.
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 TRUEEXIT,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.
The PLT program is as follows:

******************************************************************************
* PROGRAM NAME:    TRUEENAB
* *
* DESCRIPTION:    MAINLINE CODE THAT RUNS AT CICS
* *
******************************************************************************

******************************************************************************
* REGISTER EQUATES
******************************************************************************

******************************************************************************
* WORKING STORAGE DEFINITIONS
******************************************************************************

DFHEISTG DSECT
WSMESS DS CL50       CSMT MESSAGE FIELD
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=(10),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'  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 WSAREAL,R9
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= PROTOTYPE STATEMENT
AGO .PGNAME
.* G.RALLO TP 16/4/80
.* .PGNAME ANOP
.*
.* THIS VARIABLE FOR TIME AND DATE STAMPING
LCLC &VMTMDT TIME/DATE STAMP
LCLC &RELEASE VERSION
.*
.* AIF (T'&R NE '0').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
*================================================================
*
*================================================================****
 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 CL20 FUNCTION CODE DECODIFIED
DS CL1

ERC DS CL15 ERROR CODE DECODIFIED
DS CL1

ORG STDAREA1
DS CL8Ø

ORG *-3Ø

ERMSG DS ØCL30

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
R10 EQU 10
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=ØØ1

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   EIBCALLEN,=Y(DEERRØAF-DEERRØAI) VERIFY COMMAREA LEN
BL    COMER
MVC   ERMSGS,STDAREA       MOVE BLANK
CLI   ERRESNAM,C' '        MOVE RESOURCE NAME
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    NORSØ          .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,''
MVI   TIME+1,'-
UNPK  TKN,EIBTASKN
OI    TKN+L'TKN-1,X'FØ'
MVI   TKN+L'TKN,'-
UNPK  FNEIB(L'FNEIB+1),ERFUNCOD(3)
TR    FNEIB(L'FNEIB+1),TABEX-24Ø
MVI   FNEIB+L'FNEIB,'-
UNPK  RCODEEIB(L'RCODEEIB+1),ERFUNCOD+2(L'EIBRCODE+1)
TR    RCODEEIB(L'RCODEEIB+1),TABEX-24Ø
MVI   RCODEEIB+L'RCODEEIB,'-
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 0H
ST RWKR14,SAVE14
SH RWKR14,H'4'
LPR RWKR14,RWKR14
LA RWKR14,ERFUNCOD+2(RWKR14)
CLC 0(1,RWKR14),0(RWKR15)
L RWKR14,SAVE14
BNE RLOOPFN2
MVI SWF,X'FF'
B RLOOPERC

COMER DS 0H
MVC ERMG,=CL30'COMMAREA LENGTH ERROR'
B WRITETDA

WRITETD DS 0H
CLC EIBCALEN,=Y(L'ERFUNCOD+L'ERERRCOD)
BE WRITETDA
MVC ERMSGS,FNC

WRITETDA DS 0H
 *
CLI SWTD,X'FF'
BE RETURN
MVC RESC,=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 0H
 *
EXEC CICS RETURN
 *
TABEX DC C'0123456789ABCDEF'
 *
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'00000000' 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 XL2'0202',CL20'ADDRESS'
DC XL2'0204',CL20'HANDLE CONDITION'
DC XL2'0206',CL20'HANDLE AID'
DC XL2'0208',CL20'ASSIGN'
DC XL2'0210',CL20'IGNORE CONDITION'
DC XL2'0212',CL20'HANDLE AID SET'
DC XL2'0402',CL20'RECEIVE'
DC XL2'0404',CL20'SEND'
DC XL2'0406',CL20'CONVERSE'
DC XL2'0408',CL20'ISSUE EODS'
DC XL2'040A',CL20'ISSUE COPY'
DC XL2'040C',CL20'WAIT TERMINAL'
DC XL2'0410',CL20'WAIT SIGNAL'
DC XL2'0412',CL20'ISSUE RESET'
DC XL2'0414',CL20'ISSUE DISCONNECT'
DC XL2'0416',CL20'ISSUE ENDDOUTPUT'
DC XL2'0418',CL20'ISSUE ERASEUP'
DC XL2'041A',CL20'ISSUE ENDFILE'
DC XL2'041C',CL20'ISSUE PRINT'
DC XL2'041E',CL20'ISSUE SIGNAL'
DC XL2'0420',CL20'ALLOCATE'
DC XL2'0422',CL20'FREE'
DC XL2'0424',CL20'POINT'
DC XL2'0426',CL20'BUILD ATTACH'
DC XL2'0428',CL20'EXTRACT ATTACH'
DC XL2'042A',CL20'EXTRACT TCT'
DC XL2'042C',CL20'WAIT CONVID'
DC XL2'042E',CL20'EXTRACT PROCESS'
DC XL2'0430',CL20'ISSUE ABEND'
DC XL2'0432',CL20'CONNECT PROCESS'
DC XL2'0434',CL20'ISSUE CONFIRMATION'
DC XL2'0436',CL20'ISSUE ERROR'
DC XL2'0438',CL20'ISSUE PREPARE'
DC XL2'043A',CL20'ISSUE PASS'
DC XL2'043C',CL20'EXTRACT LOGONMSG'
DC XL2'043E',CL20'EXTRACT ATTRIBUTES'
DC XL2'5E32',CL20'WAITCICS'
DC XL2'0602',CL20'READ'
DC XL2'0604',CL20'WRITE'
DC XL2'0606',CL20'REWRITE'
DC XL2'0608',CL20'DELETE'
DC XL2'060A',CL20'UNLOCK'
DC XL2'060C',CL20'STARTBR'
DC XL2'060E',CL20'READNEXT'
DC XL2'0610',CL20'READPREV'
DC XL2'0612',CL20'ENDBR'
DC XL2'0614',CL20'RESETBR'
DC XL2'0802',CL20'WRITEQ TD'
DC XL2'0804',CL20'READQ TD'
DC XL2'0806',CL20'DELETEQ TD'
DC XL2'0A02',CL20'WRITEQ TS'
DC XL2'0A04',CL20'READQ TS'
DC XL2'0A06',CL20'DELETEQ TS'
DC XL2'0C02',CL20'GETMAIN'
DC XL2'0C04',CL20'FREEMAIN'
DC XL2'0E02',CL20'LINK'
DC XL2'0E04',CL20'XCTIL'
DC XL2'0E06',CL20'LOAD'
DC XL2'0E08',CL20'RETURN'
DC XL2'0E0A',CL20'RELEASE'
DC XL2'0E0C',CL20'ABEND'
DC XL2'0E0E',CL20'HANDLE ABEND'
DC XL2'1002',CL20'ASKTIME'
DC XL2'1004',CL20'DELAY'
DC XL2'1006',CL20'POST'
DC XL2'1008',CL20'START'
XL2'108A',CL20'RETRIEVE'
XL2'108C',CL20'CANCEL'
XL2'1284',CL20'WAIT EVENT'
XL2'1288',CL20'ENQ'
XL2'128A',CL20'DEQ'
XL2'128C',CL20'SUSPEND'
XL2'1402',CL20'WRITE JOURNAL'
XL2'1404',CL20'WAIT JOURNAL'
XL2'1682',CL20'SYNCPPOINT'
XL2'1684',CL20'RESYNC'
XL2'1802',CL20'RECEIVE MAP'
XL2'1804',CL20'SEND MAP'
XL2'1806',CL20'SEND TEXT'
XL2'1808',CL20'SEND PAGE'
XL2'180A',CL20'PURGE MESSAGE'
XL2'180C',CL20'ROUTE'
XL2'180E',CL20'RECEIVE PARTN'
XL2'1810',CL20'SEND PARTNSET'
XL2'1812',CL20'SEND CONTROL'
XL2'1A92',CL20'TRACE ON/OFF'
XL2'1A94',CL20'ENTER'
XL2'1CB2',CL20'DUMP'
XL2'1E82',CL20'ISSUE ADD'
XL2'1E84',CL20'ISSUE ERASE'
XL2'1E86',CL20'ISSUE REPLACE'
XL2'1E88',CL20'ISSUE ABORT'
XL2'1E8A',CL20'ISSUE QUERY'
XL2'1E8C',CL20'ISSUE END'
XL2'1EC2',CL20'ISSUE RECEIVE'
XL2'1E10',CL20'ISSUE NOTE'
XL2'1E12',CL20'ISSUE WAIT'
XL2'1E14',CL20'ISSUE SEND'
XL2'2002',CL20'BIF DEEDIT'
XL2'2202',CL20'EXIT ENABLE'
XL2'2204',CL20'EXIT DISABLE'
XL2'2206',CL20'EXIT EXTRACT'
XL2'4802',CL20'ENTER TRACENUM'
XL2'4804',CL20'MONITOR'
XL2'4A02',CL20'ASKTIME ABSTIME'
XL2'4A04',CL20'FORMATTIME'
XL2'4C02',CL20'INQUIRE FILE'
XL2'4C04',CL20'SET FILE'
XL2'4E02',CL20'INQUIRE PROGRAM'
XL2'4E04',CL20'SET PROGRAM'
XL2'5002',CL20'INQUIRE TRANSACTION'
XL2'5004',CL20'SET TRANSACTION'
XL2'5202',CL20'INQUIRE TERMINAL'
XL2'5204',CL20'SET TERMINAL'
XL2'5206',CL20'INQUIRE NETNAME'
XL2'5402',CL20'INQUIRE SYSTEM'
DC    XL2'5404',CL20'SET SYSTEM'
DC    XL2'5602',CL20'SPOOLOPEN'
DC    XL2'5604',CL20'SPOOLREAD'
DC    XL2'5606',CL20'SPOOLWRITE'
DC    XL2'5610',CL20'SPOOLCLOSE'
DC    XL2'5802',CL20'INQUIRE CONNECTION'
DC    XL2'5804',CL20'SET CONNECTION'
DC    XL2'5A02',CL20'INQUIRE MODENAME'
DC    XL2'5A04',CL20'SET MODENAME'
DC    XL2'5E06',CL20'CHANGE TASK'
DC    XL2'5E22',CL20'WAIT EXTERNAL'
DC    XL2'6614',CL20'SET TRANDUMPCODE'
DC    XL2'6A02',CL20'QUERY SECURITY'
DC    XL2'6C02',CL20'WRITE OPERATOR'
DC    XL2'6C12',CL20'ISSUE DFHWTO'
DC    XL2'7402',CL20'SIGN ON'
DC    XL2'7404',CL20'SIGN OFF'
DC    XL2'7E02',CL20'DUMP TRANSACTION'
DC    XL2'7E04',CL20'DUMP SYSTEM'
DC    XL2'820E',CL20'AP NOOP'
DC    XL2'8210',CL20'ALLOCATE'
DC    XL2'8212',CL20'CONVERSE FORMATTED'
DC    XL2'8214',CL20'CONVERSE DATASTREAM'
DC    XL2'8216',CL20'EXTRACT CONV'
DC    XL2'8218',CL20'EXTRACT FIELDS'
DC    XL2'821A',CL20'EXTRACT STSN'
DC    XL2'821C',CL20'FREE'
DC    XL2'821E',CL20'ISSUE'
DC    XL2'8220',CL20'RECEIVE FORMATTED'
DC    XL2'8222',CL20'RECEIVE DATASTREAM'
DC    XL2'8224',CL20'SEND FORMATTED'
DC    XL2'8226',CL20'SEND DATASTREAM'
DC    XL2'8228',CL20'START'
DC    XL2'8402',CL20'CICS NORMAL SHUTDOWN'
DC    XL2'8404',CL20'CICS IMMED. SHUTDOWN'
DC    XL2'8406',CL20'CICS FORCED SHUTDOWN'
DC    XL2'8408',CL20'CICS END-OF-TASK'
DC    XL2'840E',CL20'SP NOOP'
DC    XL2'8422',CL20'INQUIRE PROPERTYSET'
DC    XL2'8428',CL20'INSTALL PROPERTYSET'
DC    XL2'8430',CL20'DISCARD PROPERTYSET'
DC    XL2'8442',CL20'INQUIRE NODE'
DC    XL2'8444',CL20'SET NODE'
DC    XL2'8448',CL20'INSTALL Nodelist'
DC    XL2'844A',CL20'ADD POOL'
DC    XL2'844C',CL20'DELETE POOL'
DC    XL2'8450',CL20'DISCARD Nodelist'
DC    XL2'8462',CL20'INQUIRE POOL'
DC    XL2'8464',CL20'SET POOL'

© 1999. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
ENDTBFN EQU *

NTABEC DC Y((ENDTABEC-TABEC)/L'TABEC)

* EIBFN(BYTE 0) EIBRCODE(BYTES 0-3)

TABEC DS 0XL20

DC X'02',XL4'E0000000',CL15'INVREQ'
DC X'04',XL4'04000000',CL15'EOF'
DC X'04',XL4'10000000',CL15'EODS'
DC X'04',XL4'C1000000',CL15'EOF'
DC X'04',XL4'C2000000',CL15'ENDINPT'
DC X'04',XL4'D0000000',CL15'SYSIDERR'
DC X'04',XL4'D0040000',CL15'REQ.FUN.NOT.VAL'
DC X'04',XL4'D0040400',CL15'NO SESSION AVLB'
DC X'04',XL4'D0040800',CL15'MODENAME NOT.FO'
DC X'04',XL4'D0040C00',CL15'MODENAME NOT.VA'
DC X'04',XL4'D0041000',CL15'TASK.CANCELLED'
DC X'04',XL4'D0041400',CL15'MODE.GROUP.OUT'
DC X'04',XL4'D0041800',CL15'CLOSE.DRAIN=ALL'
DC X'04',XL4'D0080000',CL15'SYSID.OUT.OF.SE'
DC X'04',XL4'D00C0000',CL15'NAME.NOT= TCTSE'
DC X'04',XL4'D00C0400',CL15'NAME.NOT= REMOT'
DC X'04',XL4'D00C0800',CL15'MODE.NAME.NOTFO'
DC X'04',XL4'D00C0C00',CL15'PROFILE.NOT.FOU'
DC X'04',XL4'D0080000',CL15'LINK.OUT.OF.SRVC'
DC X'04',XL4'D00C0000',CL15'NAME.UNKNOWN'
DC X'04',XL4'D00C0400',CL15'NOT.NAME.OF.SE'
DC X'04',XL4'D00C0800',CL15'MODENAME.NOTFND'
DC X'04',XL4'D00C0C00',CL15'PROFILE.NOTFND'
DC X'04',XL4'D0200000',CL15'SESSIONERR'
DC X'04',XL4'D0204000',CL15'NOT.NAME.OF.S.E.'
DC X'04',XL4'D0208000',CL15'LINK.OUT.OF.SRVC'
DC X'04',XL4'D020C000',CL15'NAME.UNKNOWN'
DC X'04',XL4'D020C000',CL15'PROFILE.UNKNOWN'
DC X'04',XL4'D0300000',CL15'SYSBUSY'
DC X'04',XL4'D0300001',CL15'SYSBUSY.(TOR)'
DC X'04',XL4'D0300002',CL15'SYSBUSY.(TOR)'
DC X'04',XL4'D0400000',CL15'SESSBUSY'
DC X'04',XL4'D0500000',CL15'NOTALLOC'
DC X'04',XL4'E0000000',CL15'INVREQ'
DC X'04',XL4'E0000004',CL15'TE.ALRDY.ALCT'
DC X'04',XL4'E0000008',CL15'TE.WRONG.STATE'
DC X'04',XL4'E000000C',CL15'SYNCL2.NOT.SUP.'
X'04',XL4'EI000000',CL15'LENGERR'
X'04',XL4'E1040000',CL15'OUTPUT LENERR'
X'04',XL4'E1000000',CL15'INPUT LENERR'
X'04',XL4'EO1000000',CL15'LENGERR'
X'04',XL4'EO3000000',CL15'WRBRK'
X'04',XL4'EO4000000',CL15'RDATT'
X'04',XL4'EO5000000',CL15'SIGNAL'
X'04',XL4'EO6000000',CL15'TERMIERR'
X'04',XL4'EO7000000',CL15'N0PASSBKRD'
X'04',XL4'EO8000000',CL15'N0PASSBKWR'
X'04',XL4'EA0000000',CL15'IGREQCD'
X'04',XL4'EB0000000',CL15'CBIDERR'
X'04',XL4'F10000000',CL15'TERMERR'
X'04',XL4'F00200000',CL15'EOC'
X'04',XL4'F00400000',CL15'IMBFMH'
X'04',XL4'F00000006',CL15'N0START'
X'04',XL4'F00000007',CL15'NONVAL'
X'04',XL4'F00200000',CL15'EOC'
X'04',XL4'F00400000',CL15'INBFMH'
X'04',XL4'F00000006',CL15'N0START'
X'04',XL4'F00000007',CL15'NONVAL'
X'06',XL4'F01000000',CL15'DSIDERR'
X'06',XL4'F02000000',CL15'ILLOGIC'
X'06',XL4'F04000000',CL15'SEGIDERR'
X'06',XL4'F08000000',CL15'INVREQ'
X'06',XL4'F0C000000',CL15'NOTOPEN'
X'06',XL4'F0D000000',CL15'DISABLED'
X'06',XL4'F0F000000',CL15'ENDFILE'
X'06',XL4'F8000000',CL15'IOERR'
X'06',XL4'F8100000',CL15'NOTFIND'
X'06',XL4'F8200000',CL15'DUPREC'
X'06',XL4'F83000000',CL15'NOSPACE'
X'06',XL4'F84000000',CL15'DUPKEY'
X'06',XL4'F85000000',CL15'SUPPRESSED'
X'06',XL4'F86000000',CL15'LOADING'
X'06',XL4'F8000000',CL15'TASK CANCELLED'
X'06',XL4'F8000000',CL15'CLOSE DRAIN=ALL'

© 1999. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
| DC    | X'0A', XL4'D0040800', CL15'MODENAME NOT FO' |
| DC    | X'0A', XL4'D004C00', CL15'MODENAME NOT VA' |
| DC    | X'0A', XL4'D0040100', CL15'TASK CANCELLED' |
| DC    | X'0A', XL4'D0041400', CL15'MODE GROUP OUT' |
| DC    | X'0A', XL4'D0041800', CL15'CLOSE DRAIN=ALL' |
| DC    | X'0A', XL4'D0000000', CL15'LINK OUT OF SRVC' |
| DC    | X'0A', XL4'D00C0000', CL15'NAME UNKNOWN' |
| DC    | X'0A', XL4'D00C0400', CL15'NAME NOT= REMOT' |
| DC    | X'0A', XL4'D00C0800', CL15'MODE NAME NOTFO' |
| DC    | X'0A', XL4'D00C0C00', CL15'PROFILE NOT FOU' |
| DC    | X'0A', XL4'D0000000', CL15'LINK OUT OF SRVC' |
| DC    | X'0A', XL4'D00C0000', CL15'NAME UNKNOWN' |
| DC    | X'0A', XL4'D00C0400', CL15'NOT NAME OF SE' |
| DC    | X'0A', XL4'D00C0800', CL15'MODENAME NOTFND' |
| DC    | X'0A', XL4'D00C0C00', CL15'PROFILE NOTFND' |
| DC    | X'0A', XL4'D1000000', CL15'ISCINVREQ' |
| DC    | X'0A', XL4'D6000000', CL15'NOTAUTH' |
| DC    | X'0A', XL4'E1000000', CL15'LENGTHERR' |
| DC    | X'0C', XL4'E1000000', CL15'LENGTHERR' |
| DC    | X'0C', XL4'E2000000', CL15'NOSTG' |
| DC    | X'0E', XL4'01000000', CL15'PGMIDERR' |
| DC    | X'0E', XL4'D6000000', CL15'NOTAUTH' |
| DC    | X'0E', XL4'E0000000', CL15'INVREQ' |
| DC    | X'0E', XL4'D0000000', CL15'SYSIDERR' |
| DC    | X'0E', XL4'D0041400', CL15'SYSIDERR' |
| DC    | X'0E', XL4'F1000000', CL15'TERMERR' |
| DC    | X'10', XL4'01000000', CL15'ENDDATA' |
| DC    | X'10', XL4'04000000', CL15'IOERR' |
| DC    | X'10', XL4'11000000', CL15'TRANSIDERR' |
| DC    | X'10', XL4'12000000', CL15'TERMDERR' |
| DC    | X'10', XL4'14000000', CL15'INVTSREQ' |
| DC    | X'10', XL4'20000000', CL15'EXPIRED' |
| DC    | X'10', XL4'81000000', CL15'NOTFND' |
| DC    | X'10', XL4'D0000000', CL15'SYSIDERR' |
| DC    | X'10', XL4'D1000000', CL15'ISCINVREQ' |
| DC    | X'10', XL4'D6000000', CL15'NOTAUTH' |
| DC    | X'10', XL4'E1000000', CL15'LENGTHERR' |
| DC    | X'10', XL4'E9000000', CL15'ENVDEFERR' |
| DC    | X'10', XL4'FF000000', CL15'INVREQ' |
| DC    | X'12', XL4'32000000', CL15'ENQBUSY' |
| DC    | X'12', XL4'E0000000', CL15'INVREQ' |
| DC    | X'14', XL4'01000000', CL15'JIDERR' |
| DC    | X'14', XL4'02000000', CL15'INVREQ' |
| DC    | X'14', XL4'05000000', CL15'NOTOPEN' |
| DC    | X'14', XL4'06000000', CL15'LENGTHERR' |
| DC    | X'14', XL4'07000000', CL15'IOERR' |
| DC    | X'14', XL4'09000000', CL15'NOJBUFSP' |
| DC    | X'14', XL4'D0000000', CL15'NOTAUTH' |
| DC    | X'16', XL4'01000000', CL15'ROLLEDBACK' |
Editor’s note: this article will be concluded next month.

Giuseppe Rallo  
Senior Technical Analyst  
Sicilcassa (Italy)  
© Xephon 1999
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 ØØ CHARACTERS
  RECORDING MODE F
  LABEL RECORDS OMITTED.

Ø1 INPUT-RECORD    PIC X(ØØ).

WORKING-STORAGE SECTION.

COPY DFHXCPLO.

Ø1 OUTPUT-RETAREA.
  Ø5 FILLER             PIC X(ØØ) 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(Ø8) 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 'FILE REQUEST'.
    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
     "  "  "  "<<<

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(EXC1-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-NOFI LE
            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.
EXEC CICS HANDLE CONDITION ERROR(900-ERRORS) END-EXEC.

IF EIBCALEN = +24
  MOVE DFHCOMMAREA TO COMMAREA
ELSE
  EXEC CICS ABEND
PERFORM 100-INQ-FILE THRU 100-EXIT.

EVALUATE CMD-RESP
  WHEN DFHRESP(NORMAL)
    CONTINUE
  WHEN DFHRESP(FILENOTFOUND)
    SET CA-RC-NOFILE TO TRUE
    GO TO 000-EXIT
  WHEN OTHER
    SET CA-RC-FAILED TO TRUE
    GO TO 000-EXIT
END-EVALUATE.

PERFORM 200-SET-REQ-CVDAS THRU 200-EXIT.

IF CA-RC-GOLDEN
  CONTINUE
ELSE
  GO TO 000-EXIT
END-IF.

PERFORM 300-SET-FILE-INITIAL THRU 300-EXIT.

EVALUATE CMD-RESP
  WHEN DFHRESP(NORMAL)
    CONTINUE
  WHEN DFHRESP(INVREQ)
    SET CA-RC-NOTSET TO TRUE
    GO TO 000-EXIT
  WHEN OTHER
    SET CA-RC-FAILED TO TRUE
    GO TO 000-EXIT
END-EVALUATE.

IF CVDA-SET-OPE = CVDA-CLO-OPE
  AND CVDA-SET-ENA = CVDA-DIS-ENA
  GO TO 000-EXIT
END-IF.

PERFORM 400-SET-FILE-FINAL THRU 400-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(OPN) 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 200-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 200-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 200-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 200-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 200-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.

900-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'
 /**  20-80   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

JCL – SAMPLE EXECUTION 2

//EXCIRDO EXEC EXCIFILE,PARM='PRODCICS'
//SYSIN   DD *
ANYFILE1OPEENAYNNN
ANYFILE2OPEENAYNNN
ANYFILE3OPEENAYNNN

Russell Hunt
Senior Systems Programmer (USA) © Xephon 1999

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 unavailable 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
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.
  01 MY-DATA.
    03 OPTI1 PIC X(1).
    03 FILLER1 PIC X(1).
```
PROCEDURE DIVISION.

03 LANGTRAN PIC X(4).
03 FILLER2 PIC X(1).
03 PROGNAME PIC X(8).
03 FILLER3 PIC X(1).
03 SECTRAN PIC X(1).
03 FILLER4 PIC X(1).
03 TWASIZE PIC X(2).
03 FILLER5 PIC X(1).
03 IFBELOW PIC X(1).
03 FILLER5 PIC X(50).
01 TDNAME PIC X(4) VALUE 'DEFR'.
01 TDOUT PIC X(4) VALUE 'DCPM'.
01 TXT-LEN PIC 9(4) COMP VALUE 9.
01 TXT-MSG.
 03 TXT-PROG PIC X(10) VALUE 'SYSDEFR: ' .
 03 TXT-SAY PIC X(30) .
 03 TXT-VAR2 PIC X(10) .
01 STAT1 PIC S9(8) COMP.
01 SWITCH PIC S9(1) COMP VALUE 0.
01 HOWSTART PIC X(2) .
01 UTIME PIC S9(15) COMP-3 .
01 ATIME PIC X(8).
01 FILLME REDEFINES ATIME .
 03 CURR-HH PIC 99 .
 03 FILL1 PIC X(1).
 03 CURR-MM PIC 99 .
 03 FILL2 PIC X(1).
 03 CURR-SS PIC 99 .
01 CURR-HH-F PIC S9(8) COMP.
01 CRE-PROG .
 03 OPTION PIC X(8) VALUE 'LANGUAGE'.
 03 FILLER1 PIC X(1) VALUE '(' .
 03 LANG PIC X(8).
 03 FILLER2 PIC X(1) VALUE ')'.
01 CRE-TRAN .
 03 OPTION PIC X(8) VALUE 'PROGRAM'.
 03 FILLER1 PIC X(1) VALUE '(' .
 03 PROGCRE PIC X(8) .
 03 FILLER2 PIC X(1) VALUE ')'.
 03 FILLER3 PIC X(22) VALUE 'SPURGE(YES) DTIMOUT('.
 03 DTIME PIC X(3) VALUE '100'.
 03 FILLER4 PIC X(11) VALUE ')') TWASIZE('.
 03 ATWA PIC X(2) VALUE '00'.
 03 FILLER5 PIC X(9) VALUE 'RESSEC('.
 03 YESNO PIC X(3) .
 03 FILLER6 PIC X(1) VALUE ')'.
01 CRE-MAP .
 03 FILLER1 PIC X(1) VALUE '.
01 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 AUTOMATICALLY
EXEC CICS START TRANSID(EIBTRNID) INTERVAL(010000) 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(0) SECONDS(0) 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=IKJEFT01,DYNAMNBR=100
//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 
//SYSTSPT 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 0) THEN 
//STEP2  EXEC PGM=IKJEFT01,DYNAMNBR=100
//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) 
//SYSTSPT DD  SYSOUT=* 
//SYSTSIN DD  * 
// PROFILE NOPREFIX
//EX 'MYREXX.LIB.EXEC(DEFICICS2)'
IFCOLD

/* REXX - PREPARE THE CSD JOB */
/* DO NOT START THE FIX PART .......... */
TRACE ALL
/*ADDRESS TSO 'ALLOC FILE(IN) DA(SYS.P.CICS410.SYSIN(A01CICSU)) SHR'*/
"EXECIO * DISKR IN (FINIS STEM ROWBASE"
IF RC > 0 THEN DO
    SAY "ERROR READING DATASET :" DSNAME
    SIGNAL OUT
END
/* SO FAR WRITING TO FIX PAR IS COMPLETE */
CODE = 20
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 PARING 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(100) 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
**
//S004JOB JOB (SS04,A1,10),URIC,MSGCLASS=T,NOTIFY=S004
/*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 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.

***

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.

***