157 CICS update

December 1998

In this issue

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

North American office
Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75077-2150
USA
Telephone: 940 455 7050

Contributions
Articles published in *CICS Update* are paid for at the rate of £170 ($250) per 1000 words and £90 ($140) per 100 lines of code for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

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

Editor
Robert Burgess

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

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

Printed in England.

Submitting JCL from CICS to JES

INTRODUCTION

This article describes a mechanism for submitting JCL from CICS to a dynamically allocated JES internal reader. It also returns the JES JOBNUM to the user transaction for later identification of the output.

There are other ways of submitting a job to JES from CICS, but the main feature of the method described here is that the JOBCARD does not need to have USER=userid and PASSWORD=password hard-coded, but still runs under the user-id of the end user.

I did try using the SPOOLOPEN SPOOLWRITE API interface, but this requires USER and PASSWORD coded on the JOBCARD – which is not suitable for my site. Alternatively, it would require SURROGAT class profiles to be created to allow the CICS region user-id to submit jobs on behalf of another user. This was considered to be both an administrative burden and incompatible with existing production usage.

SOFTWARE REQUIRED

This facility was developed under CICS 4.1, OS/390 1.2, and RACF 1.9.2. I am not aware of any reason why it should not work on lower or higher-level versions of these products. I have tried it with MVS/ESA 5.2.2, RACF 1.9.2, and OS/390 Version 1.3/OS/390 Security Server 1.3. All programming techniques are fairly conventional and should apply to a range of different levels of these products.

HOW IT WORKS

To submit a job to JES without coding USER= and PASSWORD= in the JOBCARD, the MVS TCB that is in control at the time the JCL is copied to the internal reader (INTRDR) must present the ACEE of the end user to RACF.

This can be done by updating the TCBSENV in the TCB with the

ACEE address of the user. If TCBSENV is left as hex zeros, the ACEE address in the ASCB will determine the user-id against which any permissions will be checked.

If we do not attempt to set TCBSENV to the ACEE of the user, the JCL will be submitted by the CICS region user-id and would need access to the resources required by the Job to be submitted.

Unfortunately, updating the TCB requires us to be authorized. Even though the CICS STEPLIB concatenation is authorized, the CICS QR TCB running program DFHSIP is not. This is because CICS relinquishes authorization at the earliest possible point during region initialization.

A solution to this problem is to create a user-written SVC, within which we can update the TCBSENV field. I chose SVC number 237, which was unused at my site – you may need to choose another number. This SVC can be used by anyone and so the security people at your site will certainly require access to be restricted to the CICS region user-id. The SVC checks a FACILITY class profile 'CICSSVC237' to which the CICS region user-id is granted READ access.

After switching to the environment of the user, the INTRDR is opened using VSAM GENCB and MODCB macros to create an ACB and an RPL that can be opened in the usual way. It isn't necessary to retain the environment of the user any longer, so, before writing the JCL to the INTRDR, we can reset the TCBSENV field to zeros and return to the security level of the CICS region user.

When the JCL cards have been submitted, the ENDREQ macro is issued to obtain the JES job-number. If JES doesn't run the job for some reason once the JCL is submitted, this status is not passed back to the user. During testing I noticed that the job number of the last successfully run job was returned in such circumstances. You will need to devise a solution to this if it is a problem for you. A simple suggestion may be to compare the current job number to the previously returned job number and flag an alert if it is the same.

If the CICS region JCL does not include a DD card for an internal reader then one is dynamically allocated. A relatively unsophisticated

search of the TIOT is performed looking for a DDNAME of INTRDR – which was a standard DDNAME at my site for internal readers in CICS region JCL. Because of the number of CICS regions, we generally will not allocate internal readers in CICS region JCL, preferring to allocate and unallocate them dynamically as needed. Furthermore, we serialize allocation of internal readers so as not to allocate more than one at a time.

MVS SUBTASK ATTACH

Serializing allocation of the internal reader and I/O are not wise operations under the CICS QR TCB, because the normal operation of CICS would be suspended for long periods. Using CICS facilities to ENQUEUE a resource named CSGSUBM will avoid halting CICS on the allocation of the internal reader, by suspending the task pending the availability of the CSGSUBM resource.

We can solve the I/O problem by ATTACHing another task under CICS to perform the I/O without delaying the CICS main task. An EXEC CICS WAIT EXTERNAL call will post an ECB when the I/O has finished and the subtask detached, and resume the user's CICS task.

If you were to submit many jobs, then the repeated ATTACH/DETACH activity may be better solved by permanently creating a task to which the JCL is passed when required. It was not perceived to be a significant problem at my site, so I chose simply to create a new task and allocate an internal reader when required. Five parameters are passed to this attached task:

- A pointer to the address of the user's ACEE.
- The JCL stream in multiples of 80.
- A pointer to the address of the 80-byte card image JCL.
- A pointer to 8 bytes where the job number can be returned.
- A pointer to a fullword return code field.

Note that no check is performed to verify that the length passed is the true end of the JCL.

EXAMPLE PROGRAM

An example program illustrating the submission of JCL from CICS follows:

```
 PROCESS DATA(31),NODYNAM,RENT,NONUMBER,NOSEQUENCE,LIB
 PROCESS APOST LIST
 PROCESS OPTIMIZE
 IDENTIFICATION DIVISION.
 PROGRAM-ID. TESTDJCL.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*****************************************************************
*
* Program name         : TESTDJCL
* Tranid               : TSUB
* CICS version         : 4.1
* Written by           : Wilfred Kazoks
* Year 2000 dates      :
* COBOL compile parms  : DATA(31),NODYNAM,RENT,NONUMBER,NOSEQ,LIB
* Version              : 1.0
*
* MAPS                 : NONE
*
*
* RDO Parms             TRANSaction    : TSUB
*                       Group          : GSSUBM
*                       TWasize        : 00000
*                       PROFile        : DFHCICST
*                       PArtitionset   :
*                       TASKDATALoc    : Any
*                       TASKDATAKey    : USER
*                       ISolate        : Yes
*                       Cmdsec         : No
*
*
* Control Blocks used.
*
* Dependencies on other programs.
*
*-------------------------------
*
* Function:
*
* TSUB is a test transaction for CSGDJCL.
*-------------------------------
 1 WORKFLDS.
   2 RESP-CODE              PIC 9(8) BINARY.
   2 TEST-MESSAGE.
     3 RET-CODE             PIC X(12) VALUE 'RETURN CODE:'.
```

```
           3 RC                    PIC X VALUE '-'.
           3 JOBNUMB-MESSAGE       PIC X(8) VALUE 'JOBNUMB:'.
           3 JOBNUMB               PIC X(8) VALUE SPACE.
       *CBL NOSOURCE
        COPY DFHAID.
       *CBL SOURCE
        LINKAGE SECTION.
        1 DFHCOMMAREA.
          2 COMM-JOBNAME           PIC X(8).
          2 COMM-RETURN-CODE       PIC X(4).
          2 COMM-JCL-LENGTH        PIC 9(8) BINARY.
          2 COMM-JCL-ADDRESS       USAGE IS POINTER.
        1 JCL.
          2 JCL-CARDS              PIC X(8Ø) OCCURS 4 VALUE SPACES.
        PROCEDURE DIVISION.
        MAIN-PROCESSING SECTION.
            EXEC CICS GETMAIN SET(ADDRESS OF DFHCOMMAREA) RESP(RESP-CODE)
                              FLENGTH(LENGTH OF DFHCOMMAREA)
                              END-EXEC.
            IF RESP-CODE NOT EQUAL DFHRESP(NORMAL) THEN
              EXEC CICS ABEND ABCODE('GMCA') END-EXEC.
            EXEC CICS GETMAIN SET(ADDRESS OF JCL)
                              FLENGTH(LENGTH OF JCL) RESP(RESP-CODE)
                              END-EXEC.
            IF RESP-CODE NOT EQUAL DFHRESP(NORMAL) THEN
              EXEC CICS ABEND ABCODE('GMCB') END-EXEC.
            MOVE '//C6Ø5Ø88A JOB (YWSDAØØ5),''CICS JCL SUBMIT'','
              TO JCL-CARDS(1).
            MOVE '// MSGCLASS=X,CLASS=A'
              TO JCL-CARDS(2).
            MOVE '//STEP1 EXEC PGM=IEFBR14'
              TO JCL-CARDS(3).
            MOVE '//DD2 DD DISP=SHR,DSN=TPCICS.WILF.CNTL'
              TO JCL-CARDS(4).
            MOVE 32Ø TO COMM-JCL-LENGTH.
            SET COMM-JCL-ADDRESS TO ADDRESS OF JCL.
             EXEC CICS LINK PROGRAM('CSGDJCL') COMMAREA(DFHCOMMAREA)
                 RESP(RESP-CODE) END-EXEC.
            MOVE COMM-JOBNUMB TO JOBNUMB.
            MOVE COMM-RETURN-CODE(4:1) TO RC.
            EXEC CICS SEND TEXT FROM(TEST-MESSAGE) END-EXEC.
            GOBACK.
        END PROGRAM TESTDJCL.
```

CSGDJCL

This program will attach an MVS TCB:

```
*************************************************************************
*                                                                       *
```

```
* PROGRAM NAME : CSGDJCL                                               *
* VERSION      : 1.Ø                                                   *
* MODULE NAME  : CSGDJCL                                               *
* LANGUAGE     : ASSEMBLER                                             *
* MODE         : COMMAND LEVEL                                         *
* TRANSID      : NONE                                                  *
* CSDGROUP     : GSSUBM                                                *
* FUNCTION     : SUBMIT JCL TO JES WITH JOBNUMBER RETURN               *
* WRITTEN BY   : WILFRED KAZOKS                                        *
*                                                                      *
*----------------------------------*
*                                                                      *
* DESCRIPTION  : THIS PROGRAM ALLOWS JOB SUBMISSION WITHOUT THE NEED   *
*                TO SUPPLY A PASSWORD. THE ACTUAL SUBMISSION IS DONE   *
*                USING AN MVS SUBTASK WHERE THE INTERNAL               *
*                READER IS OPENED, THE JCL CARDS WRITTEN OUT, AND      *
*                THE READER THEN CLOSED. THE INTERNAL READER IS        *
*                DYNAMICALLY ALLOCATED IF THE REQUIRED DD CARD WAS     *
*                NOT SUPPLIED. THE PROPAGATION OF THE USER'S RACF      *
*                ID IS ACCOMPLISHED BY SETTING THE TCBSENV FIELD OF    *
*                THE SUBTASK TO THE USER'S ACEE.                       *
*                                                                      *
*----------------------------------*
*                                                                      *
* DEPENDENCIES : THIS PROGRAM ATTACHES CSGSUBMA                        *
*                WHICH IN TURN USES SVC 237                            *
*----------------------------------*
*                                                                      *
************************************************************************
*
*  PARAMETERS  ON INVOKATION (COBOL)         (ASSEMBLER)
*
*  USER-ACEE-ADDRESS PIC 9(4) BINARY.          F
*  JOBNUMB           PIC X(8).                 CL8
*  RETURN-CODE       PIC X(4).                 F  IN RIGHTMOST BYTE
*  JCL-BYTE-COUNT    PIC 9(4) BINARY.          F
*  JCL-ADDRESS       USAGE IS POINTER.         F
*
************************************************************************
* RETURN CODES
* 'Ø' OK  N.B. EBCDIC X'FØ'
* 'S' FAILED TO ATTACH SUBTASK ROUTINE 'CSGSUBMA'
* 'R' NO VALID USER SECURITY ENVIRONMENT FOUND (ACEE)
* 'A' FAILED TO ALLOCATE AN INTERNAL READER
* 'T' FAILED TO CHANGE TO USERS SECURITY ENVIRONMENT (SVC 237)
* 'O' FAILED TO OPEN THE INTERNAL READER
* 'Z' FAILED TO OPEN VSAM RPL FOR INTERNAL READER I/O
*
*
************************************************************************
```

```
CSGDJCL TITLE 'CICS BATCH JOBS SUBMISSION PROGRAM'
CSGDJCL AMODE 31
CSGDJCL RMODE ANY
         YREGS
DFHEISTG DSECT
ATTACHD  ATTACHX SF=L
RESP_CODE  DS F
RESP2_CODE DS F
ECB_ADDR_LIST DS F              LIST OF ECB ADDRESSES
DT_ATCB  DS    F                ADDR OF ATTACHED TCB
ACEEADDR DS    F                SAVE AREA FOR ACEE
ST_PRM   DS    5F               PARM LIST FOR INTRDR SUBTASK ATTACH
         ORG   ST_PRM
ST_PRM1  DS    F                @@ ACEE
ST_PRM2  DS    F                @ LENGTH OF JCL TO SUBMIT
ST_PRM3  DS    F                @@ OF JCL
ST_PRM4  DS    F                @ OF RETURN JOBNAME
ST_PRM5  DS    F                @ OF RETURN CODE
ENQUEUE  DS    CL8              ENQUEUE NAME
*
ECB_STG  DSECT                  ECB GOES TO SHARED CICS STORAGE
DT_AECB  DS    F                ECB TO WAIT ON WHILE COPY TO INTRDR
ECB_END  DS    ØH
*
COMMA    DSECT                  COMMAREA TO BE PASSED
JOBNAME  DS    CL8              RETURN JOBNAME TO THIS ADDRESS
RETC     DS    CL4              RETURN CODE
JCL_LENGTH DS  F                LENGTH OF JCL CARDS TO SUBMIT
ADDRESS_JCL DS F                POINTER TO JCL TO SUBMIT
*
CSGDJCL  DFHEIENT CODEREG=(R9),DATAREG=(R1Ø),EIBREG=(R11)
EXEC  CICS ADDRESS ACEE(R7) COMMAREA(R8)
         USING COMMA,R8
         CL    R7,=X'FFØØØØØØ'
         BE    LBLØ95Ø
         ST    R7,ACEEADDR
         EXEC CICS ENQ RESOURCE(ENQNAME) LENGTH(7)
         MVC   ATTACHD(ATTACHL),ATTACHS   COPY STATIC TO DYNAMIC LIST
         EXEC CICS GETMAIN SHARED CICSDATAKEY FLENGTH(ECB_LENGTH)      X
                   SET(R5) INITIMG(NULL) RESP(RESP_CODE)
         USING ECB_STG,R5
         ST    R5,ECB_ADDR_LIST
         ATTACHX SF=(E,ATTACHD),MF=(E,ST_PRM),ECB=((R5)),              X
               PARAM=(ACEEADDR,JCL_LENGTH,ADDRESS_JCL,JOBNAME,RETC)
         ST    R1,DT_ATCB            STORE TCB ADDR OF  TASK
         LTR   R15,R15               CHECK ATTACH
         BNZ   LBLØ9ØØ
         LA    R5,ECB_ADDR_LIST
         EXEC CICS WAIT EXTERNAL RESP(RESP_CODE) RESP2(RESP2_CODE)     X
              ECBLIST(R5) NAME('CSGSUBMA')                             X
```

```
             NUMEVENTS(1) NOTPURGEABLE
          DETACH DT_ATCB
          EXEC CICS DEQ RESOURCE(ENQNAME) LENGTH(7)
          L     R5,ECB_ADDR_LIST
          EXEC CICS FREEMAIN DATAPOINTER(R5) RESP(RESP_CODE)
          B     LBL9999
LBLØ9ØØ   DS    ØH
          ICM   RØ,B'ØØØ1',=C'S'    ATTACH SUBTASK FAILED
          ST    RØ,RETC
          B     LBL9999
LBLØ95Ø   DS    ØH                   USER HAS NO RACF ACEE
          ICM   RØ,B'ØØØ1',=C'R'
          ST    RØ,RETC
          B     LBL9999
LBL9999   EXEC  CICS RETURN
          LTORG
          DS    ØF
ENQNAME   DC    CL7'CSGSUBM'         RESOURCE NAME FOR ENQ
ATTACHS   ATTACHX SF=L,EP=CSGSUBMA MODULE TO ATTACH
ATTACHL   EQU   *-ATTACHS
ECB_LENGTH DC A(ECB_END-ECB_STG)
NULL      DC    X'ØØ'
          PRINT OFF
          IHAACEE
          END
```

CSGSUBMA

This program is attached as an MVS task to allocate an internal reader and write a job to a JES reader. It also uses SVC 237 to switch the TCBSENV field to the ACEE of the end-user and return the job number of the submitted job:

```
*********************************************************************
*                                                                   *
* PROGRAM NAME    : CSGSUBMA                                        *
* TRANSACTION NAME: NONE                                            *
* RDO             : NOT REQUIRED, MODULE LOADED FROM STEPLIB        *
* MAPS            : NONE
* RESIDENT MODE   : 24                                              *
* ADDRESS MODE    : ANY                                             *
* BINDER ATTRIBS  : NORENT NOREUS                                   *
* AUTHORIZATION   : NOT AUTHORIZED                                  *
*                                                                   *
* FUNCTION        :                                                 *
*                                                                   *
* WRITTEN BY      : CICS SUPPORT GROUP - WILFRED KAZOKS             *
*                                                                   *
```

```
* INPUT PARMS     : R1 ->     +-----------------------------------+    *
*                           +Ø |@@ USER ACEE                     |    *
*                              |                                 |    *
*                              +---------------------------------+    *
*                           +4 |@ FULLWORD BINARY LENGTH FIELD   |    *
*                              |OF 8Ø BYTE JCL CARDS FOLLOWING   |    *
*                              +---------------------------------+    *
*                           +8 |@@STORAGE CONTAINING JCL CARDS   |    *
*                              |                                 |    *
*                              +---------------------------------+    *
*                          +12| POINTER TO 8 BYTES WHERE JOBNAME |    *
*                              | OF SUBMITTED JOB IS PASSED BACK |    *
*                              +---------------------------------+    *
*                          +16| POINTER TO 4 BYTES WHERE RETURN |    *
*                              | CODE IS PASSED BACK             |    *
*                              +---------------------------------+    *
* RETURN CODES    :                                                   *
*                                                                     *
***********************************************************************
         TITLE     'CSGSUBMA - SUBMIT JCL IN CICS'
*NOTE*NOTE**NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE
*
* NOTE THE CONDITIONAL DEBUG ASSEMBLY TO INSERT A WTOR
*
*NOTE*NOTE**NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE*NOTE
         GBLC  &DEBUG
&DEBUG   SETC  'FALSE' NO INSERT OF WTOR
*DEBUG   SETC  'TRUE' ENABLE DEBUG MODE WITH WTOR INCLUDED
CSGSUBMA AMODE 31
CSGSUBMA RMODE 24
CSGSUBMA CSECT
         LR    R12,R15             ESTABLISH PROGRAM BASE REG
         USING CSGSUBMA,R12
         CSGVER 1.Ø
         LR    R8,R14
         LR    R1Ø,R1              SAVE INPUT PARM ADDR IN R1Ø
         STORAGE OBTAIN,LENGTH=DYNDATL,COND=YES
         LR    R11,R1              ESTABLISH DYNAMIC VARIABLE BASE
         USING DYNDAT,R11
         ST    R8,RETADDR
         L     R1,Ø(,R1Ø)          RETRIEVE @ ACEE @
         L     R1,Ø(,R1)
         ST    R1,ACEEADDR         SAVE ACEE POINTER
         L     R1,4(,R1Ø)          GET ADDRESS OF JCL LENGTH
         L     R1,Ø(,R1)           GET BINARY JCL LENGTH
         ST    R1,JCLLEN           BYTE LENGTH ON JCL DECK
         L     R1,8(,R1Ø)
         L     R1,Ø(,R1)
         ST    R1,JCLADDR          ADDRESS OF JCL TO SUBMIT
         L     R1,12(,R1Ø)
```

```
         ST    R1,JOBNAME_ADDRESS  SAVE LOC'N OF JOBNAME SAVEAREA
         L     R1,16(,R1Ø)
         ST    R1,RETC_ADDRESS     SAVE LOC'N OF RETURN CODE SAVEAREA
         L     R2,CVTPTR           GET CVT ADDR
         USING CVTMAP,R2
         L     R2,CVTTCBP          GET ADDR OF TCB WORDS
         L     R3,4(Ø,R2)          GET ADDR OF MY TCB
         USING TCB,R3
         L     R2,TCBTIO            GET ADDR OF TIOT
         USING TIOT1,R2             MAP TIOT
LBLØØØ1  DS    ØH
         CLI   TIOELNGH,X'ØØ'       END OF TIOT?
         BZ    LBLØØØ5
         TM    TIOESTTA,TIOSLTYP    FREED TIOT ENTRY?
         BO    LBLØØØ3 YES SKIP TIOT FREE SPACE
         CLC   TIOEDDNM,=C'INTRDR  '
         BNE   LBLØØØ3
         OI    FLAG,INTRDR
         B     LBLØØØ5
LBLØØØ3  DS    ØH
         SR    R1,R1
         IC    R1,TIOELNGH         GET LENGTH OF TIOT
         AR    R2,R1               POINT TO NEXT TIOT ENTRY
         B     LBLØØØ1             BACK TO START
LBLØØØ5  DS    ØH
         TM    FLAG,INTRDR         IF NO INTRDR THEN DYNALLOC IT
         BO    LBLØ1ØØ             ALREADY IN JCL SO DON'T DYNALLOC
         LA    R1Ø,RBP
         USING S99RBP,R1Ø
         LA    R9,RB
         USING S99RB,R9
         ST    R9,S99RBPTR         MAKE RBP POINT TO RB
         OI    S99RBPTR,S99RBPND   TURN ON THE HIGH ORDER BIT IN RBPTR
         DROP  R1Ø
         XC    S99RB(RBLEN),S99RB  ZERO OUT ENTIRE RB
         MVI   S99RBLN,RBLEN       FILL IN RB LENGTH
         MVI   S99VERB,S99VRBAL    SET VERB CODE FIELD TO ALLOCATION
         LA    R1Ø,TEXT1           BUILD POINTER TABLE FOR EACH TEXT
         ST    R1Ø,TXTUNIT1        UNIT
         LA    R1Ø,TEXT2
         ST    R1Ø,TXTUNIT2
         LA    R1Ø,TEXT3
         ST    R1Ø,TXTUNIT3
         LA    R1Ø,TXTUNIT3
         USING S99TUPL,R1Ø
         OI    S99TUPTR,S99TUPLN   SET HIGH ORDER BIT TO MARK END
         LA    R1Ø,TXTUNIT1
         ST    R1Ø,S99TXTPP        POINT RB AT TEXTUNIT POINTERS
         LA    R1,RBP
         DYNALLOC
```

```
         LTR   R15,R15
         BNZ   LBL91ØØ             DYNALLOC FAILED
LBLØ1ØØ  DS    ØH
         SR    RØ,RØ
         L     R1,ACEEADDR
         SVC   237                SWITCH TO USER ACEE
         LTR   R15,R15
         BNZ   LBL92ØØ
         GENCB BLK=ACB,DDNAME=INTRDR
         LR    R6,R1              ADDRESS OF ACB IN R6
         GENCB BLK=RPL,                                          +
               OPTCD=(MVE,ADR,SEQ,SYN,NUP),                      +
               ACB=(R6),                                         +
               RECLEN=8Ø,                                        +
               AREALEN=8Ø
         LTR   R15,R15            SAVE RETURN CODE
         BNZ   LBL95ØØ
         LR    R5,R1              ADDRESS OF RPL IN R5
         LA    R1,CARD
         MODCB RPL=(R5),AREA=(R1) R1 -> DATA TO BE SENT TO INTRDR
         LTR   R15,R15            SAVE RETURN CODE
         BNZ   LBL95ØØ
         OPEN  ((R6),(OUTPUT))
         LTR   R15,R15
         BNZ   LBL93ØØ
         SR    RØ,RØ
         SR    R1,R1
         SVC   237                 SWITCH BACK TO CICS ACEE TCBSENV=Ø
         LTR   R15,R15
         BNZ   LBL92ØØ
         L     R2,JCLLEN           GET ADDRESS OF BINARY JCL LENGTH
         L     R3,JCLADDR          GET ADDR OF JCL
         AR    R2,R3               SET ADDRESS OF JCL END
LBLØ2ØØ  DS    ØH
         MVC   CARD,Ø(R3)          GET NEXT CARD FOR INTRDR
         PUT   RPL=(R5)
         LA    R3,8Ø(R3)           GET NEXT JCL CARD
         CR    R2,R3
         BH    LBLØ2ØØ
         ENDREQ RPL=(R5)           GET JES JOB NUMBER
         USING IFGRPL,R5
         L     R3,JOBNAME_ADDRESS
         MVC   Ø(L'RPLRBAR,R3),RPLRBAR
         CLOSE ((R6))
         LA    R2,Ø
         ICM   R2,B'ØØØ1',=C'Ø'    SET GOOD RETURN CODE
         B     LBL9999
LBL91ØØ  DS    ØH
         LA    R2,Ø
         ICM   R2,B'ØØØ1',=C'A'    SET ALLOCATION FAILED RETURN CODE
```

```
         B     LBL9999
LBL92ØØ  DS    ØH
         LA    R2,Ø
         ICM   R2,B'ØØØ1',=C'T'    SET ACEE CHANGE FAILED RETURN CODE
         B     LBL9999
LBL93ØØ  DS    ØH
         LA    R2,Ø
         ICM   R2,B'ØØØ1',=C'O'    SET OPEN INTRDR FAILED RETURN CODE
         B     LBL9999
LBL95ØØ  DS    ØH
         LA    R2,Ø
         ICM   R2,B'ØØØ1',=C'Z'    SET OPEN RPL    FAILED RETURN CODE
         B     LBL9999
LBL9999  DS    ØH
         AIF   ('&DEBUG' NE 'TRUE').DEBG1ØØ
         MVC   WTOR_LD(WTORLEN),WTOR_LS
         LA    R4,REPLY
         LA    R5,WTOR_ECB
         XC    WTOR_ECB,WTOR_ECB
         WTOR  ,(R4),1,(R5),MF=(E,WTOR_LD)
         WAIT  ECB=WTOR_ECB
.DEBG1ØØ ANOP
         L     R3,RETC_ADDRESS
         ST    R2,Ø(,R3)
         L     R8,RETADDR
         STORAGE RELEASE,LENGTH=DYNDATL,ADDR=(R11),COND=YES
         BR    R8                   RC IN R15
*********************************************************************
*
*  DEFINE PROGRAM CONSTANTS
*
*********************************************************************
WORKSIZE DC    AL4(DYNDATL)         SIZE OF STORAGE TO OBTAIN
TEXT1    DC    Y(DALSYSOU)          DEFINE TEXT UNIT FOR SYSOUT
         DC    X'ØØØØ'              DEFAULT CLASS
TEXT2    DC    Y(DALSPGNM)          DEFINE TEXT UNIT FOR INTERNAL RDR
         DC    X'ØØØ1' ONE SUBPARM
         DC    X'ØØØ8'              LENGTH OF INTERNAL READER NAME
         DC    C'INTRDR  '
TEXT3    DC    Y(DALDDNAM)          DEFINE INTRDR DDNAME TEXT UNIT
         DC    X'ØØØ1'
         DC    X'ØØØ8'
         DC    C'INTRDR  '
         AIF   ('&DEBUG' NE 'TRUE').DEBG2ØØ
WTOR_LS  WTOR  'CSGSUBMA Y/N',,1,,MF=L
WTORLEN  EQU   *-WTOR_LS
.DEBG2ØØ ANOP
         LTORG
DYNDAT   DSECT
RBLEN    EQU   (S99RBEND-S99RB)
```

```
RBP      DS    F
RB       DS    5F
TXTUNITS DS    ØFL3
TXTUNIT1 DS    F
TXTUNIT2 DS    F
TXTUNIT3 DS    F
         AIF   ('&DEBUG' NE 'TRUE').DEBG3ØØ
REPLY    DS    C
WTOR_ECB DS    F
WTOR_LD  WTOR  'CSGSUBMA Y/N',REPLY,1,WTOR_ECB,MF=L
.DEBG3ØØ ANOP
FLAG     DS    X
INTRDR   EQU   X'8Ø'                INTERNAL READER ALLOCATED IN JCL
RETADDR  DS    F
ACEEADDR DS    F                    ACEE ADDRESS RETURNED BY RACROUTE
JCLADDR  DS    F                    ADDRESS OF JCL CARDS
JCLLEN   DS    F                    LENGTH OF JCL CARDS  (BYTES)
JOBNAME_ADDRESS DS F
RETC_ADDRESS    DS F
CARD     DS    CL8Ø
DYNDATL  EQU   *-DYNDAT
         PRINT NOGEN
         YREGS                      DEFINE REGISTER EQUATES
         IEFZB4DØ
         IEFZB4D2
         IEFTIOT1
         IHAPSA DSECT=YES,LIST=NO
         IKJTCB DSECT=YES,LIST=NO
         IHAACEE
         IFGRPL DSECT=YES
         CVT    DSECT=YES
         END   CSGSUBMA
```

SVC CODE

This SVC checks a FACILITY class profile CICSSVC237 to verify the authority of a CICS region to set the TCBSENV field to the ACEE of the end-user.

Register 1 contains the address of the end-user's ACEE to set TCBSENV or binary zeros to reset TCBSENV to return to the security environment of the CICS region user-id:

```
* PROGRAM NAME    : IGCØØ23G                                          *
* TRANSACTION NAME: CALLED VIA MVS ATTACHED TCB FROM CSGSUBMA         *
* RDO             : NOT REQUIRED
* MAPS            : NONE
* RESIDENT MODE   : 31                                                *
```

```
* ADDRESS MODE    : ANY                                                *
* BINDER ATTRIBS  : RENT REUS                                          *
* AUTHORIZATION   : SVC                                                *
*                                                                      *
* FUNCTION        : THIS PROGRAM PROVIDES AN APF AUTHORIZED FACILITY   *
*                   TO CICS. IT CHECKS A PROFILE, FACILITY CLASS       *
*                   'CICSSVC237' FOR READ ACCESS. IF READ ACCESS       *
*                   IS NOT GRANTED THEN A RC(12) IS RETURNED AND NO    *
*                   INSERTION OF THE USERS ACEE INTO THE TCB IS DONE.  *
*                                                                      *
*                   IF THIS PROFILE CHECK RETURNS ACCESS=READ OR       *
*                   HIGHER THEN THE USERS ACEE IS INSERTED INTO THE    *
*                   TCBSENV FIELD TO ENABLE CICS TO SUBMIT JCL TO THE  *
*                   INTERNAL READER ON BEHALF OF THE USER WITHOUT      *
*                   REQUIRING THE USER-ID AND PASSWORD OF THE USER IN  *
*                   JOBCARD.                                           *
*                                                                      *
* WRITTEN BY      : CICS SUPPORT GROUP - WILFRED KAZOKS                *
*                                                                      *
* INPUT PARMS     : R1 ADDRESS OF USERS ACEE INSERTED INTO TCBSENV     *
*                   R1 SET TO ZEROS TO RETURN TCBSENV TO ZERO
* RETURN CODES    : RETURN CODE IS PASSED BACK VIA R15                 *
*                                                                      *
*                   RC=Ø OK                                            *
*                   RC=12 SVC237 NOT ALLOWED HERE                      *
*                                                                      *
************************************************************************
         TITLE     'IGCØØ23G - ENABLE CICS JCL SUBMISSION'
IGCØØ23G AMODE 31
IGCØØ23G RMODE ANY
IGCØØ23G CSECT
         USING IGCØØ23G,R6
         B     AROUND
         DC    AL1(L'EYEBALL)
EYEBALL  DC    C'IGCØØ23G &SYSDATE &SYSTIME'
AROUND   DS    ØH
         LR    R12,R6              ESTABLISH PROGRAM BASE REG SVC
         USING IGCØØ23G,R12
         LR    R8,R14
         LR    R7,R1               SAVE INPUT PARM ADDR IN R7
         STORAGE OBTAIN,LENGTH=DYNDATL,COND=YES
         LR    R11,R1              ESTABLISH DYNAMIC VARIABLE BASE
         USING DYNDAT,R11
         ST    R8,RETADDR
         LTR   R7,R7               R7=@ACEE OR Ø TO RESET TCBSENV
         BZ    LBLØ1ØØ
         MVI   ENTITY,C' '         BLANKOUT VARIABLE 'ENTITY'
         MVC   ENTITY+1(L'ENTITY),ENTITY
         MVC   ENTITY(L'SVCAUTH),SVCAUTH  CHECK SVC IS PERMITTED HERE
         TRT   ENTITY,SCANTBL      LOOKING FOR A SPACE
```

```
         LA    RØ,ENTITY
         SR    R1,RØ               R1 IS ADDR OF SPACE
         STCM  R1,B'ØØØ1',ENTITYL
         LA    RØ,ENTITYL
         ST    RØ,PARMLIST         FIRST WORD OF PARMLIST
         SR    RØ,RØ
         LA    RØ,CLFACL           1 BYTE LEN MUST PRECEDE CLASS NAME
         ST    RØ,PARMLIST+4       SAVE ADDRESS OF 2ND PARM
         LA    R1,PARMLIST
         BAL   R14,LBL2ØØØ         DO RACROUTE REQ=AUTH.
         LTR   R15,R15             EXPECT R15=Ø
         BNZ   EXIT12
         BZ    LBLØ11Ø             GO SET TCBSENV TO USERS ACEE
*------------------------------------------------------------------- *
* AT THIS POINT WE ARE AUTHORIZED TO UPDATE THE TCBSENV FIELD.       *
*------------------------------------------------------------------- *
LBLØ1ØØ  DS    ØH
LBLØ11Ø  DS    ØH
         ST    R7,TCBSENV-TCB(R4) STORE ACEE ADDRESS
         SR    R2,R2              SET ZERO RC
         B     LBL9999
EXIT12   DS    ØH
         LA    R2,12
LBL9999  DS    ØH
         L     R8,RETADDR
         STORAGE RELEASE,LENGTH=DYNDATL,ADDR=(R11),COND=YES
         LR    R15,R2
         BR    R8                  RC IN R15
*********************************************************************
* THIS ROUTINE PERFORMS A RACROUTE REQUEST=AUTH.
*
* THE PARM LIST FOR THIS ROUTINE IS A TABLE OF 2 FULLWORDS POINTED TO
* BY R1
*
* R1: -> @ OF STRUCTURE OF 1 BYTE LEN FIELD FOLLOWED BY ENTITY NAME
*          @ OF 1 BYTE LENGTH FOLLOWED BY CLASS NAME, EG 'FACILITY'
*
*********************************************************************
LBL2ØØØ  DS    ØH
         STM   R2,R14,LBL2ØØØS
         L     R9,4(,R1)           GET CLASS ADDRESS
*
* CLEAR 512 BYTES STORAGE FOR SAF/RACF WORKAREA
*
         XC    SAFWORK(L'SAFWORK/2),SAFWORK
         XC    SAFWORK+L'SAFWORK/2(L'SAFWORK/2),SAFWORK+L'SAFWORK/2
         MVC   PROFLEN1,PROFSIZE   PRIME ENTITYX BUFFER LENGTH
         XC    PROFLEN2,PROFLEN2   ZERO PROFILE LENGTH FIELD
         MVI   PROFBUF,C' '        BLANK OUT PROFILE BUFFER
         MVC   PROFBUF+1(L'PROFBUF-1),PROFBUF
```

```
         MVC   AUTHCHK(RACRAL),RACRAS INIT RACROUTE MACRO
         L     R2,Ø(,R1)           GET ENTITY LENGTH ADDRESS
         SR    R1Ø,R1Ø
         ICM   R1Ø,B'ØØØ1',Ø(R2)   ENTITY LENGTH IN R1Ø
         STH   R1Ø,PROFLEN2
         BCTR  R1Ø,Ø               DECR R1Ø FOR EX INSTRUCTION
         LA    R2,1(,R2)           R2 POINTS TO ENTITY TO CHECK
         LA    R1,PROFBUF
         EX    R1Ø,LBL2Ø1Ø
         B     LBL21ØØ
LBL2Ø1Ø  MVC   Ø(Ø,R1),Ø(R2)       MOVE THE ENTITY NAME
LBL21ØØ  DS    ØH
         RACROUTE REQUEST=AUTH,    RACHECK REQUEST.
               RELEASE=1.9.2,                                          X
               CLASS=(R9),                                             X
               ATTR=READ,                                              X
               ENTITYX=PROFLEN1,                                       X
               WORKA=SAFWORK,      SAF WORK AREA.                      X
               MF=(E,AUTHCHK)
         LA    R2,AUTHCHK
         USING SAFP,R2
         L     RØ,SAFPRRET
         L     R1,SAFPRREA
         STCM  R15,B'ØØØ1',SRC     SAVE SAF RETURN CODE
         STCM  RØ,B'ØØØ1',RRC      SAVE RACF RETURN CODE
         STCM  R1,B'ØØØ1',RRSNC    SAVE RACF REASON CODE
         DROP  R2
         LM    R2,R14,LBL2ØØØS
         BR    R14
         EJECT
         DS    ØD
RACRAS   RACROUTE REQUEST=AUTH,WORKA=*-*,RELEASE=1.9.2,MF=L
RACRAL   EQU   *-RACRAS
CLFACL   DC    AL1(CLFACSZ)
CLFACLTY DC    C'FACILITY'
CLFACSZ  EQU   *-CLFACLTY
         DS    ØF
WORKSIZE DC    AL4(DYNDATL)        SIZE OF STORAGE TO OBTAIN
PROFSIZE DC    AL2(PROFBLEN)       LENGTH OF PROFILE BUFFER
SVCAUTH  DC    C'CICSSVC237'       PROFILE NAME
SCANTBL  DC    256X'ØØ'    ONLY INTERESTED IN MATCHING SPACES
         ORG   SCANTBL+C' '
         DC    X'FF'
         ORG
         LTORG
DYNDAT   DSECT
LBL2ØØØS DS    18F
RETADDR  DS    F
SRC      DS    X                   SAF RETURN CODE SAVEAREA
RRC      DS    X                   RACF RETURN CODE SAVE AREA
```

```
RRSNC    DS    X                    RACF REASON CODE SAVE AREA
SAFWORK  DS    CL512                SAF WORK AREA.
PARMLIST DS    2F
         DS    ØD
AUTHCHK  RACROUTE REQUEST=AUTH,WORKA=*-*,RELEASE=1.9.2,MF=L
PROFLEN1 DS    H                    LENGTH OF PROFILE BUFFER
PROFLEN2 DS    H                    LENGTH OF PROFILE (OR ZERO)
PROFBUF  DS    CL255                PROFILE BUFFER
PROFBLEN EQU   *-PROFBUF
ENTITYL  DS    X
ENTITY   DS    CL39                 STAGING BUFFER FOR ENTITY BUILD
DYNDATL  EQU   *-DYNDAT
         PRINT NOGEN
         YREGS                      DEFINE REGISTER EQUATES
         ICHSAFP
         IKJTCB                     TCB
         END   IGC0ØØ23G
/*
```

Wilfred Kazoks
CICS System Programmer
IBM Global Services (Australia)

Quick log-off from CICS

A colleague recently made the following request, having just converted from ACF/2 to RACF. ACF/2 allowed a quick log-off and disconnect from CICS by typing the transaction LOGO (as in LOGOFF). While the same function can be performed by typing the transaction CESF LOGOFF, users still wanted to be able to use the shortened command.

As an exercise in using some of the new features introduced in CICS Version 4.1, as well as an exercise in using the SPI, the following programs were quickly developed. The programs are associated with the transaction LOGO, but could just as well be associated with a PF key to speed log-off.

The first example uses the RETURN IMMEDIATE feature to process the request, and demonstrates how a message may be passed to the

program so that the program receiving control treats the input message as if it were typed at the terminal.

```
         PRINT NOGEN
DFHEISTG DSECT
MSG      DS    CL11 ,                AREA TO HOLD MESSAGE
LEN      DS    H ,                   LENGTH OF MESSAGE
LOGO     CSECT
LOGO     AMODE 31
LOGO     RMODE ANY
         MVC   MSG,=C'CESF LOGOFF'   MOVE LOGOFF TEXT TO MSG
         MVC   LEN,=AL2(L'MSG)       SET LENGTH OF MESSAGE
         EXEC CICS RETURN TRANSID(=C'CESF')                          X
               IMMEDIATE INPUTMSG(MSG) INPUTMSGLEN(LEN)
         END    LOGO
```

The second program uses the SIGNOFF command and the SPI command INQUIRE TERMINAL, together with the CICS API command ISSUE DISCONNECT, to obtain the same results.

Note: because this program was written as an exercise rather than as a production application, error handling is less than complete.

```
         PRINT NOGEN
DFHEISTG DSECT
DISC     DS    F
MSG      DS    CL18
LOGO     CSECT
LOGO     AMODE 31
LOGO     RMODE ANY
         EXEC CICS INQUIRE TERMINAL(EIBTRMID) DISCREQST(DISC)
         CLC   DFHVALUE(NODISCREQ),DISC
         BE    SEND
         EXEC CICS SIGNOFF NOHANDLE
         EXEC CICS ISSUE DISCONNECT
         B     RETURN
SEND     DS    0H
         MVC   MSG,=C'LOGOFF NOT ALLOWED'
         EXEC CICS SEND FROM(MSG) LENGTH(=H'18')
RETURN   DS    0H
         EXEC CICS RETURN
         END   LOGO
```

Donald H Blake
Systems Programmer
Key Services (USA)

Analysing abended transactions

This article 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.

At the time of an abend, the program DFHPEP writes the following information about the abended task into file CICSAB:

- Transaction-id
- Date of abend
- Time of abend
- Current abend code
- Original abend code
- Abended program-id
- Task number
- Terminal-id
- Start code
- EIB
- PSW at abend
- Execution key
- Storage hit
- Registers.

If required, the transaction ABND can display the information and read file DFHCMACD for more help about the abend code.

```
ABND
************************************************************************
*                                                                      *
*      TASK ABEND                                                      *
*                                                                      *
```

```
*                      INFORMATION                                *
*                                                                 *
*******************************************************************
ABND     DFHCOVER
*==================================================================
* <<< SOURCE CODE PROGRAM ABND >>>
* NOTE:
*    1) THIS PROGRAM LINKS TO DERCODE TO DECODE ANY EXEC CICS ERROR
*       CONDITION
*    2) THIS PROGRAM LINKS TO GETCMAC TO DECODE CICS ABEND FROM
*       DFHCMACD FILE
*    3) THIS PROGRAM READS FILE 'CICSAB' WRITTEN BY PROGRAM 'DFHPEP'
*
*==================================================================
         PRINT NOGEN
         TITLE  'MACRO DEFINITIONS'
         MACRO                     MACRO HEADER
         PGMID &MEMBER,&R=         PROTYPE STATEMENT
         AGO   .PGNAME
.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
*
*
*
*        PROTOTYPE STATEMENT
         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
         DFHCOVER
DFHEISTG DSECT                 DEFINE DYNAMIC STORAGE
*********************************
* TRANSACTION CODE:  ABND      *
*********************************
*
* DERCODE Commarea
*
DEERRØAI DS    ØH
ERFUNCOD DS    CL2  FUNCTION CODE
ERERRCOD DS    CL6  ERROR CODE
ERRESNAM DS    CL8  RESOURCE NAME
ERTDQNAM DS    CL4  TD NAME
*              CL4'XXXX'   TD QUEUE NAME SPECIFIED BY CALLER
*              X'ØØØØØØØØ' DEFAULT TD QUEUE (CSMT)
*              CL4' '      DEFAULT TD QUEUE (CSMT)
*              X'FF'       DO NOT SEND MSG TO TD QUEUE
ERPGMCAL DS    CL8  CALLING PROGRAM
ERMSGS   DS    CL36 ERROR MSG
```

```
*
DEERRØAF EQU   *
         ORG   DEERRØAI
DEERRØAG DS    CL(DEERRØAF-DEERRØAI)
DEERRØAL EQU   L'DEERRØAG
*
* END of DERCODE COMMAREA
*
DOUBLE   DS    D       WORKAREA
         DS    D       WORKAREA
YEAR     DS    F       CENTURY FROM CICS
CRESP    DS    F       CICS RESPONSE
SAVE1415 DS    2A      SAVE REG.14/15
VOXBAL1  DS    A       FIRST LEVEL ROUTINE
VOXBAL2  DS    A       SECOND LEVEL ROUTINE
TABEND   DS    F       ABENDS
GETCMACT DS    F       ABEND CODE FOR HELP
CENTURY  DS    CL2     CENTURY ZONED
DATEW    DS    CL8     YYYYMMDD DATE
TIMEW    DS    CL8     HHMMSS..
FILENAME DS    CL8     FILENAME
NETNAMEW DS    CL8     NETNAME
APPLIDW  DS    CL8     VTAM APPLID
USER     DS    CL8     USER ID
LEN      DS    H       LENGTH
STC      DS    CL2     STARTCODE
TIOA     DS    CL2Ø    YYYYMMDD,HHMMSS,TTTT
*
* COMMAREA passed by itself
*
WCOMM    DS    ØCL42
WDATE    DS    CL8     YYYYMMDD
WTIME    DS    CL6     HHMMSS
WTIMES   DS    CL6     HHMMSS Time selected by operator
WTRANID  DS    CL4     TRANSACTION-ID
WTRANIDS DS    CL4     TRANSACTION-ID selected by operator
STRACB   DS    CL2     EQ/NE
WCABC    DS    CL4     CURRENT ABEND CODE
WCABCS   DS    CL4     ABEND CODE selected by operator
ABCCB    DS    CL2     EQ/NE
WFUN     DS    CL1     FUNCTION (Ø - 1 - 2)
         DS    CL1     FREE
*
* End of COMMAREA passed by itself
*
SWF      DS    CL1     F = AT LEAST 1 RECORD FOUND
SKEY     DS    CL34    SAVE KEY
*
* FILE CICSAB WORK AREA
*
```

```
         COPY  TACBREC
*
* MAPSET LAYOUT
*
         COPY  MAPTACBD
*
         TITLE '*** EQUATES ***'
*
         COPY  DFHAID
*
         COPY  DFHBMSCA
*
*****************************************************************
*                                                               *
*              REGISTER EQUATES                                 *
*                                                               *
*****************************************************************
         SPACE 5
*
***
*
         SPACE
R0       EQU   0
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
         SPACE
RWKR1    EQU   R1  WORK REGISTER
RWKR2    EQU   R2  WORK REGISTER
RWKR3    EQU   R3  WORK REGISTER
RWKR14   EQU   R14 WORK REGISTER
RWKR15   EQU   R15 WORK REGISTER
RBAL1    EQU   R1  1ST LEVEL ROUTINES
RBAL2    EQU   R2  2ND LEVEL ROUTINES
         SPACE
CODEREG1 EQU   R4  CODE REGISTER 1
CODEREG2 EQU   R5  CODE REGISTER 2
DATAREG1 EQU   R10 DATA REGISTER 1
```

```
EIBREG   EQU   R12 EIB BASE REGISTER
RCOMM    EQU   R9  COMMAREA BASE REGISTER
         SPACE
*
***
*
         TITLE '>>>>>>>>>>> ABND CONTROL SECTION <<<<<<<<<<<'
*
*
ABND     DFHEIENT CODEREG=(CODEREG1,CODEREG2),EIBREG=(EIBREG),         *
               DATAREG=(DATAREG1)
         SPACE
ABND     AMODE ANY
ABND     RMODE ANY
         SPACE
         B     APGMID
         PGMID ABND,R=ØØØ1
APGMID   DS    ØH
         SPACE
*
* Perform initial functions
*
         BAS   RBAL1,KICKOFF
*
         SPACE
         CLC   =H'Ø',EIBCALEN          No COMMAREA
         BE    FIRST                   ...Yes, First Entry
         CLC   =Y(L'WCOMM),EIBCALEN    Wrong COMMAREA Length ?
         BNE   FIRST                   ...Yes, force First Entry
*
         L     RCOMM,DFHEICAP          Get COMMAREA Address
         MVC   WCOMM,Ø(RCOMM)          Move in Working Storage
         CLI   WFUN,C'Ø'               Function Ø ?
         BE    FUNØ                    ...Yes
         CLI   WFUN,C'1'               Function 1 ?
         BE    FUN1                    ...Yes
         CLI   WFUN,C'2'               Function 2 ?
         BE    FUN2                    ...Yes
         B     FIRST                   ??? , force First Entry
FIRST    DS    ØH
*
* Send Map
*
         BAS   RBAL2,CLEARMPØ          Clear MapØ
*
         MVC   SDATEØO,DATEW           Move date in MapØ
         MVC   STIMEØO,HEXØ            Reset Time
         MVC   STRANØO,HEXØ            Reset Transaction-id
*
         MVC   ERRESNAM,=CL8'MAPTACB'  Move resource
```

```
*
         EXEC  CICS SEND MAP('MAPØ') MAPSET('MAPTACB') ERASE FREEKB
*
         XC    ERRESNAM,ERRESNAM        Reset resource
*
         MVI   WFUN,C'Ø'                Set Function Ø
         B     RETURNCM                 Return to CICS trans-id with
*                                       COMMAREA
         EJECT
FUNØ     DS    ØH
*
         BAS   RBAL1,CHECKINP           Check Terminal Input
*
         MVC   TACBREC_APPLID,APPLIDW   Set Vtam Applid
         MVC   TACBREC_DATE,WDATE       Set Date required
         MVC   TACBREC_TIME,WTIME       Set Time required
         MVC   TACBREC_TRX,WTRANID      Set Transaction-id
         XC    TACBREC_CABC,TACBREC_CABC Set Current Abend Code
         CLC   WCABC,HEXØ               Set Abend Code
         BE    WCABCNS                  ...No
         CLC   WCABC,BLANK              Set Abend Code
         BE    WCABCNS                  ...No
         MVC   TACBREC_CABC,WCABC       Set Abend Code
WCABCNS  DS    ØH
* Browse to count abends
         XC    TACBREC_TSKN,TACBREC_TSKN ReSet Task number
         XC    TABEND,TABEND            ReSet Abend counter
         MVC   SKEY,TACBREC_KEY         Save key for next browse
         MVI   SWF,C' '                 ReSet found record switch
*
         BAS   RBAL1,STARTBR            Begin file Browse
*
WCABCNS1 DS    ØH
*
         BAS   RBAL1,READN              Get Next Record
*
         CLC   CRESP,DFHRESP(NORMAL)    Normal response ?
         BNE   WCABCNSF                 ...No, terminate browse
*
         L     RWKR1,TABEND             add 1 in abend counter
         LA    RWKR1,1(RWKR1)           and
         ST    RWKR1,TABEND             continue browse
         B     WCABCNS1
WCABCNSF DS    ØH
*
         BAS   RBAL1,ENDBR              Terminate browse
*
         MVC   TACBREC_KEY,SKEY         Restore key
*
* Begin browse again to display abends
```

```
*
         BAS   RBAL1,STARTBR            Begin file Browse
*
FIRST2   DS    ØH
*
         BAS   RBAL2,CLEARMP1           Clear Map1
*
         LA    RWKR15,MAPSEL1O          Point to first element in map
         LA    RWKR14,5                 Number of elements in map
FIRSTL   DS    ØH
         STM   RWKR14,RWKR15,SAVE1415   Save registers 14 & 15
*
         BAS   RBAL1,READN              Get Record
*
         CLC   CRESP,DFHRESP(NOTFND)    Not Found ?
         BE    NOABE                    ...Yes
         MVI   SWF,C'F'                 Set found record
         LM    RWKR14,RWKR15,SAVE1415   Restore registers 14 & 15
LOOPMAP  DS    ØH
         MVI   MAPSEL1O,C'_'            Initialize field
         MVC   MAPTRX1O-MAPSEL1O(L'MAPTRX1O,RWKR15),TACBREC_TRX
         MVC   MAPDAT1O-MAPSEL1O(L'MAPDAT1O,RWKR15),TACBREC_DATE
         MVC   MAPTIM1O-MAPSEL1O(L'MAPTIM1O,RWKR15),TACBREC_TIME
         MVC   MAPABC1O-MAPSEL1O(L'MAPABC1O,RWKR15),TACBREC_CABC
         MVC   MAPABO1O-MAPSEL1O(L'MAPABO1O,RWKR15),TACB_COM_ORIGINAL_A/
               BEND_CODE
         MVC   MAPPGM1O-MAPSEL1O(L'MAPPGM1O,RWKR15),TACB_COM_ABPROGRAM
         LA    RWKR15,MAPSEL2O-MAPSEL1O(RWKR15)
         BCT   RWKR14,FIRSTL            Read Next Record
*
* Send map and continue
*
         BAS   RBAL1,SENDPAGE           Send Page Accum
*
         BAS   RBAL2,CLEARMP1           Clear Map1
*
         LA    RWKR15,MAPSEL1O          Point to first element in map
         LA    RWKR14,5                 Number of elements in map
         B     FIRSTL                   Continue Loop
         SPACE
ENDMAP   DS    ØH
         LM    RWKR14,RWKR15,SAVE1415   Restore Registers 14 & 15
         CH    RWKR14,=H'5'             At least 1 row set ?
         BE    ENDMAP1                  ...No
*
         BAS   RBAL1,SENDPAGE           Send Page Accum
*
ENDMAP1  DS    ØH
*
         BAS   RBAL1,ENDBR              Terminate browse
```

```
*
         BAS   RBAL1,ENDPAGE            Terminate Paging
*
         MVI   WFUN,C'1'                Set Function 1
         B     RETURNCM                 Return with commarea
         EJECT
FUN1     DS    ØH
         MVC   ERRESNAM,=CL8'MAPTACB'   Set Resource Name
*
         EXEC  CICS RECEIVE MAP('MAP1') MAPSET('MAPTACB')             *
               RESP(CRESP)
*
         CLI   EIBAID,DFHPA1            Exit
         BE    RETURNCL                 ...Yes, return and erase screen
         CLI   EIBAID,DFHCLEAR          Exit
         BE    RETURNCL                 ...Yes, return and erase screen
         CLC   CRESP,DFHRESP(MAPFAIL)   Map fail
         BE    STARTTX                  ...yes, Restart TAsk
         CLC   CRESP,DFHRESP(NORMAL)    OK
         BNE   ERROR                    ...No, go to Error Routine
         XC    ERRESNAM,ERRESNAM        Reset Resource name
         LA    RWKR1,MAPSEL1I
         LA    RWKR14,5                 Number of elements in map
LOOPSL1  DS    ØH
         CLI   Ø(RWKR1),C'_'            Field Not Selected
         BE    NEXTSL1                  ...No, Go to Next Field
         CLI   Ø(RWKR1),C' '            Field Not Selected
         BE    NEXTSL1                  ...No, Go to Next Field
         CLI   Ø(RWKR1),X'Ø'            Field Not Selected
         BE    NEXTSL1                  ...No, Go to Next Field
         B     SL1                      Field selected
NEXTSL1  DS    ØH
         LA    RWKR1,MAPSEL2I-MAPSEL1I(RWKR1)
         BCT   RWKR14,LOOPSL1           Go To next Abend Code
* No fields selected
         B     STARTTX                  Restart with Start trans-id
SL1      DS    ØH
* Build CICSAB Key
         MVC   TACBREC_APPLID,APPLIDW
         MVC   TACBREC_DATE,MAPDAT1I-MAPSEL1I(RWKR1)
         MVC   TACBREC_TIME,MAPTIM1I-MAPSEL1I(RWKR1)
         MVC   TACBREC_TRX,MAPTRX1I-MAPSEL1I(RWKR1)
         MVC   TACBREC_CABC,MAPABC1I-MAPSEL1I(RWKR1)
         ZAP   TACBREC_TSKN,=P'Ø'
         MVC   WDATE,TACBREC_DATE
         MVC   WTIME,TACBREC_TIME
         MVC   WTRANID,TACBREC_TRX
*
         BAS   RBAL1,STARTBR            Start Browse
*
```

```
         BAS   RBAL1,READN              Begin Browse
*
         CLC   CRESP,DFHRESP(NORMAL)    OK ?
         BNE   ERROR                    ...No, go to Error Routine
*
         BAS   RBAL1,ENDBR              End Browse
*
         BAS   RBAL2,CLEARMP2           Clear Map 2
*
* Initialize map from record
*
         MVC   MAP2TRXO,TACBREC_TRX
         MVC   MAP2TDTO,TACBREC_DATE
         MVC   MAP2TTMO,TACBREC_TIME
         UNPK  MAP2TKNO,TACBREC_TSKN
         OI    MAP2TKNO+L'MAP2TKNO-1,X'FØ'
*
* Initialize map from EIB
*
         LR    RWKR15,DFHEIBR           Save current EIB Address
         LA    DFHEIBR,TACB_COM_USERS_EIB Load Task abended EIB addr
         MVC   MAP2TRMO,EIBTRMID
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'EIBAID*2)+1),EIBAID(L'EIBAID+1)
         TR    DOUBLE((L'EIBAID*2)+1),TABEX
         MVC   MAP2AIDO,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'EIBCALEN*2)+1),EIBCALEN(L'EIBCALEN+1)
         TR    DOUBLE((L'EIBCALEN*2)+1),TABEX
         MVC   MAP2OMLO,DOUBLE
         MVC   MAP2RSRO,EIBRSRCE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'EIBFN*2)+1),EIBFN(L'EIBFN+1)
         TR    DOUBLE((L'EIBFN*2)+1),TABEX
         MVC   MAP2FNO,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'EIBRCODE*2)+1),EIBRCODE(L'EIBRCODE+1)
         TR    DOUBLE((L'EIBRCODE*2)+1),TABEX
         MVC   MAP2ECO,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'EIBERR*2)+1),EIBERR(L'EIBERR+1)
         TR    DOUBLE((L'EIBERR*2)+1),TABEX
         MVC   MAP2ERRO,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'EIBERRCD*2)+1),EIBERRCD(L'EIBERRCD+1)
         TR    DOUBLE((L'EIBERRCD*2)+1),TABEX
         MVC   MAP2ERCO,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'EIBRESP*2)+1),EIBRESP(L'EIBRESP+1)
         TR    DOUBLE((L'EIBRESP*2)+1),TABEX
```

```
         MVC   MAP2RSO,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'EIBRESP2*2)+1),EIBRESP2(L'EIBRESP2+1)
         TR    DOUBLE((L'EIBRESP2*2)+1),TABEX
         MVC   MAP2RS2O,DOUBLE
         LR    DFHEIBR,RWKR15           Reload Current EIB address
*
         MVC   MAP2PGMO,TACB_COM_ABPROGRAM
         MVC   MAP2STCO,TACBREC_STC
         MVC   MAP2ABCO,TACB_COM_CURRENT_ABEND_CODE
         MVC   MAP2ABOO,TACB_COM_ORIGINAL_ABEND_CODE
         XC    DOUBLE,DOUBLE
   UNPK  DOUBLE(L'TACB_COM_PSW+1),TACB_COM_PSW((L'TACB_COM_PSW/2)+1)
         TR    DOUBLE(L'TACB_COM_PSW+1),TABEX
         MVC   MAP2PSWO(L'MAP2PSWO/2),DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE(L'TACB_COM_PSW+1),TACB_COM_PSW+((L'TACB_COM_PSW/2*
               )+1)
         TR    DOUBLE(L'TACB_COM_PSW+1),TABEX
         MVC   MAP2PSWO+L'MAP2PSWO/2(L'MAP2PSWO/2),DOUBLE
         XC    DOUBLE,DOUBLE
   UNPK  DOUBLE((L'TACB_COM_KEY*2)+1),TACB_COM_KEY(L'TACB_COM_KEY+1)
         TR    DOUBLE((L'TACB_COM_KEY*2)+1),TABEX
         MVC   MAP2EXKO,=CL2'NA'
         CLC   =CL2'Ø8',DOUBLE
         BE    MVEXK
         CLC   =CL2'Ø9',DOUBLE
         BNE   AMVEXK
MVEXK    DS    ØH
         MVC   MAP2EXKO,DOUBLE
AMVEXK   DS    ØH
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE((L'TACB_COM_STORAGE_HIT*2)+1),TACB_COM_STORAGE_HI*
               T(L'TACB_COM_STORAGE_HIT+1)
         TR    DOUBLE((L'TACB_COM_STORAGE_HIT*2)+1),TABEX
         MVC   MAP2STHO,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE(9),TACB_COM_REGØ(5)
         TR    DOUBLE(9),TABEX
         MVC   MAP2RØO,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE(9),TACB_COM_REG1(5)
         TR    DOUBLE(9),TABEX
         MVC   MAP2R1O,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE(9),TACB_COM_REG2(5)
         TR    DOUBLE(9),TABEX
         MVC   MAP2R2O,DOUBLE
         XC    DOUBLE,DOUBLE
         UNPK  DOUBLE(9),TACB_COM_REG3(5)
```

```
TR    DOUBLE(9),TABEX
MVC   MAP2R3O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG4(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R4O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG5(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R5O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG6(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R6O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG7(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R7O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG8(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R8O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG9(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R9O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG1Ø(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R1ØO,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG11(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R11O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG12(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R12O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG13(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R13O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG14(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R14O,DOUBLE
XC    DOUBLE,DOUBLE
UNPK  DOUBLE(9),TACB_COM_REG15(5)
TR    DOUBLE(9),TABEX
MVC   MAP2R15O,DOUBLE
```

```
         MVC   MAP2ABNO,=CL4'____'
         MVC   MAP2NOTO(68),=CL68'<- -  Press Enter to Continue  - -  o*
               r set abcode for more help'
*
* Send Map
*
         MVC   ERRESNAM,=CL8'MAPTACB'   Set resource
*
         EXEC  CICS SEND MAP('MAP2') MAPSET('MAPTACB') ERASE FREEKB
*
         XC    ERRESNAM,ERRESNAM        Reset resource
         MVI   WFUN,C'2'                Set function 2
         B     RETURNCM
         EJECT
FUN2     DS    ØH
         MVC   ERRESNAM,=CL8'MAPTACB'   Set resource
*
         EXEC  CICS RECEIVE MAP('MAP2') MAPSET('MAPTACB') RESP(CRESP)
*
         XC    ERRESNAM,ERRESNAM        Reset resource
         CLC   CRESP,DFHRESP(NORMAL)    Response OK ?
         BNE   FUN2A                    ...No, ignore error and resend
         CLI   MAP2ABNI,C'_'
         BE    FUN2A
         CLI   MAP2ABNI,C' '
         BE    FUN2A
         CLI   MAP2ABNI,X'Ø'
         BE    FUN2A
         MVC   GETCMACT,MAP2ABNI
         MVC   ERRESNAM,=CL8'GETCMAC'
*
         EXEC  CICS LINK PROGRAM('GETCMAC') COMMAREA(GETCMACT)         *
               LENGTH(=Y(L'GETCMACT)) RESP(CRESP)
*
         XC    ERRESNAM,ERRESNAM
*
         EXEC  CICS PURGE MESSAGE
*
         SPACE
*
FUN2A    DS    ØH
         MVC   WTIME,WTIMES             Reset original time
         MVC   WTRANID,WTRANIDS         Reset original tran-id
         MVC   WCABC,WCABCS             Reset original abend code
*
         BAS   RBAL2,CLEARMP2           Clear Map 2
*
         MVC   TACBREC_APPLID,APPLIDW   Set Vtam Appl-id
         MVC   TACBREC_DATE,WDATE       Set Date required
         MVC   TACBREC_TIME,WTIME       Set Time required
```

```
         MVC   TACBREC_TRX,WTRANID      Set Transaction-id
         XC    TACBREC_CABC,TACBREC_CABC Set Current Abend Code
         XC    TACBREC_TSKN,TACBREC_TSKN Set Task number
*
         BAS   RBAL1,STARTBR            Begin file Browse
*
         XC    WTRANID,WTRANID          To select All records
         XC    WTIME,WTIME              To select All records
         B     FIRST2
         SPACE
*
* Restart Itself via start trans-id
*
STARTTX  DS    ØH
*
         EXEC  CICS START TRANSID('ABND') INTERVAL(Ø)                  *
               TERMID(EIBTRMID)
*
         B     RETURN
         SPACE
*
* Return to CICS with trans-id
*
RETURNID DS    ØH
*
         EXEC  CICS RETURN TRANSID('ABND')
*
         SPACE
*
* Return to CICS with trans-id and COMMAREA
*
RETURNCM DS    ØH
*
         EXEC  CICS RETURN TRANSID('ABND') COMMAREA(WCOMM)             *
               LENGTH(=Y(L'WCOMM))
*
         SPACE
*
* Return to CICS and erase the screen
*
RETURNCL DS    ØH
*
         EXEC  CICS SEND CONTROL ERASE FREEKB
*

         B     RETURN
         SPACE
*
* Return to CICS
*
```

```
RETURN   DS    ØH
*
         EXEC  CICS RETURN
*
NOABE    DS    ØH
         CLI   SWF,C'F'                 At least 1 record
         BE    ENDMAP                   ...Yes
*
         EXEC  CICS SEND FROM(MSG1) LENGTH(=Y(L'MSG1)) ERASE
*
         B     RETURN
         SPACE 5
KICKOFF  DS    ØH
         ST    RBAL1,VOXBAL1             Save Return address
         MVC   FILENAME,FILENAMD         SET FILENAME
*
* SET ERROR CONDITION
*
         EXEC  CICS HANDLE CONDITION ERROR(ERROR)
*
* SET AID BRANCH
*
         EXEC  CICS HANDLE AID PA1(RETURNCL) CLEAR(RETURNCL)
*
* GETS TIME
*
         EXEC  CICS ASKTIME ABSTIME(DOUBLE)
*
* FORMATS TIME
*
         EXEC  CICS FORMATTIME ABSTIME(DOUBLE) YEAR(YEAR) YYMMDD(DATEW)*
               TIME(TIMEW)
*
         MVC   DOUBLE,DATEW                  FORMAT DATE
         MVC   DATEW+2(6),DOUBLE
         L     RWKR1,YEAR                    FORMAT CENTURY
         CVD   RWKR1,DOUBLE           XXXXXXXXXXX1997+
         UNPK  DOUBLE(5),DOUBLE+5(3)
         MVC   CENTURY,DOUBLE+1
         MVC   DATEW(2),CENTURY
*
* GET APPLID - NETNAME - USERID - STARTCODE
*
         SPACE
*
         EXEC  CICS ASSIGN APPLID(APPLIDW) NETNAME(NETNAMEW)          *
               USERID(USER) STARTCODE(STC)
*
         L     RBAL1,VOXBAL1             Load Return Address
         BR    RBAL1
```

```
         SPACE 5
CHECKINP DS    ØH
         ST    RBAL1,VOXBAL1            Save Return Address
         CLC   =CL2'TD',STC             Terminal Facility ?
         BNE   FCHECK                   ...NO
         MVC   ERRESNAM,=CL8'MAPTACB'   Set Resource Name
*
         EXEC  CICS RECEIVE MAP('MAPØ') MAPSET('MAPTACB')             *
               RESP(CRESP)
*
         CLI   EIBAID,DFHPA1            Exit
         BE    RETURNCL                 ...Yes, return and erase screen
         CLI   EIBAID,DFHCLEAR          Exit
         BE    RETURNCL                 ...Yes, return and erase screen
         CLC   CRESP,DFHRESP(MAPFAIL)   Map fail
         BE    STARTTX                  ...yes, Restart Task
         CLC   CRESP,DFHRESP(NORMAL)    OK
         BNE   ERROR                    ...No, go to Error Routine
         XC    ERRESNAM,ERRESNAM        Reset Resource name
* SET DEFAULT VALUES
         MVC   WDATE,DATEW
         XC    WTIME,WTIME
         XC    WTIMES,WTIMES
         MVC   WTRANID,BLANK
         MVC   WTRANIDS,BLANK
         MVC   WCABC,BLANK
         MVC   WCABCS,BLANK
         MVC   STRACB,BLANK
         MVC   ABCCB,BLANK
*
         CLI   SDATEØI,C' '             Date set ?
         BE    CHECKTR                  ...No
         CLI   SDATEØI,X'Ø'             Date set ?
         BE    CHECKTR                  ...No
         CLI   SDATEØI,C'_'             Date set ?
         BE    CHECKTR                  ...No
         TRT   SDATEØI,TABNUM           Test Numeric
         BNZ   ERR1
         CLC   SDATEØI+4(2),=CL2'12'    Formal check
         BH    ERR1
         CLC   SDATEØI+4(2),=CL2'Ø1'    Formal check
         BL    ERR1
         CLC   SDATEØI+6(2),=CL2'31'    Formal check
         BH    ERR1
         CLC   SDATEØI+6(2),=CL2'Ø1'    Formal check
         BL    ERR1
         MVC   WDATE,SDATEØI
         CLI   STIMEØI,C' '             Time set ?
         BE    CHECKTR                  ...No
         CLI   STIMEØI,X'Ø'             Time set ?
```

```
         BE    CHECKTR                  ...No
         CLI   STIMEØI,C'_'             Time set ?
         BE    CHECKTR                  ...No
         CLC   STIMEØI,=6C'Ø'           Time set ?
         BE    CHECKTR                  ...No
         TRT   STIMEØI,TABNUM           Test Numeric
         BNZ   ERR1
         CLC   STIMEØI(2),=CL2'23'      Formal check
         BH    ERR1
         CLC   STIMEØI+2(2),=CL2'59'    Formal check
         BH    ERR1
         CLC   STIMEØI+4(2),=CL2'59'    Formal check
         BH    ERR1
         MVC   WTIME,STIMEØI
         MVC   WTIMES,STIMEØI
CHECKTR  DS    ØH
         CLI   STRANØI,C' '             Trans-id set ?
         BE    CHECKABC                 ...No
         CLI   STRANØI,X'Ø'             Trans-id set ?
         BE    CHECKABC                 ...No
         CLI   STRANØI,C'_'             Trans-id set ?
         BE    CHECKABC                 ...No
         MVC   WTRANID,STRANØI
         MVC   WTRANIDS,STRANØI
         MVC   STRACB,STRACØI
CHECKABC DS    ØH
         CLI   ABCØI,C' '               Trans-id set ?
         BE    FCHECK                   ...No
         CLI   ABCØI,X'Ø'               Trans-id set ?
         BE    FCHECK                   ...No
         CLI   ABCØI,C'_'               Trans-id set ?
         BE    FCHECK                   ...No
         MVC   WCABC,ABCØI
         MVC   WCABCS,ABCØI
         MVC   ABCCB,ABCCØI
FCHECK   DS    ØH
         L     RBAL1,VOXBAL1            Load Return Address
         BR    RBAL1
         SPACE 5
STARTBR  DS    ØH
         ST    RBAL1,VOXBAL1
         MVC   ERRESNAM,FILENAME
*
         EXEC  CICS STARTBR FILE(FILENAME) RIDFLD(TACBREC_KEY)         *
               GTEQ RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NOTFND)    Not Found ?
         BE    NOABE                    ...Yes
         CLC   CRESP,DFHRESP(NORMAL)
         BNE   ERROR
```

```
         XC    ERRESNAM,ERRESNAM
         L     RBAL1,VOXBAL1
         BR    RBAL1
ENDBR    DS    ØH
         ST    RBAL1,VOXBAL1
         MVC   ERRESNAM,FILENAME
*
         EXEC  CICS ENDBR FILE(FILENAME) NOHANDLE
*
         XC    ERRESNAM,ERRESNAM
         L     RBAL1,VOXBAL1
         BR    RBAL1
READN    DS    ØH
         ST    RBAL1,VOXBAL1
READN1   DS    ØH
         MVC   LEN,=Y(TACBRECL)
         MVC   ERRESNAM,FILENAME
*
         EXEC  CICS READNEXT FILE(FILENAME) RIDFLD(TACBREC_KEY)        *
               INTO(TACBREC) LENGTH(LEN) RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NORMAL)
         BE    OKREAD1
         CLC   CRESP,DFHRESP(ENDFILE)
         BE    OKREAD2
         CLC   CRESP,DFHRESP(NOTFND)
         BNE   ERROR
         CLI   TACBREC_KEY,X'FF'
         BE    OKREAD2
OKREAD   DS    ØH
         XC    ERRESNAM,ERRESNAM
         L     RBAL1,VOXBAL1
         BR    RBAL1
OKREAD1  DS    ØH
         CLI   TACBREC_KEY,X'FF'
         BE    OKREAD2
         CLC   TACBREC_APPLID,APPLIDW
         BNE   OKREAD2
         CLC   TACBREC_DATE,WDATE
         BNE   OKREAD2
         CLI   WTRANIDS,C' '
         BE    OKREAD1A
         CLI   WTRANIDS,X'Ø'
         BE    OKREAD1A
         CLC   =CL2'EQ',STRACB
         BE    TRAEQ
         CLC   STRACB,HEXØ
         BE    TRAEQ
         CLC   STRACB,BLANK
         BE    TRAEQ
```

```
         CLC   =CL2'NE',STRACB
         BNE   TRAEQ
         CLC   TACBREC_TRX,WTRANIDS
         BE    READN1
         B     OKREAD1A
TRAEQ    DS    ØH
         CLC   TACBREC_TRX,WTRANIDS
         BNE   READN1
OKREAD1A DS    ØH
         CLI   WCABCS,C' '
         BE    OKREAD1B
         CLI   WCABCS,X'Ø'
         BE    OKREAD1B
         CLC   =CL2'EQ',ABCCB
         BE    ABCEQ
         CLC   ABCCB,HEXØ
         BE    ABCEQ
         CLC   ABCCB,BLANK
         BE    ABCEQ
         CLC   =CL2'NE',ABCCB
         BNE   ABCEQ
         CLC   TACBREC_CABC,WCABCS
         BE    READN1
         B     OKREAD1B
ABCEQ    DS    ØH
         CLC   TACBREC_CABC,WCABCS
         BNE   READN1
OKREAD1B DS    ØH
         CLI   WTIME,C' '
         BE    OKREAD
         CLI   WTIME,X'Ø'
         BE    OKREAD
         CLC   WTIME,=6C'Ø'
         BE    OKREAD
         CLC   TACBREC_TIME,WTIME
         BL    READN1
         B     OKREAD
OKREAD2  DS    ØH
         MVC   CRESP,DFHRESP(NOTFND)
         B     OKREAD
         SPACE 5
CLEARMPØ DS    ØH
         ST    RBAL2,VOXBAL2
         LA    RWKR2,MAPØO
         LA    RWKR3,MAPØL
         LR    RWKR14,RWKR2
         SR    RWKR15,RWKR15
         MVCL  RWKR2,RWKR14
         MVC   NETNAMØO,NETNAMEW Set netname
         MVC   OPIDØO,USER         Set user-id
         MVC   DATEØO(2),DATEW+6  Day
```

```
         MVI   DATEØO+2,C'/'
         MVC   DATEØO+3(2),DATEW+4 Month
         MVI   DATEØO+5,C'/'
         MVC   DATEØO+6(4),DATEW  Year
         MVC   TIMEØO(2),TIMEW    Hours
         MVI   TIMEØO+2,C':'
         MVC   TIMEØO+3(2),TIMEW+2 Minutes
         MVC   APPLIDØO,APPLIDW   VTAM Appl-id
         L     RBAL2,VOXBAL2
         BR    RBAL2
         SPACE 5
CLEARMP1 DS    ØH
         ST    RBAL2,VOXBAL2
         LA    RWKR2,MAP1O
         LA    RWKR3,MAP1L
         LR    RWKR14,RWKR2
         SR    RWKR15,RWKR15
         MVCL  RWKR2,RWKR14
         MVC   NETNAMEO,NETNAMEW Set netname
         MVC   OPIDO,USER         Set user-id
         MVC   DATEO(2),DATEW+6  Day
         MVI   DATEO+2,C'/'
         MVC   DATEO+3(2),DATEW+4 Month
         MVI   DATEO+5,C'/'
         MVC   DATEO+6(4),DATEW  Year
         MVC   TIMEO(2),TIMEW    Hours
         MVI   TIMEO+2,C':'
         MVC   TIMEO+3(2),TIMEW+2 Minutes
         MVC   APPLIDO,APPLIDW   VTAM Appl-id
         L     RWKR1,TABEND
         CVD   RWKR1,DOUBLE
         UNPK  ABNDTOTO,DOUBLE+5(3)
         OI    ABNDTOTO+L'ABNDTOTO-1,X'FØ'
         L     RBAL2,VOXBAL2
         BR    RBAL2
         SPACE 5
CLEARMP2 DS    ØH
         ST    RBAL2,VOXBAL2
         LA    RWKR2,MAP2O
         LA    RWKR3,MAP2L
         LR    RWKR14,RWKR2
         SR    RWKR15,RWKR15
         MVCL  RWKR2,RWKR14
         MVC   MAP2NETO,NETNAMEW Set netname
         MVC   MAP2OPO,USER      Set user-id
         MVC   MAP2DATO(2),DATEW+6 Day
         MVI   MAP2DATO+2,C'/'
         MVC   MAP2DATO+3(2),DATEW+4 Month
         MVI   MAP2DATO+5,C'/'
         MVC   MAP2DATO+6(4),DATEW Year
         MVC   MAP2TIMO(2),TIMEW Hours
```

```
         MVI   MAP2TIMO+2,C':'
         MVC   MAP2TIMO+3(2),TIMEW+2 Minutes
         MVC   MAP2APPO,APPLIDW  VTAM Appl-id
         L     RBAL2,VOXBAL2
         BR    RBAL2
         SPACE 5
SENDPAGE DS    ØH
         ST    RBAL1,VOXBAL1
         MVC   ERRESNAM,=CL8'MAPTACB'
*
         EXEC  CICS SEND MAP('MAP1') MAPSET('MAPTACB') ERASE FREEKB    *
               PAGING ACCUM
*
         XC    ERRESNAM,ERRESNAM
         L     RBAL1,VOXBAL1
         BR    RBAL1
         SPACE 5
ENDPAGE  DS    ØH
         ST    RBAL1,VOXBAL1
*
         MVC   ERRESNAM,=CL8'MAPTACB'
*
         EXEC  CICS SEND MAP('MAP3') MAPSET('MAPTACB') ERASE FREEKB    *
               PAGING ACCUM
*
         XC    ERRESNAM,ERRESNAM
*
         EXEC  CICS SEND PAGE NOAUTOPAGE
*
         EXEC  CICS PURGE MESSAGE
*
         L     RBAL1,VOXBAL1
         BR    RBAL1
         SPACE 5
ERR1     DS    ØH
*
* Send Map
*
         BAS   RBAL2,CLEARMPØ           Clear MapØ
*
         MVC   SDATEØO,DATEW
         MVC   STIMEØO,HEXØ
         MVC   STRANØO,HEXØ
         MVC   MAPNOTØO,=CL79'                 <<<   Data Error    >>>'
*
         MVC   ERRESNAM,=CL8'MAPTACB'
*
         EXEC  CICS SEND MAP('MAPØ') MAPSET('MAPTACB') ERASE FREEKB
*
         XC    ERRESNAM,ERRESNAM
*
```

```
         MVI   WFUN,C'Ø'                Set Function
         B     RETURNCM
         SPACE 5
ERROR    DS    ØH
         MVC   ERFUNCOD,EIBFN
         MVC   ERERRCOD,EIBRCODE
         MVC   ERPGMCAL,CSNAME
*
         EXEC  CICS IGNORE CONDITION ERROR
*
         EXEC  CICS LINK PROGRAM('DERCODE') COMMAREA(DEERRØAI)         *
               LENGTH(=Y(DEERRØAL))
*
         EXEC  CICS SEND FROM(MSGE) LENGTH(=Y(L'MSGE)) ERASE
*
         B     RETURN
*
FILENAMD DC    CL8'CICSAB'
*
MSG1     DC    CL23'    No task abend found'
MSGE     DC    CL27'    Request not satisfiable'
*
TABNUM   DS    ØCL256
BLANK    DC    CL(C'Ø')' '
HEXØ     DC    1ØX'Ø'
         DC    CL(X'FF'-C'9')' '
*
TABEX    DC    256X'Ø'
         ORG   TABEX+X'FØ'
         DC    C'Ø123456789ABCDEF'
         ORG
*
         CSNAME
*
         END   ABND
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
*
* COPY TACBREC
*
*
* FILE CICSAB - RECORD LAYOUT -
*
TACBREC  DS    ØF
*
*************************************************************************
*
TACBREC_KEY  DS    ØXL34
TACBREC_APPLID DS  CL8    CICS APPL-ID
TACBREC_DATE DS    CL8    DATE YYYYMMDD
TACBREC_TIME DS    CL6    TIME HHMMSS
TACBREC_TRX  DS    CL4    TRANSACTION-ID
```

```
TACBREC_CABC DS    CL4    CURRENT ABEND CODE
TACBREC_TSKN DS    PL4    TASK NUMBER
*
*********************************************************************
*
TACBREC_STC  DS    CL2    STARTCODE
*
*********************************************************************
*
*
*                            Standard header section
*
TACB_COM_STANDARD            DS     ØF
TACB_COM_FUNCTION            DS     CL1       always '1'
TACB_COM_COMPONENT           DS     CL2       always 'PC'
TACB_COM_RESERVED            DS     C         ...reserved...
*
*                            Abend codes and EIB
*
TACB_COM_CURRENT_ABEND_CODE  DS     CL4       current abend code
TACB_COM_ORIGINAL_ABEND_CODE DS     CL4       original abend code
TACB_COM_USERS_EIB           DS     CL(EIBRLDBK-EIBTIME+L'EIBRLDBK)   *
EIB at abend
*
*                            Debugging information (program, PSW,
*                            registers and execution key at time of
*                            abend, hit storage indicator.)
*
TACB_COM_DEBUG               DS     ØF
TACB_COM_ABPROGRAM           DS     CL8       ABENDING PROGRAM
TACB_COM_PSW                 DS     CL8       PSW at abend
TACB_COM_REGISTERS           DS     ØXL64     registers at abend
TACB_COM_REGØ                DS     XL4       register Ø
TACB_COM_REG1                DS     XL4       register 1
TACB_COM_REG2                DS     XL4       register 2
TACB_COM_REG3                DS     XL4       register 3
TACB_COM_REG4                DS     XL4       register 4
TACB_COM_REG5                DS     XL4       register 5
TACB_COM_REG6                DS     XL4       register 6
TACB_COM_REG7                DS     XL4       register 7
TACB_COM_REG8                DS     XL4       register 8
TACB_COM_REG9                DS     XL4       register 9
TACB_COM_REG1Ø               DS     XL4       register 1Ø
TACB_COM_REG11               DS     XL4       register 11
TACB_COM_REG12               DS     XL4       register 12
TACB_COM_REG13               DS     XL4       register 13
TACB_COM_REG14               DS     XL4       register 14
TACB_COM_REG15               DS     XL4       register 15
TACB_COM_KEY                 DS     X         execution key at abend  *
                                              in form x'Øn' (ASRA     *
                                              AND ASRB ONLY)
```

```
*
TACB_COM_USER_KEY             EQU    9         USER KEY
TACB_COM_CICS_KEY             EQU    8         CICS KEY
*
TACB_COM_STORAGE_HIT          DS     X         storage type hit by ØC4 *
                                               (ASRA ONLY)
*
TACB_COM_NO_HIT               EQU    Ø         NO HIT OR NOT ØC4
TACB_COM_CDSA_HIT             EQU    1         CDSA HIT
TACB_COM_ECDSA_HIT            EQU    2         ECDSA HIT
TACB_COM_ERDSA_HIT            EQU    3         ERDSA HIT
*
TACB_COM_PADDING              DS     CL2       RESERVED
*
*                             Return code
*
TACB_COM_RETURN_CODE          DS     F
*
TACB_COM_RETURN_OK            EQU    Ø
TACB_COM_RETURN_DISABLE       EQU    4         Disable transaction
*
*                             length of DFHPEP_COMMAREA
*
TACB_COM_LEN EQU *-TACB_COM_STANDARD
*
*****************************************************************
*
TACBRECL EQU   *-TACBREC
*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
*
* MAPSET MAPTACB Type=Map
*
         TITLE 'BMS: MAPTACB                    '
         PRINT ON,NOGEN
MAPTACB  DFHMSD TYPE=MAP,LANG=ASM,MODE=INOUT,STORAGE=AUTO,SUFFIX=M
         TITLE 'BMS: MAPTACB  MAPØ              '
MAPØ     DFHMDI SIZE=(24,8Ø),CTRL=(FREEKB,ALARM),MAPATTS=(COLOR,HILIGHT*
               ,SOSI),DSATTS=(COLOR,HILIGHT,SOSI),COLUMN=1,LINE=1,     *
               DATA=FIELD,TIOAPFX=YES,OBFMT=NO
         DFHMDF POS=(1,1),LENGTH=79,                                   *
               INITIAL='+————————————————————*
               ————————————+',ATTRB=(PROT,BRT),     *
               COLOR=NEUTRAL
         DFHMDF POS=(2,1),LENGTH=1,INITIAL='+',ATTRB=(PROT,BRT),       *
               COLOR=NEUTRAL
```

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

Giuseppe Rallo
Senior Technical Analyst
Sicilcassa (Italy)

An enhancement to PINQPGM

Chorng S (Jack) Hwang's article *Determining the library using PINQPGM* was published in *CICS Update*, Issue 152, July 1998.

I have written a small enhancement that will translate lower-case letters to upper case.

```
*****************************************************************
*   WRITTEN BY CHORNG S (JACK) HWANG                            *
*   TITLE : PINQPGM ISSUE 152 CICS UPDATE FROM JULY 1998        *
*****************************************************************
*
.
.
.
*
BLDLAREA DS    CL2Ø
REGSTORE DS    16F
MVSREGSA DS    18F
RSTORE59 DS    5F                  <= NEW
*
.
.
.
.
         OC    PGMNAM,=CL8' '      CLEAR PGMNAM
         EXEC  CICS RECEIVE SET(1Ø) LENGTH(TEXTLEN)
*
*                    NEW CODE
*
TRANS    EQU   *                   PREPARE TRANSLATE
         STM   5,9,RSTORE59        BE CAREFULLY, SAFE REGISTERS
         XR    5,5                 CLEAR R5
         XR    6,6                 CLEAR R6
         LH    5,TEXTLEN           LOAD SLIP
         LA    7,TABØ1
         LA    8,TRANS1
         LR    9,1Ø                LOAD INPUT
TRANS1   EQU   *                   TRANSLATE
         IC    6,Ø(Ø,9)            CHARACTER FROM INPUT
         IC    6,Ø(6,7)            TRANSLATE CHARACTER
         STC   6,Ø(Ø,9)            RETURN TO INPUT
         LA    9,1(Ø,9)            LOAD ADR NEXT
         BCTR  5,8                 TRANSLATE NEXT OR END
         LM    5,9,RSTORE59        RELOAD REGISTERS - FINISHED
*
```

```
*              END OF NEW CODE
*
         CLC   Ø(4,1Ø),EIBTRNID     IS THIS UNFORMATTED?
         BE    SENDINIT             YES, GO SEND INITIAL
.
.
.
CONCATSO DS    CL4
DSNAMEL  EQU   *-DSNAMES
*
         DS    ØF              <== NEW TAB
TABØ1    EQU   *               TRANSLATE FROM LOWER TO UPPER
         DC    X'4Ø'           TRANSLATE X'ØØ' TO X'4Ø'
         DC    127AL1(*-TABØ1)
         DC    X'8ØC1C2C3C4C5C6C7C8C98A8B8C8D8E8F'    A - I
         DC    X'9ØD1D2D3D4D5D6D7D8D99A9B9C9D9E9F'    J - R
         DC    X'AØA1E2E3E4E5E6E7E8E9AAABACADAEAF'    S - Z
         DC    8ØAL1(*-TABØ1)
.
.
.
         END
```

J Hoffmann
Systems Programmer (Germany)

Using a transaction across several CICSs

We have had CICS in our shop for a long time and have several CICSs for several different applications. For a long time, users have asked to be able to switch from one CICS to another without a log-off and a log-on operation.

The solution is to use the ISC facility of CICS; however, before doing this you must be certain that your security policy is OK.

CICS ISC

```
****** in csd of cics A355SRSR
DEFINE CONNECTION(TEST) GROUP(REDLU62)
       NETNAME(A356SRSR) ACCESSMETHOD(VTAM) PROTOCOL(APPC)
       SINGLESESS(NO) DATASTREAM(USER) RECORDFORMAT(U) QUEUELIMIT(NO)
```

```
       MAXQTIME(NO) AUTOCONNECT(YES) INSERVICE(YES) ATTACHSEC(LOCAL)
       BINDSECURITY(NO) USEDFLTUSER(YES) PSRECOVERY(SYSDEFAULT)
DEFINE SESSIONS(TESTSESS) GROUP(REDLU62)
       CONNECTION(TEST) PROTOCOL(APPC) MAXIMUM(1Ø,5) SENDSIZE(4Ø96)
       RECEIVESIZE(4Ø96) SESSPRIORITY(Ø) AUTOCONNECT(YES)
       BUILDCHAIN(YES) USERAREALEN(Ø) IOAREALEN(Ø,Ø) RELREQ(NO)
       DISCREQ(NO) NEPCLASS(Ø) RECOVOPTION(SYSDEFAULT)
****** A METTRE DANS CICS A356SRSR
DEFINE CONNECTION(TES6) GROUP(REDLU62)
       NETNAME(A355SRSR) ACCESSMETHOD(VTAM) PROTOCOL(APPC)
       SINGLESESS(NO) DATASTREAM(USER) RECORDFORMAT(U) QUEUELIMIT(NO)
       MAXQTIME(NO) AUTOCONNECT(YES) INSERVICE(YES) ATTACHSEC(LOCAL)
       BINDSECURITY(NO) USEDFLTUSER(YES) PSRECOVERY(SYSDEFAULT)
DEFINE SESSIONS(TES6SESS) GROUP(REDLU62)
       CONNECTION(TES6) PROTOCOL(APPC) MAXIMUM(1Ø,5) SENDSIZE(4Ø96)
       RECEIVESIZE(4Ø96) SESSPRIORITY(Ø) AUTOCONNECT(YES)
       BUILDCHAIN(YES) USERAREALEN(Ø) IOAREALEN(Ø,Ø) RELREQ(NO)
       DISCREQ(NO) NEPCLASS(Ø) RECOVOPTION(SYSDEFAULT)
****    VTAM list
APPLCICS VBUILD TYPE=APPL
A356SRSR APPL  AUTH=(ACQ,VPACE,BLOCK,PASS),                            *
               EAS=2Ø,                                                 *
               VPACING=1,                                              *
               PARSESS=YES,                                            *
               SONSCIP=YES,                                            *
               APPC=NO
****   MODIFIER VTAMLST AINSI
APPLCICS VBUILD TYPE=APPL
A355SRSR APPL  AUTH=(ACQ,VPACE,BLOCK,PASS),                            *
               EAS=2Ø,                                                 *
               VPACING=1,                                              *
               PARSESS=YES,                                            *
               SONSCIP=YES,                                            *
               APPC=NO
******  CECI READ FILE(VVVVVV) with SYSID(TEST OU TES6)
       checks with CEMT I C LES CONNECTIONS
                     ET   I MOD
```

Claude Dunand (France)

CICS Update is looking for JCL, macros, program code, etc, that experienced CICS users have written to make their life, or the lives of their users, easier. Articles can be of any length and can be sent or e-mailed to Robert Burgess at any of the addresses shown on page 2. Why not call now for a free copy of our *Notes for contributors*?

CICS news

CICS users can benefit from new Domino Connectors for leading transaction processing systems, announced by Lotus, that allow native integration of transactional data in Domino applications.

New Connectors include modules for linking into CICS, TXSeries, and IMS, plus BEA Tuxedo, all through either DECS or LEI. The Connectors can also be accessed programmatically through Domino Classes using LotusScript or Java, supplied with LEI. The Connectors enable two-way data integration between Domino and transaction processing systems such as CICS.

For further information contact:
Lotus Development, 55 Cambridge Parkway, Cambridge, MA 02142-1295, USA.
Tel: (617) 577 8500.
Lotus Development Ltd (UK), Lotus Park, The Causeway, Staines, Middx, TW18 3AG.
Tel: (01784) 455445.
URL: http://www.lotus.com.

* * *

Lincoln Software has released a new version of its repository-based application development tool, Engineer for CICS, which can now generate a full Java application. Users can now choose whether to develop in HTML or Java, using the CICS Web Interface or CICS Gateway for Java, and switch from one to the other. Hence HTML can be used today and applications can be moved to Java in the future. The product models the application first and then generates the code in whichever language is preferred.

Engineer generates Java for the user interface and all the necessary code to link back to the CICS system. It also works with CICS on NT or Unix platforms.

For further information contact:
Lincoln Software Ltd, Marlborough Court, Pickford Street, Macclesfield, Cheshire, SK11 6JD, UK.
Tel: (01625) 616722.
URL: http://www.ipsys.com.

* * *

CICS users can now benefit from BMC's announcement of the first products in its Enterprise Data Propagation management strategy for integrating applications and enabling near real-time data warehousing.

ChangeDataMove provides transaction-based change capture for CICS, DB2 for OS/390, IMS, VSAM, and VSAM batch applications with near real-time or scheduled propagation of the captured changes to DB2 for OS/390 and to Oracle on Unix. The product is designed to support operational applications executing hundreds of transactions per second.

For further information contact:
BMC Software, 2101 CityWest Boulevard, Houston, TX 77042-2827, USA.
Tel: (713) 918 8800.
BMC Software, Compass House, 207-215 London Road, Camberley, Surrey, GU15 3EY, UK.
Tel: (01276) 24622.
URL: http://www.bmc.com.

* * *