176 CICS update

July 2000

In this issue

CICS Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38342
From USA: 01144 1635 38342
E-mail: trevore@xephon.com

North American office

Xephon
PO Box 350100
Westminster, CO 80035-0100
USA
Telephone: 303 410 9344

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.

CICS Update on-line

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

Editor

Trevor Eddolls

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.

Contributions

Articles published in *CICS Update* are paid for at the rate of £170 ($260) per 1000 words and £100 ($160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 ($80) per 100 lines. In addition, there is a flat fee of £30 ($50) per article. 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*, or you can download a copy from www.xephon.com/contnote.html.

Printed in England.

Stopping people bringing up CICS regions and leaving them

Here's a little program I use to stop people bringing up CICS regions and leaving them there all day without using them. You add it to the PLT so that it runs at start-up, then restarts itself every hour (you can set the interval yourself). It then checks the EIBTASKN, compares it to what it was an hour ago, and, if the minimum number of tasks have not been run, it shuts down the region. I wrote it for CICS 4.1 but it seems to work for CICS TS 1.3 as well.

```
*ASM XOPTS(PROLOG,NOEPILOG,SP)
         TITLE 'BWATCHDG - TECHNICAL SERVICES TRXN ACTIVITY WATCH DOG'
*********************************************************************
* BWATCHDG:                                                         *
* --------                                                          *
*                                                                   *
* DESCRIPTION:                                                      *
*   - TECHNICAL SERVICES CICS PLTPI (PHASE 2) PROGRAM               *
*   - INVOKED DURING THE THIRD STAGE OF CICS INITIALIZATION.        *
*   - THIS PROGRAM THEN RE-INVOKES ITSELF AT 60 MINUTE INTERVALS    *
*                                                                   *
* ATTRIBUTES:                                                       *
*   - HIGH LEVEL ASSEMBLER                                          *
*   - COMMAND LEVEL CICS/ESA V4.1.0                                 *
*   - CICS TRANSLATOR OPTION "SP" IS REQUIRED                       *
*   - EXECKEY(CICS)                                                 *
*   - TASKDATAKEY(CICS)                                             *
*   - 31 BIT ADDRESSING                                             *
*                                                                   *
* PROCESSING:                                                       *
*   - GET CURRENT TASK NUMBER.                                      *
*   - COMPARE CURRENT TASK NUMBER TO TASK NUMBER OBTAINED BY THE    *
*     PREVIOUS INVOCATION OF THIS PROGRAM.                          *
*   - IF NUMBER OF TASKS SINCE PREVIOUS INVOCATION OF THIS PROGRAM  *
*     IS LESS THAN 8...                                             *
*         THEN KICK OFF A NOTE TO ALL RELEVANT PEOPLE               *
*     ELSE...                                                       *
*         SAVE CURRENT TASK NUMBER IN A TSQ AND RESCHEDULE THIS     *
*         PROGRAM TO BE STARTED AGAIN IN 60 MINUTES TIME.           *
*                                                                   *
*********************************************************************
         TITLE 'BWATCHDG - REGISTER EQUATES'
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
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         TITLE 'BWATCHDG - CICS PROLOG CODE / ADDRESSING MODE'
BWATCHDG CSECT
BWATCHDG AMODE 31
BWATCHDG RMODE ANY
         TITLE 'BWATCHDG - MAINLINE CODE'
*********************************************************************
* BRANCH PAST PROGRAM NAME "EYECATCHER".                            *
*********************************************************************
MAIN     DS    ØH
         B     MAIN1ØØ
         DC    CL2'<<'
         DC    CL1' '
         DC    CL39'BRISTOL AND WEST BUILDING SOCIETY (TSG)'
         DC    CL1' '
         DC    CL8'&SYSDATE'
         DC    CL1' '
         DC    CL5'&SYSTIME'
         DC    CL1' '
         DC    CL45'BWATCHDG - WATCH DOG FOR TRANSACTION ACTIVITY'
         DC    CL1' '
         DC    CL2'>>'
*********************************************************************
* MAINLINE CODE                                                     *
*********************************************************************
MAIN1ØØ  DS    ØH
*                                     GET TASK NUMBER SAVED BY THE
*                                     PREVIOUS INVOCATION OF THIS
*                                     PROGRAM.
*
         EXEC CICS ASSIGN APPLID(APPLID)
         LA    R5,APP_TAB
CHK_APP  DS    ØH
         CLC   Ø(8,R5),APPLID
         BE    SKIPPROC
         CLC   Ø(8,R5),=C'ZZZZZZZZ'
         BE    CONTINUE
```

```
         LA    R5,8(R5)
         B     CHK_APP
CONTINUE DS    ØH
         WTO  'BWATCHDG - AUTO SHUTDOWN IS ACTIVE IN THIS REGION'
         EXEC CICS ASKTIME ABSTIME(ABSTIME)
         EXEC CICS FORMATTIME ABSTIME(ABSTIME)                          X
               DATE(DISP_DAT) DATESEP('/')                              X
               TIME(DISP_TIM) TIMESEP(':')
         MVC  DISP_TXT,MSG1
         EXEC CICS WRITEQ TD                                            X
               QUEUE('CSML')                                            X
               FROM(DISP_MSG)                                           X
               LENGTH(L'DISP_MSG)
*
         EXEC CICS CANCEL TRANSID(TRNID)                                X
               REQID(APPLID)                                            X
               RESP(RESP)
*
         EXEC CICS INQUIRE SYSTEM                                       X
               CICSSTATUS(CICSSTAT)                                     X
               RESP(RESP)
         CLC  CICSSTAT,DFHVALUE(STARTUP)
         BE   MAIN8ØØ
*
         EXEC CICS READQ TS                                             X
               QUEUE('WATCHDOG')                                        X
               ITEM(1)                                                  X
               INTO(TSQ_REC)                                            X
               RESP(RESP).
*
         CLC RESP,DFHRESP(NORMAL)
         BNE MAIN8ØØ
*
         UNPK TASK_#(4),TSQ_REC
         MVZ  TASK_#+3(1),=X'FØ'
         MVC  MSG4+22(4),TASK_#
         EXEC CICS ASKTIME ABSTIME(ABSTIME)
         EXEC CICS FORMATTIME ABSTIME(ABSTIME)                          X
               DATE(DISP_DAT) DATESEP('/')                              X
               TIME(DISP_TIM) TIMESEP(':')
         MVC  DISP_TXT,MSG4
         EXEC CICS WRITEQ TD                                            X
               QUEUE('CSML')                                            X
               FROM(DISP_MSG)                                           X
               LENGTH(L'DISP_MSG)
*
         ZAP  CURR_#,EIBTASKN
         ZAP  PREV_#,TSQ_REC
         ZAP  COMP_#,CURR_#
         SP   COMP_#,PREV_#
```

```
         CP   COMP_#,=P'3'
         BNH  SHUTDOWN
MAIN2ØØ  ZAP   EIBT_#,EIBTASKN
*
         EXEC CICS DELETEQ TS                                          X
               QUEUE('WATCHDOG')                                       X
               RESP(RESP).
*
         B     MAIN82Ø
*
MAIN8ØØ  DS   ØH
*
         EXEC CICS DELETEQ TS                                          X
               QUEUE('WATCHDOG')                                       X
               RESP(RESP).
*
         ZAP   EIBT_#,=P'1'
MAIN82Ø  DS   ØH
         UNPK TASK_#(4),EIBT_#
         MVZ  TASK_#+3(1),=X'FØ'
         MVC  MSG5+22(4),TASK_#
         EXEC CICS ASKTIME ABSTIME(ABSTIME)
         EXEC CICS FORMATTIME ABSTIME(ABSTIME)                         X
               DATE(DISP_DAT) DATESEP('/')                             X
               TIME(DISP_TIM) TIMESEP(':')
         MVC  DISP_TXT,MSG5
         EXEC CICS WRITEQ TD                                           X
               QUEUE('CSML')                                           X
               FROM(DISP_MSG)                                          X
               LENGTH(L'DISP_MSG)
*
         EXEC CICS WRITEQ TS                                           X
               QUEUE('WATCHDOG')                                       X
               FROM(EIBT_#)                                            X
               LENGTH(L'EIBT_#)                                        X
               RESP(RESP).
*
         CLC RESP,DFHRESP(NORMAL)
         BNE RETURN
*
MAIN9ØØ  DS   ØH
*
         CLC  APPLID,=C'TESTCIC2'
         BE   SHORT
*
         EXEC CICS START                                               X
               AFTER MINUTES(6Ø)                                       X
               TRANSID(TRNID)                                          X
               REQID(APPLID)                                           X
               RESP(RESP).
```

```
         B    RETURN
SHORT    DS   ØH
*
         EXEC CICS START                                          X
               AFTER MINUTES(6Ø)                                  X
               TRANSID(TRNID)                                     X
               REQID(APPLID)                                      X
               RESP(RESP).
         B    RETURN
*
SKIPPROC DS   ØH
         WTO  'BWATCHDG - AUTO SHUTDOWN NOT ACTIVE IN THIS REGION'
*
RETURN   DS   ØH
*
         EXEC CICS ASKTIME ABSTIME(ABSTIME)
         EXEC CICS FORMATTIME ABSTIME(ABSTIME)                    X
               DATE(DISP_DAT) DATESEP('/')                        X
               TIME(DISP_TIM) TIMESEP(':')
         MVC  DISP_TXT,MSG2
         EXEC CICS WRITEQ TD                                      X
               QUEUE('CSML')                                      X
               FROM(DISP_MSG)                                     X
               LENGTH(L'DISP_MSG)
*
         EXEC CICS RETURN
*
SHUTDOWN DS   ØH
*
         ZAP  EIBT_#,EIBTASKN
         UNPK TASK_#(4),EIBT_#
         MVZ  TASK_#+3(1),=X'FØ'
         MVC  MSG5+22(4),TASK_#
         EXEC CICS ASKTIME ABSTIME(ABSTIME)
         EXEC CICS FORMATTIME ABSTIME(ABSTIME)                    X
               DATE(DISP_DAT) DATESEP('/')                        X
               TIME(DISP_TIM) TIMESEP(':')
         MVC  DISP_TXT,MSG5
         EXEC CICS WRITEQ TD                                      X
               QUEUE('CSML')                                      X
               FROM(DISP_MSG)                                     X
               LENGTH(L'DISP_MSG)
         MVC  MSG3+11(8),APPLID
         EXEC CICS ASKTIME ABSTIME(ABSTIME)
         EXEC CICS FORMATTIME ABSTIME(ABSTIME)                    X
               DATE(DISP_DAT) DATESEP('/')                        X
               TIME(DISP_TIM) TIMESEP(':')
         MVC  DISP_TXT,MSG3
         EXEC CICS WRITEQ TD                                      X
               QUEUE('CSML')                                      X
```

```
               FROM(DISP_MSG)                                          X
               LENGTH(L'DISP_MSG)
*
*                                      --------------
          MVC   COMMAREA(L'REQUEST),REQUEST
*
          EXEC CICS XCTL   PROGRAM('DFHEMTP')                          X
                     COMMAREA(REQUEST) LENGTH(MAX_LEN)
*
          EXEC CICS RETURN
*
*
          B    MAIN2ØØ
*
          TITLE 'BWATCHDG - LITERAL POOL'
          LTORG
MSG1      DC    CL8Ø'BWATCHØ1 - PROGRAM START'
MSG2      DC    CL8Ø'BWATCHØ2 - PROGRAM END - CICS CONTINUES'
MSG3      DC    CL8Ø'BWATCHØ3 - XXXXXXXX CICS NOT USED - SHUTDOWN'
MSG4      DC    CL8Ø'BWATCHØ4 - EIBTASKN = XXXX (READ FROM QUEUE)'
MSG5      DC    CL8Ø'BWATCHØ5 - EIBTASKN = XXXX (READ FROM EIB)'
REQUEST   DC    C'CEMT P SHUT'
MAX_LEN   DC    H'11'
TRNID     DC    CL4'BWAT'
APP_TAB   DC    CL8'A1CICTD1'
*         DC    CL8'A1CICTC6'
*         DC    CL8'A1CICTCA'
*         DC    CL8'A1CICTK2'
          DC    CL8'ZZZZZZZZ'
          TITLE 'BWATCHDG - DFHEISTG DSECT'
DFHEISTG DSECT
DISP_MSG DS    ØCL1ØØ
DISP_DAT DS    CL8
DISP_FIL DS    CL1
DISP_TIM DS    CL8
DISP_FI1 DS    CL1
DISP_TXT DS    CL8Ø
ABSTIME  DS    PL8
TSQ_REC  DS    PL4
CURR_#   DS    PL4
PREV_#   DS    PL4
COMP_#   DS    PL4
TASK_#   DS    CL4
EIBT_#   DS    PL4
EIB_LEN  DS    H
EIBT_D   DS    D
EIBT_F   DS    F
RESP     DS    F
RESP2    DS    F
CICSSTAT DS    F
```

```
APPLID   DS    CL8
COMMAREA DS    CL11
         END
```

Graham Clark
Senior Systems Programmer
Bristol & West (UK)

A simple interface to DFHCSDUP to manage a CICS CSD resource

It is possible to invoke DFHCSDUP from a user program. This method enables you to create a flexible interface to the utility.

There is a CSD back-up utility program to produce a file of DFHCSDUP DEFINE control statements. The program is DFH0CBDC and it is written in VS COBOL II. It produces a sequential file containing all the resource definitions in a CSD file.

You can use the output from the utility:

- For later editing and commenting to document CSD resources.
- For distribution, in part or as a whole, to other CICS installations.
- To recreate or add resource definitions to any CSD using DFHCSDUP.
- To build or rebuild CSD definitions.
- As a sample back-up copy of the CSD VSAM file.

The program DFH0CBDC must be run against an EXTRACT command using the following formats:

```
EXTRACT GROUP(group name) OBJECTS USERPROGRAM(DFH0CBDC)
```

or:

```
EXTRACT LIST(list name) OBJECTS USERPROGRAM(DFH0CBDC)
```

In the file produced by DFH0CBDC, any DEFINE statements that

relate to CICS-supplied resources are commented out (there is an asterisk in column 1).

This is important if you use the file as input to define resources to a CSD because the CICS-supplied definitions are already present in the CSD, having been produced automatically when it was initialized.

The DEFINE statements for user CICS resources aren't preceded by an asterisk in column 1.

When an extract is carried out for a CICS list, we obtain a sequential file that contains all the definitions of the CICS list. This can make the management of single groups more complex.

For a simpler way to manage single groups, once the execution of the DFH0CBDC user program to extract the CICS list is ended, we can process the output sequential file by subdividing the CICS list into groups.

In order to provide this functionality, I have developed the CICSCSD CLIST. The CLIST subdivides the CICS list into CICS groups. It has been developed using the REXX language.

It reads the sequential file produced from the DFH0CBDC user program and produces a partitioned file.

With the execution of CICSCSD CLIST, we obtain, therefore, a partitioned file that contains a member for every group in the CICS list, and a member (NEWLIST) with the control statements to define a new CICS list with DFHCSDUP.

This makes the management of individual CICS groups much simpler. It makes it easier to use the CICS resource definitions in the CSD file.

This utility was developed and tested in the following environment: OS/390 1.3 and OS/390 2.7, CICS/ESA 4.1.0, and CICS TS 1.3.

COMPILE AND LINK-EDITING THE EXTRACT USER PROGRAM

You must compile and link-edit the DFH0CBDC user programs as batch programs, not as CICS applications.

When you link-edit the programs, you must specify the following link-edit control statements:

- An ENTRY statement that defines the entry name as DFHEXTRA. DFHEXTRA is the entry name in the CICS supplied stub, DFHEXCI.
- An INCLUDE statement for a CICS-supplied stub that must be included in your user program. Include DFHEXCI in any COBOL language user program that you write for use with the DFHCSDUP EXTRACT command. DFHEXCI is the interface stub between DFHCSDUP and the COBOL user program.
- Specify the COBOL routines on the INCLUDE statements.
- A CHANGE statement to change the dummy CSECT name in the CICS-supplied stub from EXITEP to the name of your user program. The CICS supplied stub, DFHEXCI, is generated with a link to the user program using a dummy CSECT name (EXITEP). Use the link-edit CHANGE statement to change the CSECT name from EXITEP to the name specified on the PROGRAM-ID statement in the user program. The CSD user program DFH0CBDC uses the program-id BDEFCSD.
- You must use AMODE(24) and RMODE(24).

SAMPLE JOB TO COMPILE AND LINK-EDIT THE USER PROGRAM

```
//ZZNS611M JOB (ZZNS0000),
//         CLASS=S,
//         MSGCLASS=X,
//         MSGLEVEL=(1,1),
//         NOTIFY=&SYSUID
//*
//*-------------------------------------------------------------*/
//*--------   Step to compile COBOL II  user program     ---------*/
//*-------------------------------------------------------------*/
//*
//COB2    EXEC PGM=IGYCRCTL,
//   PARM='APOST,LIB,OFFSET,MAP,NOSEQ,BUFSIZE(30K),NORENT,NORES'
//*
//STEPLIB  DD DISP=SHR,DSN=COBII.COB2COMP
//SYSIN    DD DISP=SHR,DSN=ZZNS611.LIB.SOURCE(DFH0CBDC)
//SYSLIB   DD DISP=SHR,DSN=ZZNS611.LIB.SOURCE
```

```
//SYSLIN   DD DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=VIO,
//         SPACE=(800,(1000,1000))
//SYSUDUMP DD SYSOUT=*
//SYSUT1   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//*
//MODOBJ   EXEC PGM=IEBGENER
//SYSUT1    DD DSN=&&LOADSET,DISP=(OLD,PASS)
//SYSUT2    DD DSN=CICS.LIBRARY.USER.OBJ(DFHØCBDC),DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//*
//*-------------------------------------------------------------*/
//*--------   Step to link-edit COBOL II  user program    --------*/
//*-------------------------------------------------------------*/
//*
//LKED    EXEC PGM=IEWL,
//   PARM='LIST,XREF,LET,AMODE(24),RMODE(24)'
//*
//SYSLIB   DD DISP=SHR,DSN=COBII.COB2LIB
//CICSLIB  DD DISP=SHR,DSN=CICS.SDFHLOAD
//OBJLIB   DD DISP=SHR,DSN=CICS.LIBRARY.USER.OBJ
//SYSLMOD  DD DISP=SHR,DSN=CICS.LIBRARY.USER.LOAD
//SYSUT1   DD SPACE=(1024,(50,50)),UNIT=VIO
//SYSPRINT DD SYSOUT=*
//COBLIB   DD DISP=SHR,DSN=COBII.COB2LIB
//SYSUDUMP DD SYSOUT=*
//SYSLIN   DD *
 ENTRY     DFHEXTRA
 CHANGE    EXITEP(BDEFCSD)
 INCLUDE   CICSLIB(DFHEXCI)
 INCLUDE   SYSLIB(ILBOSRV)
 INCLUDE   SYSLIB(ILBOCMM)
 INCLUDE   SYSLIB(ILBOBEG)
 INCLUDE   OBJLIB(DFHØCBDC)
 NAME      DFHØCBDC(R)
/*
/*
//
```

PROGRAM DFH0CBDC

```
*****************************************************************
*                                                               *
* MODULE NAME = DFHØCBDC                                        *
*                                                               *
*  Sample program to produce a CSD listing from EXTRACT         *
*                                                               *
*****************************************************************
*                                                               *
```

```
* Descriptions.                                                  *
*                                                               *
*---------------------------------------------------------------*
*                                                               *
* LANGUAGE: COBOL II                                            *
*                                                               *
* BDEFCSD takes input from the batch EXTRACT command and        *
* produces a listing of part or all of the CSD that can be used *
* to build or rebuild CSD definitions, or just as a summary     *
* listing of the CSD. The listing can be used as a means of     *
* passing CSD definitions from one CSD to another in a readable,*
* commentable form.                                             *
* The output varies according to the EXTRACT command being      *
* processed as follows:                                         *
*                                                               *
* EXTRACT GROUP( .. ):                                          *
* For every resource in the specified group, a partial DEFINE   *
* command is generated, eg:                                     *
*              DEFINE resource(XXXXXXX) GROUP(YYYYYYYY)         *
* Note that this command is not acceptable input to the batch   *
* utility. However, it does provide an editable list of the     *
* resources in each group.                                      *
*                                                               *
* EXTRACT GROUP( .. ) OBJECTS:                                  *
* For every resource in the specified group, a DEFINE command is*
* generated, eg:                                                *
*              DEFINE PROGRAM(XXXXXXXX) GROUP(YYYYYYYY)         *
*              DESCRIPTION( ... )                               *
*              LANGUAGE(COBOL) RELOAD(NO) RESIDENT( ...         *
*                                                               *
* This is acceptable input to the batch utility.                *
*                                                               *
* EXTRACT LIST:                                                 *
* For every occurrence of a group in the specified list an ADD  *
* command is generated, eg:                                     *
*              ADD GROUP(XXXXXXXX) LIST(YYYYYYYY)               *
*                                                               *
* EXTRACT LIST OBJECTS:                                         *
* As above, plus the equivalent of EXTRACT GROUP( .. ) OBJECTS  *
* for every group in the specified list.                        *
*                                                               *
* Notes:                                                        *
*       ADD and DEFINE commands and DESCRIPTION keywords begin  *
*       in column 1. All other keywords are listed starting in  *
*       column 8. If commented out, these values are increased  *
*       by 1.                                                   *
*                                                               *
*       An ADD command featuring a list beginning 'DFH' is      *
*       prefixed by a DFHCSDUP comment character, '*'.          *
*                                                               *
```

```
*       A DEFINE command featuring a group beginning 'DFH' is    *
*       prefixed by a DFHCSDUP comment character, '*'.           *
*                                                                *
*       Any resource keyword containing a null value is not      *
*       output because it is not valid input to DEFINE.          *
*                                                                *
*       Any resource keyword and its value (other than           *
*       DESCRIPTION) that will not fit into columns 8-71 is      *
*       output over multiple lines in a format acceptable as     *
*       input to DFHCSDUP.                                       *
*                                                                *
*                                                                *
* Inputs:  Data from DFHCULIS                                    *
* Outputs: Data file (see DATA-OUT) or the following return      *
* code:                                                          *
*                                                                *
*      Ø2: Invalid call type supplied to program                 *
*                                                                *
*----------------------------------------------------------------*
*                                                                *
* OUTPUT FILE                                                    *
* DEFINE and ADD commands will be written to this file.          *
*                                                                *
* Note that NO error processing is performed for this file,      *
* eg open/close errors.                                          *
*                                                                *
******************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. BDEFCSD.
 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT DATA-OUT
            ASSIGN TO SYSØØ2-DA-338Ø-S-CBDOUT
            ORGANIZATION IS SEQUENTIAL
            ACCESS IS SEQUENTIAL.
 EJECT.
 DATA DIVISION.
 FILE SECTION.
 FD  DATA-OUT
     RECORD CONTAINS 8Ø CHARACTERS
     BLOCK CONTAINS Ø RECORDS
     RECORDING MODE IS F
     LABEL RECORDS ARE OMITTED.
 Ø1  WRITE-BUFFER PIC X(8Ø).
 EJECT.
 WORKING-STORAGE SECTION.
*
* Call types, as defined by DFHCSDUP.
*
```

```
 77  INITIAL-CALL                  PIC 99 VALUE IS Ø.
 77  LIST-START-CALL               PIC 99 VALUE IS 2.
 77  GROUP-START-CALL              PIC 99 VALUE IS 4.
 77  OBJECT-START-CALL             PIC 99 VALUE IS 6.
 77  KEYWORD-CALL                  PIC 99 VALUE IS 8.
 77  OBJECT-END-CALL               PIC 99 VALUE IS 1Ø.
 77  GROUP-END-CALL                PIC 99 VALUE IS 12.
 77  LIST-END-CALL                 PIC 99 VALUE IS 14.
 77  FINAL-CALL                    PIC 99 VALUE IS 16.
*
*  Current list and group being processed.
*
 Ø1  CURRENT-LIST-ID.
     Ø2 LIST-PREFIX                PIC X(3).
     Ø2 LIST-SUFFIX                PIC X(5).
 Ø1  CURRENT-GROUP-ID.
     Ø2 GROUP-PREFIX               PIC X(3).
     Ø2 GROUP-SUFFIX               PIC X(5).
*
* LONG KEYWORD VALUE HANDLING
*
 Ø1  LINE-LENGTH                   PIC S9(4) COMP.
 Ø1  LENGTH-LINE-1                 PIC S9(4) VALUE IS 49.
 Ø1  CHAR-COUNT                    PIC S9(4) COMP.
 Ø1  LAST-CHAR                     PIC X.
 Ø1  TEMP-BUFFER                   PIC X(71).
 Ø1  TEMP-BUFFER-PTR               PIC S9(4) COMP.
 Ø1  CREATE-LINE-END               PIC X(1).
*
 Ø1  INDENTED-LINE-LEN             PIC S9(4) VALUE IS 62.
*
 77  ASTERISK                      PIC X VALUE IS '*'.
 77  FLUSH-PREFIX                  PIC X(3).
* Return code, if an invalid call is made to this program
 77  INVALID-CALL-TYPE             PIC S9(4) COMP VALUE IS 2.
*
* For buffering data whilst it is being formatted.
*
 77  BUFFER                        PIC X(73).
 77  BUFFER-PTR                    PIC S9(4) COMP.
 77  NUMBER-OF-SPACES              PIC S9(4) COMP.
 77  KEYWORD-TYPE-LENGTH           PIC S9(4) COMP.
 77  KEYWORD-LENGTH                PIC S9(4) COMP.
 77  BUFFER-SPACE                  PIC S9(4) COMP.
 77  MAX-TYPE-LENGTH               PIC S9(4) COMP VALUE IS 12.
 77  BUFFER-INDENT                 PIC S9(4) COMP VALUE IS 8.
 77  PARENTHESIS-LENGTH            PIC S9(4) COMP VALUE IS 2.
 77  KEYWORD-SCAN-PTR              PIC S9(4) COMP.
 77  CHARACTER-COUNT               PIC S9(4) COMP.
 77  CURRENT-LINE-LENGTH           PIC S9(4) COMP.
```

```
* ************************************************************* *
* Define the linkage between this program and the CSD off-line   *
* utility program. The addressability to the values addressed by *
* the parameter list passed from DFHCSDUP is established         *
* automatically by the COBOL compiler so all we need to define   *
* here are the actual formats of the values themselves.          *
* ************************************************************* *
 LINKAGE SECTION.
 Ø1  EXIT-FUNCTION-CODE             PIC 99 COMP.
 Ø1  EXIT-WORK-AREA-PTR             POINTER.
 Ø1  EXIT-BACK-TRANS-CMD-PTR        POINTER.
 Ø1  EXIT-LIST-NAME                 PIC X(8).
 Ø1  EXIT-GROUP-NAME                PIC X(8).
 Ø1  EXIT-OBJECT-TYPE               PIC X(12).
 Ø1  EXIT-OBJECT-NAME               PIC X(8).
 Ø1  EXIT-KEYWORD-TYPE              PIC X(12).
 Ø1  EXIT-KEYWORD-LENGTH            PIC 999 COMP.
 Ø1  EXIT-KEYWORD-VALUE.
     Ø3 EXIT-KEYWORD-CHAR           PIC X OCCURS 1 TO 183
                             DEPENDING ON EXIT-KEYWORD-LENGTH.
 EJECT.
* ************************************************************* *
*                                                               *
*          M A I N L I N E  C O D E  S T A R T S  H E R E       *
*          -----------------------------------------------      *
*                                                               *
* ************************************************************* *
 PROCEDURE DIVISION USING EXIT-FUNCTION-CODE
                          EXIT-WORK-AREA-PTR
                          EXIT-BACK-TRANS-CMD-PTR
                          EXIT-LIST-NAME
                          EXIT-GROUP-NAME
                          EXIT-OBJECT-TYPE
                          EXIT-OBJECT-NAME
                          EXIT-KEYWORD-TYPE
                          EXIT-KEYWORD-LENGTH
                          EXIT-KEYWORD-VALUE.
*
* Perform appropriate action, according to EXIT-FUNCTION-CODE
*
     EVALUATE EXIT-FUNCTION-CODE
* Open output file and initialize BUFFER to SPACES.
        WHEN INITIAL-CALL PERFORM INITIALISE
* Store current list name.
        WHEN LIST-START-CALL
           MOVE EXIT-LIST-NAME TO CURRENT-LIST-ID
* Store current group name and output an ADD command if there
* is a current list.
        WHEN GROUP-START-CALL
           PERFORM
```

```
               MOVE EXIT-GROUP-NAME TO CURRENT-GROUP-ID
               IF CURRENT-LIST-ID NOT EQUAL SPACES
                  THEN PERFORM
                          PERFORM ADD-GROUP-TO-LIST
                          MOVE LIST-PREFIX TO FLUSH-PREFIX
                          PERFORM FLUSH-BUFFER
                       END-PERFORM
               END-IF
            END-PERFORM
* Output a partial define command.
         WHEN OBJECT-START-CALL
            PERFORM
               PERFORM PROCESS-OBJECT
               MOVE GROUP-PREFIX TO FLUSH-PREFIX
               PERFORM FLUSH-BUFFER
            END-PERFORM
* If the keyword has a value then write the keyword and value to
* the output buffer. Only write the buffer if the keyword is
* DESCRIPTION or if we overflow the buffer.
         WHEN KEYWORD-CALL
            PERFORM
               IF EXIT-KEYWORD-LENGTH NOT EQUAL ZERO
                  THEN PERFORM
                          IF EXIT-KEYWORD-TYPE = 'DESCRIPTION'
                             THEN PERFORM
                                     IF BUFFER-PTR > BUFFER-INDENT
                                        THEN PERFORM FLUSH-BUFFER
                                     END-IF
                                     PERFORM PROCESS-KEYWORD
                                     PERFORM FLUSH-BUFFER
                                  END-PERFORM
                             ELSE PERFORM
                                     IF BUFFER-PTR = 1
                                        THEN MOVE BUFFER-INDENT
                                                     TO BUFFER-PTR
                                     END-IF
                                     PERFORM PROCESS-KEYWORD
                                  END-PERFORM
                          END-IF
                       END-PERFORM
               END-IF
            END-PERFORM
* Flush the buffer at the end of a resource.
         WHEN OBJECT-END-CALL PERFORM FLUSH-BUFFER
* Clear current group name at the end of the group.
         WHEN GROUP-END-CALL MOVE SPACES TO CURRENT-GROUP-ID
* Clear current list name at the end of the list .
         WHEN LIST-END-CALL MOVE SPACES TO CURRENT-LIST-ID
* Close output file.
         WHEN FINAL-CALL PERFORM CLEAN-UP
```

```
* Signal invalid call type to DFHCULIS.
        WHEN OTHER MOVE INVALID-CALL-TYPE TO RETURN-CODE
     END-EVALUATE
*
* Must use GOBACK so that COBOL data areas are preserved and usable
* on the next invocation of the program (eg CURRENT-LIST-ID and
* CURRENT-GROUP-ID).
*
     IF EXIT-FUNCTION-CODE = FINAL-CALL THEN
        STOP RUN
     ELSE
        GOBACK.
*
 ADD-GROUP-TO-LIST.
*
* Place command on output file of the form:
*          ADD GROUP( ... ) LIST( ... )
*
* If there is no current list then this routine should not be
* called.
*
     STRING 'ADD GROUP(' DELIMITED BY SIZE
            CURRENT-GROUP-ID DELIMITED BY SPACE
            ') LIST(' DELIMITED BY SIZE
            CURRENT-LIST-ID DELIMITED BY SPACE
            ')' DELIMITED BY SIZE
            INTO BUFFER
     END-STRING.
*
 ADD-KEYWORD-TO-BUFFER.
*
* Routine to write a keyword to the output buffer.
* Note that parentheses are placed around the keyword value and
* a space is appended to the string for padding.
*
     STRING EXIT-KEYWORD-TYPE DELIMITED BY SPACE
            '(' EXIT-KEYWORD-VALUE ') ' DELIMITED BY SIZE
            INTO BUFFER
            WITH POINTER BUFFER-PTR
     END-STRING.
*
*
*
 SPLIT-KEYWORD.
*
* This procedure splits a keyword value(s) over
* multiple lines. As much of the value as possible is put
* out on the first line, and a continuation character (*)
* placed in column 72. Subsequent lines contain 71
* Keyword characters plus a continuation character, apart
```

```
* from the final line which contains the remainder of the
* Keyword value and a right parenthesis.
*
* Put keyword name at the start of a new line
     STRING EXIT-KEYWORD-TYPE DELIMITED BY SPACE
            '(' DELIMITED BY SIZE
            INTO BUFFER
            WITH POINTER BUFFER-PTR
     END-STRING.
*
* Initialize variables required for data formatting.
*
     MOVE SPACES TO TEMP-BUFFER
     MOVE Ø TO KEYWORD-SCAN-PTR
     MOVE 1 TO TEMP-BUFFER-PTR
     MOVE Ø TO CHARACTER-COUNT
*
* Now scan keyword value(s) and fill output buffer. When
* full, flush buffer and continue processing for keyword
* creating output records until all keyword value(s)
* processed.
*
     PERFORM PERFORM-SPLIT
             UNTIL KEYWORD-SCAN-PTR =
                   EXIT-KEYWORD-LENGTH.
*
* Write a closing ')' to the output buffer.
*
     STRING TEMP-BUFFER DELIMITED BY SPACE
            ') ' DELIMITED BY SIZE
            INTO BUFFER
            WITH POINTER BUFFER-PTR
     END-STRING.
*
 PERFORM-SPLIT.
*
* This routine sequentially transfers characters from
* EXIT-KEYWORD-VALUE to TEMP-BUFFER. If TEMP-BUFFER is
* filled, then its contents are written to BUFFER,
* a continuation character concatenated in column 72. The
* contents of BUFFER are then written to the output file
* via a call to FLUSH-BUFFER.
*
      ADD 1 TO KEYWORD-SCAN-PTR.
      ADD 1 TO CHARACTER-COUNT.
*
* Calculate line length so far.
*
      ADD CHARACTER-COUNT
      BUFFER-PTR
```

```
      GIVING CURRENT-LINE-LENGTH.
*
* Find the next delimited string
*
      STRING EXIT-KEYWORD-CHAR(KEYWORD-SCAN-PTR)
             DELIMITED BY SIZE
             INTO TEMP-BUFFER
             WITH POINTER TEMP-BUFFER-PTR
      END-STRING.
*
* Check for full output line, ie 71 characters.
*
      IF CURRENT-LINE-LENGTH > 71 THEN
         STRING TEMP-BUFFER DELIMITED BY SPACE
                ASTERISK DELIMITED BY SIZE
                INTO BUFFER
                WITH POINTER BUFFER-PTR
         END-STRING
         PERFORM FLUSH-BUFFER
         MOVE SPACES TO TEMP-BUFFER
         MOVE 1 TO BUFFER-PTR
         MOVE 1 TO TEMP-BUFFER-PTR
         MOVE Ø TO CHARACTER-COUNT
      END-IF.
*
 PROCESS-OBJECT.
*
* Routine to write an object type to the output file
*
     STRING 'DEFINE ' DELIMITED BY SIZE
            EXIT-OBJECT-TYPE DELIMITED BY SPACE
            '(' DELIMITED BY SIZE
            EXIT-OBJECT-NAME DELIMITED BY SPACE
            ') GROUP(' DELIMITED BY SIZE
            CURRENT-GROUP-ID DELIMITED BY SPACE
            ')' DELIMITED BY SIZE
            INTO BUFFER
            WITH POINTER BUFFER-PTR
     END-STRING.
*
 PROCESS-KEYWORD.
*
* Routine to process a keyword and its value.
* If the keyword and its value will fit in the remaining buffer
* space then write them. Otherwise flush the buffer and place
* the keyword and its value in an empty buffer. If it still won't
* fit, then issue an error message for that keyword.
*
* Calculate the length of the keyword and its value (including
* parentheses) and then check to see if it will fit in the buffer.
```

```
* If it won't fit, flush the buffer and retry. If it still won't
* then this is an error.
*
     MOVE ZERO TO NUMBER-OF-SPACES
     INSPECT EXIT-KEYWORD-TYPE
             TALLYING NUMBER-OF-SPACES
             FOR ALL SPACE
     SUBTRACT NUMBER-OF-SPACES FROM MAX-TYPE-LENGTH
              GIVING KEYWORD-TYPE-LENGTH
     ADD KEYWORD-TYPE-LENGTH EXIT-KEYWORD-LENGTH
         PARENTHESIS-LENGTH
         GIVING KEYWORD-LENGTH
     SUBTRACT BUFFER-PTR FROM 72 GIVING BUFFER-SPACE
     IF BUFFER-SPACE > KEYWORD-LENGTH - 1
        THEN PERFORM ADD-KEYWORD-TO-BUFFER
        ELSE PERFORM
            IF BUFFER-PTR > BUFFER-INDENT
            THEN
                PERFORM FLUSH-BUFFER
            END-IF
                IF EXIT-KEYWORD-TYPE NOT EQUAL 'DESCRIPTION'
                   THEN MOVE BUFFER-INDENT TO BUFFER-PTR
                END-IF
                SUBTRACT BUFFER-PTR FROM 72 GIVING BUFFER-SPACE
                IF BUFFER-SPACE > KEYWORD-LENGTH - 1
                   THEN PERFORM ADD-KEYWORD-TO-BUFFER
                   ELSE PERFORM SPLIT-KEYWORD
                END-IF
             END-PERFORM
     END-IF.
*
 FLUSH-BUFFER.
*
* Routine to write the buffer to the output file.
* If the group (for DEFINE) or the list name (for ADD) begin
* 'DFH' then prefix the output with DFHCSDUP comment
* character, '*'.
*
     MOVE SPACES TO WRITE-BUFFER
     IF FLUSH-PREFIX = 'DFH'
        THEN STRING ASTERISK BUFFER DELIMITED BY SIZE
                    INTO WRITE-BUFFER
             END-STRING
        ELSE MOVE BUFFER TO WRITE-BUFFER
     END-IF
     WRITE WRITE-BUFFER.
     MOVE SPACES TO BUFFER.
     MOVE 1 TO BUFFER-PTR.
*
 INITIALISE.
```

```
*
* Routine called at initial call to set up working environment.
*
     OPEN OUTPUT DATA-OUT.
     MOVE SPACES TO BUFFER.
     MOVE 1 TO BUFFER-PTR.
     MOVE SPACES TO CURRENT-LIST-ID.
     MOVE SPACES TO CURRENT-GROUP-ID.
*
 CLEAN-UP.
*
* Routine called at final call to clean up
*
     CLOSE DATA-OUT.
```

CICSCSD EXEC

```
/* REXX */
/* CICS CSD utility
   C-List CICSCSD
   Called by job batch or TSO user command.
   It executes the subdivision of CICS list in CICS groups.
   The relative output is written into a partitioned file.
                                                                  */
Trace ?o
PARSE ARG cxfili cxfilo
endfin = 'NO'
/*****************************************************************/
/* Alloc input file                                              */
/*****************************************************************/
ADDRESS TSO
  dd=OUTTRAP(dd.)
  "ALLOC DA('"cxfili"') F(CXFILI) SHR REUSE"
  dd=OUTTRAP('OFF')
   codret = rc
   if codret ¬= Ø then do
                        typfunc = 'Allocfili'
                        Signal CX_Error_func
                        say mess
                        Return
                       End
/*****************************************************************/
/* Read input file                                               */
/*****************************************************************/
"NEWSTACK"
  dd=OUTTRAP(dd.)
  "EXECIO Ø DISKR CXFILI (OPEN"
  "EXECIO 1 DISKR CXFILI"
  dd=OUTTRAP('OFF')
```

```
  codret = rc
  if codret ¬= Ø then do
                       typfunc = 'Readfili'
                       Signal CX_Error_func
                       say mess
                       Return
                      End
Call CX_read_func
/*************************************************************/
/* Alloc output file; Routine to write PDS file.             */
/*************************************************************/
If define = 'ADD GROUP' then
   do
   "ALLOC DA('"cxfilo"(NEWLIST)') F(CXFILOA) SHR REUSE"
   "EXECIO Ø DISKW CXFILOA (OPEN"
   Call CX_write_func
   "EXECIO 1 DISKW CXFILOA "
   "EXECIO Ø DISKW CXFILOA (FINIS"
   "ALLOC DA('"||cxfilo||"("||GRUPPO||")') F(CXFILO) SHR REUSE"
   "EXECIO Ø DISKW CXFILO (OPEN"
   "EXECIO 1 DISKR CXFILI"
   CALL CX_read_func
END
Do While endfin = 'NO'
    Do While endfin = 'NO' & define <> 'ADD GROUP'
       Call CX_write_func
       "EXECIO 1 DISKW CXFILO "
       "EXECIO 1 DISKR CXFILI"
       Call CX_read_func
   End
   "EXECIO Ø DISKW CXFILO (FINIS"
   if define = 'ADD GROUP' then
      Do
      "ALLOC DA('"||cxfilo||"(NEWLIST)') F(CXFILOA) SHR"
      "EXECIO Ø DISKR CXFILOA (OPEN"
      "EXECIO * DISKR CXFILOA (STEM NEWLIST."
      "EXECIO Ø DISKR CXFILOA (FINIS"
      "ALLOC DA('"||cxfilo||"(NEWLIST)') F(CXFILOA) SHR REUSE"
      "EXECIO Ø DISKW CXFILOA (OPEN"
      "EXECIO * DISKW CXFILOA (STEM NEWLIST."
      Call CX_write_func
      "EXECIO 1 DISKW CXFILOA"
      "EXECIO Ø DISKW CXFILOA (FINIS"
      "ALLOC DA('"||cxfilo||"("||GRUPPO||")') F(CXFILO) SHR REUSE"
      "EXECIO Ø DISKW CXFILO (OPEN"
      "EXECIO 1 DISKR CXFILI"
       Call CX_read_func
   End
END
```

```
"EXECIO Ø DISKR CXFILI (FINIS"
"FREE F(CXFILI CXFILO CXFILOA)"
Exit
CX_Error_func:
/*************************************************************/
/* Errors routine                                            */
/*************************************************************/
Select
when typfunc = 'Allocfili' then
  mess = 'Allocation INPUT file in error.'
when typfunc = 'Readfili' then
  mess = 'Read INPUT file in error.'
Otherwise nop
End
Return
CX_read_func:
 PULL cxreci.1
 define = SUBSTR(cxreci.1,1,9)
 ngroup = SUBSTR(cxreci.1,11,8)
 Parse Value ngroup With GRUPPO ')'
 if rc = 2 then
    endfin = 'YES'
Return
CX_WRITE_FUNC:
 cxreco.1 = cxreci.1
 PUSH cxreco.1
Return
```

SAMPLE JOB TO EXECUTE THE DFH0CBDC PROGRAM AND CICSCSD CLIST

```
//ZZNS611M JOB (ZZNSØØØØ),
//         CLASS=S,REGION=ØM,
//         MSGCLASS=X,
//         MSGLEVEL=(1,1),
//         NOTIFY=&SYSUID
//*
/*JOBPARM BYTES=999999,LINES=9999
//*
//*********************************************************************
//* DELETE INPUT/OUTPUT FILE FOR CSD UTILITY                          *
//*********************************************************************
//DELETE EXEC  PGM=IDCAMS,REGION=1M
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 DELETE ZZNS611.CICS.DFHCSD.PDS NONVSAM
 DELETE ZZNS611.CICS.DFHCSD.SEQ NONVSAM
/*
//*
```

```
//*********************************************************************
//* DEFINE INPUT/OUTPUT DATASET FOR CSD UTILITY                       *
//*********************************************************************
//DEFINE   EXEC PGM=IEFBR14
//DD1      DD  DISP=(NEW,CATLG,DELETE),
//             DSN=ZZNS611.CICS.DFHCSD.SEQ,
//             VOL=SER=CIXH02,UNIT=3390,
//             DCB=(DSORG=PS,BLKSIZE=80,RECFM=F,LRECL=80),
//             SPACE=(CYL,(5,5))
//DD2      DD  DISP=(NEW,CATLG,DELETE),
//             DSN=ZZNS611.CICS.DFHCSD.PDS,
//             VOL=SER=CIXH02,UNIT=3390,
//             DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=80),
//             SPACE=(CYL,(5,5,5))
//*
//*********************************************************************
//* EXEC DFH0CBDC TO EXTRACT CSD RESOURCE INTO SEQUENTIAL DATASET     *
//*********************************************************************
//CSDUP   EXEC PGM=DFHCSDUP,REGION=4096K,
//             PARM='CSD(READWRITE),PAGESIZE(60),COMPAT'
//STEPLIB   DD DSN=ZZNS611.LOAD,DISP=SHR
//          DD DSN=CICS.SDFHLOAD,DISP=SHR
//          DD DSN=CICS.LIBRARY.USER.LOAD,DISP=SHR
//DFHCSD    DD DSN=CICS.SVIL.DFHCSD,DISP=SHR
//CBDOUT    DD DSN=ZZNS611.CICS.DFHCSD.SEQ,DISP=SHR
//SYSOUT    DD SYSOUT=*
//SYSABOUT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//SYSIN     DD *
 EXTRACT LIST(EMS4LIST) OBJECTS USERPROGRAM(DFH0CBDC)
/*
//*
//*********************************************************************
//* EXEC CICSCSD CLIST TO SUBDIVIDE THE RESOURCES OF A CICS LIST INTO *
//* CICS GROUPS.                                                      *
//* CONVERT SEQUENTIAL INPUT FILE INTO PARTITIONED OUTPUT FILE        *
//*********************************************************************
//REXX1    EXEC PGM=IKJEFT01,DYNAMNBR=30,REGION=4096K
//SYSEXEC  DD DSN=ZZNS611.LIB.CLIST,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
CICSCSD ZZNS611.CICS.DFHCSD.SEQ ZZNS611.CICS.DFHCSD.PDS
/*
/*
```

Espedito Morvillo
Systems Programmer (Italy)

Dynamic Transaction Routing

Here is some code you may find useful for CICS Dynamic Transaction Routing. The code is installed as a user-replaceable module and executes in the TOR whenever a transaction defined as DYNAMIC(YES) is executed or an undefined transaction code is entered.

You must define a PPT for the program and specify 'DTRPGM=' on the CICS start-up parameters.

We implemented this code to address a CPU constraint on one of our AORs. We now have a generic AOR, which we can replicate as many times as we need to fit our capacity requirements.

SYSTEM REQUIREMENTS

Our CICS system is Release 4.1. Code has not been tested on any other release. There are no special operating system requirements.

NAMING CONVENTION

A simple naming of sysids is used – any connection to the TOR with the same 3-character prefix in its sysid is considered to be an AOR eligible for transaction routing. AORs are searched for and found dynamically. This search is done every five minutes, enabling new AORs to be used shortly after they are started.

CICS AFFINITIES

Before you can route your transaction to more than one region, you have to understand any affinities that exist. The most common one is temporary storage for pseudoconversational programming.

Our application has this type of constraint. Although you can now function ship temporary storage requests, we chose not to do this. Instead, whenever a particular terminal does an application log-on, an AOR is selected. This AOR is used for that terminal until the next application log-on.

We still produced excellent load balancing with this technique.

If you cannot easily tell the start of an application affinity, you could simply route all transactions from a particular terminal to one AOR, deleting the affinity at CICS log-off time using a node error program.

Another affinity problem is CICS/VSAM. We function-shipped our CICS/VSAM to make this work. This caused a considerable CPU overhead, which we offset slightly by using shared data tables for some files. In our case, the overall CPU increase was a price worth paying for relief of the single TCB constraint of our AOR.

ROUTING ALGORITHM

The algorithm is a simple round-robin. Once an AOR is selected it is used until the next application-level log-on.

ROUTING ERRORS

The routing program routes only to AORs known to be acquired. The actual status is checked once every five minutes.

If a routing error occurs and no affinity exists, another AOR is tried automatically.

If an affinity already exists, an error message is sent. Pressing Enter forces an application relog-on (breaking the affinity) and another AOR is selected.

LCIDYP00

```
LCIDYPØØ AMODE 31
*********************************************************************
*
*  LCIDYPØØ CICS Dynamic Transaction routing exit for BANK
*
*  Select an AOR to service a dynamically-routed transaction.
*  AORs are selected round-robin at proiv log-on time (PRCL).
*
*  After log-on, an affinity exists and all transactions for that termid
*  are routed to the same AOR unless a routing error occurs.
*
*  AOR status is maintained in shared storage. Up to 1Ø (MAXAOR) AORs
*  Can be part of an AOR group. An AOR group is a set of AORs with the
```

```
*  same 3 character SYSID prefix.
*********************************************************************
LCIDYPØØ RMODE ANY
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
LCIDYPØØ DFHEIENT CODEREG=(12),DATAREG=(13),EIBREG=(11)
*
CHECK_FOR_COMMAREA EQU *
         USING DFHDYPDS,5
         USING TMP,7
         USING AOR,6
         ICM   5,15,DFHEICAP
         BZ    EXIT_POINT
         CLI   DYRDTRXN,C'Y'           Undefined tran code ?
         BE    EXIT_POINT              Not interested
*
         MVC   DYRRETC,RETURN_CODE8
         CLC   EIBCALEN,=AL2(DYRCLEN)
         BNE   EXIT_POINT
*
*
         CLI   DYRFUNC,C'3'            INVOKED FOR STATIC ROUTE
         BE    EXIT_POINT
*
         CLI   DYRFUNC,C'2'            INVOKED AFTER TRANSACTION END?
         BE    TRANTERM
*
         CLI   DYRFUNC,C'4'            INVOKED AFTER AOR ABEND
         BE    ROUTE_ABEND
         CLI   DYRFUNC,C'Ø'            INITIAL INVOCATION OF ROUTER?
         BE    ROUTE_SELECT
*
         CLI   DYRFUNC,C'1'            INVOKED DUE TO ROUTING ERROR?
         BE    ROUTE_SELECT
*
         MVC   DYRRETC,RETURN_CODE8    INVALID REQUEST.
         B     EXIT_POINT
```

```
*
ROUTE_SELECT EQU *
*
*        Must be invoked for route or route fail
*        so do common processing
*
         MVI   PROCFLAG,X'ØØ'
*
         MVC   USERTSQ(4),=C'DYPQ'     TSQ with user affinities
         MVC   USERTSQ+4(4),EIBTRMID   C'DYPQ'YYtermid
         EXEC CICS HANDLE CONDITION QIDERR(NO_USER_TSQ)                X
               ITEMERR(NO_USER_TSQ)
CHECK_FOR_USER_TSQ EQU  *
         MVC   ENTRYNUM,=H'1'
         EXEC CICS READQ TS QUEUE(USERTSQ) INTO(USRDATA)               X
               ITEM(ENTRYNUM)                                          X
               NUMITEMS(MAXENTRIES)
         OI    PROCFLAG,PROCUSER+PROCADDR Sysid and storage address
         L     R7,USRADDR              Address shared storage
NO_USER_TSQ EQU  *
         EXEC  CICS HANDLE CONDITION QIDERR() ITEMERR()
         CLC   EIBTRNID,=C'PRCL'       Log-on transaction ?
         BNE   NOT_LOGON_TRAN          No
         TM    PROCFLAG,PROCUSER       User TSQ exists?
         BNO   NOT_LOGON_TRAN          No don't delete it
         EXEC  CICS DELETEQ TS QUEUE(USERTSQ)
         NI    PROCFLAG,255-PROCUSER   Reset user TSQ flag
NOT_LOGON_TRAN EQU  *
         TM   PROCFLAG,PROCUSER        User TSQ exists ?
         BNO  ROUTE_SELECT3            No go select route
         CLI   DYRFUNC,C'1'            INVOKED DUE TO ROUTING ERROR?
         BNE   ROUTE_SELECT2           No use existing affinity
****                                   Routing error occurred and
****                                   affinity exists.
****
         LH    R6,USRSLOT              Slot we tried to route to
         LA    R15,TMPENTS             1st slot entry
         SLL   R6,4                    * entry length
         AR    R6,R15                  Entry we tried to route to
         AP    AORFCNT,=P'1'           Update stats
NO_USABLE_AOR  EQU *
*                                      Send error message to user
*                                      Return transid PRCL will break
*                                      affinity and attempt to select
*                                      another link
         EXEC CICS SEND                                                X
               FROM(ROUTE_ERROR)                                       X
               FLENGTH(=AL4(ROUTE_ERROR_LENGTH))                       X
               ERASE
         MVC   DYRRETC,RETURN_CODE4       Terminate no error message
         EXEC CICS RETURN TRANSID('PRCL')
```

```
ROUTE_SELECT2 EQU *                    Existing affinity
         LH   R6,USRSLOT               Get slot number for user
         LA   R15,DYRUSER              User area
         AR   R15,R6                   Index to our slot
         MVI  Ø(R15),X'FF'             Note attempted to route
         SLL  R6,4                     *16 = offset to slot
         LA   R15,TMPENTS              1st entry
         AR   R6,R15                   our entry
         AP   AORRCNT,=P'1'            Add 1 to routing count
         MVC  DYRSYSID,USRSYSID        We must use existing SYSID
         B    ROUTE_OK
ROUTE_SELECT3 EQU *                    Come here to select an AOR
*
         EXEC  CICS ASKTIME                                          X
                ABSTIME(TIMENOW)
         EXEC CICS HANDLE CONDITION QIDERR(CREATE_TABLE)             X
                ITEMERR(CREATE_TABLE)
         TM    PROCFLAG,PROCADDR       Is shared storage addr known?
         BO    READ_AOR_ENTRY2         Yes - bypass TSQ read for addr
         EXEC CICS ASSIGN SYSID(USRTORID)
         MVC   TSQNAME(3),=C'AOR'
         MVC   TSQNAME+3(3),USRTORID   TSQ with shared storage ptr
         MVC   TSQNAME+6(2),=C'  '     C'AORxxx  '
         MVC   LISTNAME,TSQNAME
         LA    R1,L'LISTNAME
         STH   R1,LISTNAME_LENGTH
*
READ_FIRST_AOR_ENTRY EQU *
         MVC   ENTRYNUM,=H'1'
*
READ_AOR_ENTRY EQU *
         EXEC CICS READQ TS QUEUE(TSQNAME)                           X
               ITEM(ENTRYNUM)                                        X
               NUMITEMS(MAXENTRIES)                                  X
               INTO(USRADDR)
          L    R7,USRADDR
READ_AOR_ENTRY2 EQU *
          ZAP  TIMEWORK,TIMENOW        Current time
          TM   TMPFLAG,TMPAORS         Useable AOR exists ?
          BNO  READ_AOR_ENTRY3         No - check 3Ø sec
          SP   TIMEWORK,=P'3ØØØØØ'     - 5 minutes
          B    READ_AOR_ENTRY4
READ_AOR_ENTRY3 EQU *
          SP   TIMEWORK,=P'3ØØØØ'      - 3Ø secs
READ_AOR_ENTRY4 EQU *
          CP   TIMEWORK,TMPSCANT       Less than scan time?
          BL   LOCATE_CURRENT_AOR      Yes don't do scan
          ZAP  TMPSCANT,TIMENOW        Prevent anyone else doing scan
        EXEC CICS FORMATTIME                                         X
             ABSTIME(TIMENOW)                                        X
             TIMESEP(':')                                            X
```

```
             TIME(TMPSCANF)
*
*   This code is driven once every 5 minutes during normal
*   Processing, or every 30 seconds if no AORs exist
*   The AOR status is compared to the actual connection
*   Any new AORs discovered are added to the table dynamically
*
*
         EXEC CICS INQUIRE CONNECTION START
         NI    TMPFLAG,255-TMPAORS      Reset valid AOR flag
*
SCAN_AOR_LOOP EQU *
         EXEC CICS INQUIRE CONNECTION(CONN_NAME)                      X
               NETNAME(CONN_NETNAME)                                  X
               CONNSTATUS(CONN_STATUS) RESP(RESPONSE_CODE) NEXT
*
         CLC   RESPONSE_CODE,DFHRESP(NORMAL)
         BNE   SCAN_AOR_LOOP_END
*
         CLC   CONN_NAME(3),USRTORID    Matches tran SYSID prefix?
         BL    SCAN_AOR_LOOP            No
         BH    SCAN_AOR_LOOP_END        Processed all
*
*        Found connection we are interested in
*        Now locate the AOR for this SYSID and update its status
*
*
         LA    R6,TMPENTS               Address 1st AOR
SCAN_AOR_LOOP2 EQU *
         TM    AORFLAG,AORUSED          Slot in use
         BO    SCAN_AOR_CHECK           Yes check if it is
         MVC   AORSYSID,CONN_NAME        Found a new AOR
         OI    AORFLAG,AORUSED          This slot is used
         OI    PROCFLAG,PROCNEW         New AOR found
         B     SCAN_AOR_FOUND
SCAN_AOR_CHECK EQU  *
         CLC   AORSYSID,CONN_NAME       Is this our connection
         BNE   SCAN_AOR_NEXT            No loop to next
         NI    PROCFLAG,255-PROCNEW     Not a new AOR
SCAN_AOR_FOUND EQU  *
         BAL   R14,STATUS_CHECK         check/set AOR status
         B     SCAN_AOR_LOOP            Check next connection
SCAN_AOR_NEXT  EQU  *
         LA    R6,AORLEN(,R6)           Next AOR
         B     SCAN_AOR_LOOP2
*
*
SCAN_AOR_LOOP_END  EQU  *
         EXEC CICS INQUIRE CONNECTION END
         ZAP  TIMEWORK,TIMENOW         Current time
         SP   TIMEWORK,=P'3600000'      - 60 minutes
```

```
         CP   TIMEWORK,TMPSTATT       Less than stat time?
         BL   LOCATE_CURRENT_AOR      Yes don't do stats
         ZAP  TIMEWORK,TMPSTATT       Save last stats time
         ZAP  TMPSTATT,TIMENOW         Prevent anyone else doing stat
        EXEC CICS FORMATTIME                                          X
             ABSTIME(TIMENOW)                                         X
             TIMESEP(':')                                             X
             TIME(TMPSTATF)
*
*   This code is driven once every 60 minutes
*   The AOR stats are written to SYSLOG
*
        MVC  MSGWORK,MSG01            Move constants
*                                     Format previous stats time
        EXEC CICS FORMATTIME                                          X
             ABSTIME(TIMEWORK)                                        X
             TIMESEP(':')                                             X
             TIME(TIMEF)
         LA  R1,MSGWORK
         MVC MSG01TIM-MSG01(L'MSG01TIM,R1),TIMEF
         EXEC CICS WRITE OPERATOR                                     X
               TEXT(MSGWORK)
         LA    R6,TMPENTS              1st AOR entry
STATS_LOOP EQU  *
         TM    AORFLAG,AORUSED         AOR in use?
         BNO   LOCATE_CURRENT_AOR      No - all AORs processed
         MVC   MSGWORK,MSG02           Move constants
         LA    R1,MSGWORK
         MVC   MSG02AOR-MSG02(L'MSG02AOR,R1),AORSYSID Move name
         ED    MSG02RC-MSG02(L'MSG02RC,R1),AORRCNT Move route count
         ED    MSG02RF-MSG02(L'MSG02RF,R1),AORFCNT Move fails count
         MVC   MSG02STA-MSG02(L'MSG02STA,R1),=C'ACQ'    Assume acquired
         TM    AORFLAG,AORACQ          Aquired ?
         BO    STATS_LOOP_MSG          Yes
         MVC   MSG02STA-MSG02(L'MSG02STA,R1),=C'REL'  must be REL
STATS_LOOP_MSG EQU  *
         EXEC CICS WRITE OPERATOR                                     X
               TEXT(MSGWORK)
         ZAP   AORFCNT,=P'0'           Reset
         ZAP   AORRCNT,=P'0'           Reset
         LA    R6,AORLEN(,R6)          Next entry
         B     STATS_LOOP              Process
*                                     This is the start of the AOR
*                                     selection code
LOCATE_CURRENT_AOR EQU  *
         CLI   DYRFUNC,C'1'            Invoked due to routing error?
         BNE   LOCATE_CURRENT_AOR2     No
         LH    R6,USRSLOT              Get slot number we used
         SLL   R6,4                    * ent len
         LA    R15,TMPENTS             1st slot entry
         AR    R6,R15                  Entry we tried to route
```

```
         AP    AORFCNT,=P'1'           Add 1 to fail count
LOCATE_CURRENT_AOR2 EQU  *
         LA    R15,TMPENTS             1st slot entry
         LH    R6,TMPNEXT              Next AOR slot
         STH   R6,SAVENEXT             Save for loop check
         SLL   R6,4                    *16 (SLOT LEN)
         AR    R6,R15                  Point to next slot
*
TRY_THIS_AOR  EQU  *
         TM    AORFLAG,AORUSED+AORACQ  AOR OK ?
         BNO   TRY_NEXT_AOR
         LA    R15,DYRUSER             User area
         LH    R1,TMPNEXT              Current entry
         AR    R15,R1                  Index
         CLI   Ø(R15),X'FF'            Already tried to route this AOR?
         BE    TRY_NEXT_AOR            Yes try another
         MVC   DYRSYSID,AORSYSID       Use this AOR
USE_THIS_AOR  EQU   *
         MVC   USRSYSID,AORSYSID       Remember SYSID to use this user
         MVC   USRSLOT,TMPNEXT         Remember slot number
         LH    R1,TMPNEXT
         LA    R15,DYRUSER             User area
         AR    R15,R1                  Index to slot number
         MVI   Ø(R15),X'FF'            Note attempted to route to AOR
         AP    AORRCNT,=P'1'           Add 1 to stats
         LA    R6,AORLEN(,R6)          Next slot
         TM    AORFLAG,AORUSED         Slot in use?
         BO    REWRITE_ENTRY           Yes can use this one next
         MVC   TMPNEXT,=H'-1'          End of used slots - go back
*
REWRITE_ENTRY EQU *
         LH    R1,TMPNEXT
         LA    R1,1(,R1)               Next slot
         STH   R1,TMPNEXT              Save for next user
*        Create user TSQ to route all subsequent trans till next log-on
*        (PRCL) is issued
         EXEC CICS WRITEQ TS QUEUE(USERTSQ)                            X
               ITEM(ENTRYNUM)                                          X
               FROM(USRDATA)                                           X
               MAIN
ROUTE_OK EQU   *
         MVI   DYROPTER,C'N'
         MVC   DYRRETC,RETURN_CODEØ
         B     EXIT_POINT
*
TRY_NEXT_AOR  EQU   *
         LA    R6,AORLEN(,R6)          Point to next AOR
         TM    AORFLAG,AORUSED         At end of AORs?
         BO    TRY_NEXT_AOR2           No - try next
         MVC   TMPNEXT,=H'-1'          Go back to AOR Ø
         LA    R6,TMPENTS              Point to AOR Ø
```

```
TRY_NEXT_AOR2 EQU   *
         LH    R1,TMPNEXT
         LA    R1,1(,R1)              Next AOR
         STH   R1,TMPNEXT
         CLC   TMPNEXT,SAVENEXT       Loop check
         BNE   TRY_THIS_AOR           Not looped
         B     NO_USABLE_AOR
*
TRANTERM EQU   *
         MVI   DYROPTER,C'N'
         MVC   DYRRETC,RETURN_CODE8
         B     EXIT_POINT
*
ROUTE_ABEND EQU *
         MVI   DYROPTER,C'N'
         MVC   DYRRETC,RETURN_CODEØ
         B     EXIT_POINT
*
CREATE_TABLE EQU *
*
*        Create AOR status table.
*        Because many tasks might try to do this, ENQ is issued and
*        retest is done after successful enq.
*
         EXEC CICS ENQ RESOURCE(LISTNAME) LENGTH(LISTNAME_LENGTH)
         EXEC CICS HANDLE CONDITION                                    X
               QIDERR(CREATE_TABLE_TSQ)
*
         EXEC CICS READQ TS QUEUE(TSQNAME) INTO(USRADDR)               X
               ITEM(ENTRYNUM)
         B    GET_LIST_OF_AORS_RETURN   TS CREATED DURING ENQUEUE
CREATE_TABLE_TSQ EQU *
*
         EXEC CICS GETMAIN                                             X
               SET(R7)                                                 X
               INITIMG(X'ØØ')                                          X
               FLENGTH(=AL4(TMPLEN))                                   X
               SHARED
         ST    R7,USRADDR
         MVC   MSGWORK,MSGØ4           Init message
         LA    R2,MSGWORK
         UNPK  HEXWORK,USRADDR(5)
         TR    HEXWORK(8),HEXTAB
         MVC   MSGØ4STO-MSGØ4(L'MSGØ4STO,R2),HEXWORK
         EXEC CICS WRITE OPERATOR                                      X
               TEXT(MSGWORK)
         MVC    TMPNEXT,=H'Ø'          First AOR entry
         MVC    TMPSCANT,TIMENOW       Set time of last scan
         MVC    TMPSTATT,TIMENOW       Set time of last stats init
        EXEC CICS FORMATTIME                                           X
             ABSTIME(TIMENOW)                                          X
```

```
             TIMESEP(':')                                          X
             TIME(TMPSCANF)
          MVC  TMPSTATF,TMPSCANF
          LA     R6,TMPENTS             1st entry
          MVC   ENTRYNUM,=H'1'
          OI    PROCFLAG,PROCNEW        Note all AORs are new
GET_LIST_OF_AORS EQU *
          EXEC CICS INQUIRE CONNECTION START
          NI    TMPFLAG,255-TMPAORS      Reset valid AOR flag
*
GET_AOR_LOOP EQU *
          EXEC CICS INQUIRE CONNECTION(CONN_NAME)                   X
                NETNAME(CONN_NETNAME)                               X
                CONNSTATUS(CONN_STATUS) RESP(RESPONSE_CODE) NEXT
*
          CLC   RESPONSE_CODE,DFHRESP(NORMAL)
          BNE   GET_LIST_OF_AORS_END
*
          CLC   CONN_NAME(3),USRTORID   Matches tran SYSID prefix?
          BL    GET_AOR_LOOP            Less loop till we get there
          BH    GET_LIST_OF_AORS_END    All our entries processed
          OI    AORFLAG,AORUSED         This slot is used
          MVC   AORSYSID,CONN_NAME      AOR SYSID
          ZAP   AORFCNT,=P'0'           Route fail count
          ZAP   AORRCNT,=P'0'           Route count
*
          BAL   R14,STATUS_CHECK        Check/set AOR Status
          LA    R6,AORLEN(,R6)          Next entry
          B     GET_AOR_LOOP
*
GET_LIST_OF_AORS_END EQU *
          EXEC CICS INQUIRE CONNECTION END
WRITE_ENTRIES EQU *
          EXEC CICS WRITEQ TS QUEUE(TSQNAME)                        X
                ITEM(ENTRYNUM)                                      X
                FROM(USRADDR)                                       X
                MAIN
*
          B     GET_LIST_OF_AORS_RETURN
*
GET_LIST_OF_AORS_RETURN EQU *
          EXEC CICS DEQ RESOURCE(LISTNAME) LENGTH(LISTNAME_LENGTH)
          EXEC CICS HANDLE CONDITION                                X
                QIDERR()
          B     READ_FIRST_AOR_ENTRY
*
STATUS_CHECK EQU  *
          ST   R14,STATUS_R14
*         Compare AOR status to actual
*         Update status if required
*         Issue WTO if status has changed, or if 1st time
```

```
*
*        This code is driven each AORSCAN, and also on first
*        time building of shared storage.
*        A message is issued if the status changes unless PROCNEW
*        is set, in which case the message is unconditional.
*
         DS    ØH
         MVC   MSGWORK,MSGØ3       Move message constants
         LA    R1,MSGWORK
         MVC   MSGØ3AOR-MSGØ3(L'MSGØ3AOR,R1),AORSYSID
         CLC   CONN_STATUS,DFHVALUE(ACQUIRED)
         BNE   STATUS_CHECK_REL    No
         MVC   MSGØ3STA-MSGØ3(L'MSGØ3STA,R1),=C'ACQ'
         OI    TMPFLAG,TMPAORS     Note usable AOR exists
         TM    AORFLAG,AORACQ      We think acquired ?
         BO    STATUS_CHECK_MSG    Do we need to issue MSG?
         OI    AORFLAG,AORACQ      AOR is acquired
         B     STATUS_CHECK_MSG2   We definitely need to issue MSG
STATUS_CHECK_REL EQU  *
         MVC   MSGØ3STA-MSGØ3(L'MSGØ3STA,R1),=C'REL'
         TM    AORFLAG,AORACQ      We think released ?
         BZ    STATUS_CHECK_MSG    Do we need to issue MSG?
         NI    AORFLAG,255-AORACQ  AOR is released
         B     STATUS_CHECK_MSG2   We definitely need to issue MSG
STATUS_CHECK_MSG EQU  *
         TM    PROCFLAG,PROCNEW    Creating new entry ?
         BNO   STATUS_CHECK_EXIT   No - no need to issue MSG
STATUS_CHECK_MSG2 EQU  *
         EXEC CICS WRITE OPERATOR                                    X
               TEXT(MSGWORK)
STATUS_CHECK_EXIT EQU *
         L    R14,STATUS_R14
         BR   R14
         EJECT
ROUTE_ERROR  EQU *
             DC CL8Ø'LCIDYP1Ø Your request could not be processed due tX
               o a system problem.'
             DC CL8Ø'Press ENTER to relog-on to the Caller system.'
             DC CL8Ø'If this problem persists, please contact the HelpdX
               esk.'
ROUTE_ERROR_LENGTH  EQU *-ROUTE_ERROR
*
MSGØ1        DC CL8Ø' '
             ORG  MSGØ1
             DC  C'LCIDYPØ1 - AOR statistics follow (reset at '
MSGØ1TIM     DS  CL8
             DC  C').'
             ORG
*
MSGØ2        DC  CL8Ø' '
             ORG  MSGØ2
```

```
             DC  C'LCIDYPØ2 - AOR : '
MSGØ2AOR     DS  CL4
             DC  C' Status : '
MSGØ2STA     DS  CL3
             DC  C' Routed count : '
MSGØ2RC      DC  X'4Ø2Ø2Ø2Ø2Ø2Ø212Ø'
             DC  C' failures : '
MSGØ2RF      DC  X'4Ø2Ø2Ø2Ø2Ø2Ø212Ø'
             DC  C'.'
             ORG
*
MSGØ3        DC  CL8Ø' '
             ORG  MSGØ3
             DC  C'LCIDYPØ3 - AOR : '
MSGØ3AOR     DS  CL4
             DC  C' Status is '
MSGØ3STA     DS  CL3
             DC  C'.'
             ORG
*
MSGØ4        DC  CL8Ø' '
             ORG  MSGØ4
             DC  C'LCIDYPØ4 - Initializing shared storage at '
MSGØ4STO     DS  CL8
             DC  C'.'
             ORG
*
RETURN_CODEØ DC F'Ø'
RETURN_CODE4 DC F'4'
RETURN_CODE8 DC F'8'
HEXTAB   DC  256X'ØØ'
         ORG HEXTAB+X'FØ'
         DC  C'Ø123456789ABCDEF'
         ORG
         LTORG ,
*
EXIT_POINT DS  ØH
         EXEC CICS RETURN
*
         DFHEISTG
TIMENOW     DS  PL8
TIMEWORK    DS  PL8
RESPONSE_CODE DS F
CONN_STATUS DS F
STATUS_R14  DS F
SAVENEXT    DS H
LISTNAME_LENGTH DS AL2
LISTNAME  DS   CL8
TSQNAME   DS   CL8
USERTSQ   DS   CL8
HEXWORK   DS   CL9
```

```
          DS   ØF
USRDATA   DS   CL14
          ORG  USRDATA
USRSYSID  DS   CL4                      Affinity for this user
USRTORID  DS   CL4                      TOR ID
USRADDR   DS   CL4                      Address of shared storage
USRSLOT   DS   H                        Slot number for this Sysid
          ORG
CONN_NAME DS   CL4
CONN_NETNAME DS CL8
PROCFLAG  DS   X
PROCUSER  EQU  1                        Sysid affinity exists
PROCNEW   EQU  2                        New AOR found
PROCADDR  EQU  4                        Address found in DYRUSER
REPLY_MESSAGE_LENGTH DS F
REPLY_MESSAGE DS C
MAXENTRIES DS  H
TIMEF      DS  CL8
MSGWORK    DS  CL8Ø
ENTRYNUM DS    H
         COPY  DFHDYPDS
TMP       DSECT                         TSQ - SHARED STORAGE
TMPHDR    EQU  *                        TSQ HEADER
TMPNEXT   DS   H                        NEXT AOR TO SELECT
TMPFLAG   DS   X                        Status flag
TMPAORS   EQU  X'Ø1'                     Usable AORs exist
TNPFILL   DS   CL13                     Align nicely for CEDF
TMPSTATT  DS   PL8                      LAST AOR STATS Reset
TMPSTATF  DS   CL8                      LAST AOR STATS Reset formatted
TMPSCANT  DS   PL8                      LAST AOR SCAN TIME
TMPSCANF  DS   CL8                      LAST AOR SCAN TIME formatted
TMPHDRL   EQU  *-TMPHDR
TMPENTS   DS   CL(MAXAORS*AORLEN)   TMP ENTRIES - MAPPED BY AOR
TMPLEN    EQU  *-TMPHDR
*
MAXAORS   EQU  1Ø
AOR       DSECT
AORSYSID  DS    CL4
AORFLAG   DS   X
AORUSED   EQU  1                        AOR SLOT USED
AORACQ    EQU  2                        AOR AQCUIRED
AORFIL1   DS    XL3                     FILL TO 8 BYTES
AORRCNT   DS   PL4                      Count route requests
AORFCNT   DS   PL4                      Count route fails
AORLEN    EQU  *-AOR
*
         END
```

Joe Owens
Senior Technician
Standard Life Assurance Company (UK)

Interfacing CICS to SMTP – part 1

This package is intended to demonstrate one technique for interfacing CICS with the SMTP mail capability of TCP/IP for MVS. This package is only intended to send outbound mail and has no provision for reading mail responses, although the same technique could be used to do so.

MAILMAP.CPY

```
       Ø1  CONFIRMI.
         Ø2  FILLER                                  PICTURE X(12).
         Ø2  FILLER                                  PICTURE X.
       Ø1  CONFIRMO    REDEFINES   CONFIRMI.
         Ø2  FILLER                                  PICTURE X(12).
         Ø2  FILLER                                  PICTURE X.
       Ø1  ERRORI.
         Ø2  FILLER                                  PICTURE X(12).
         Ø2  FILLER                                  PICTURE X.
       Ø1  ERRORO      REDEFINES   ERRORI.
         Ø2  FILLER                                  PICTURE X(12).
         Ø2  FILLER                                  PICTURE X.
       Ø1  MAILMAPI.
         Ø2  FILLER                                  PICTURE X(12).
         Ø2  MAIL-DATEL                         COMP PICTURE S9(4).
         Ø2  MAIL-DATEF                              PICTURE X.
         Ø2  MAIL-DATEI                              PICTURE X(Ø1Ø).
         Ø2  MAIL-APPLIDL                       COMP PICTURE S9(4).
         Ø2  MAIL-APPLIDF                            PICTURE X.
         Ø2  MAIL-APPLIDI                            PICTURE X(ØØ8).
         Ø2  MAIL-TIMEL                         COMP PICTURE S9(4).
         Ø2  MAIL-TIMEF                              PICTURE X.
         Ø2  MAIL-TIMEI                              PICTURE X(ØØ8).
         Ø2  MAIL-USERIDL                       COMP PICTURE S9(4).
         Ø2  MAIL-USERIDF                            PICTURE X.
         Ø2  MAIL-USERIDI                            PICTURE X(ØØ8).
         Ø2  MAIL-TOL                           COMP PICTURE S9(4).
         Ø2  MAIL-TOF                                PICTURE X.
         Ø2  MAIL-TOI                                PICTURE X(Ø4Ø).
         Ø2  MAIL-REPLYL                        COMP PICTURE S9(4).
         Ø2  MAIL-REPLYF                             PICTURE X.
         Ø2  MAIL-REPLYI                             PICTURE X(Ø4Ø).
         Ø2  MAIL-FROML                         COMP PICTURE S9(4).
         Ø2  MAIL-FROMF                              PICTURE X.
         Ø2  MAIL-FROMI                              PICTURE X(Ø2Ø).
         Ø2  MAIL-SUBJECTL                      COMP PICTURE S9(4).
```

```
  Ø2  MAIL-SUBJECTF                              PICTURE X.
  Ø2  MAIL-SUBJECTI                              PICTURE X(Ø69).
  Ø2  BMSGTØØ1-MAILMAP-MAILMAP       OCCURS  15 TIMES.
    Ø3  MAIL-BODYL                          COMP PICTURE S9(4).
    Ø3  MAIL-BODYF                               PICTURE X.
    Ø3  MAIL-BODYI                               PICTURE X(Ø79).
  Ø2  MAIL-F1L                              COMP PICTURE S9(4).
  Ø2  MAIL-F1F                                   PICTURE X.
  Ø2  MAIL-F1I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F2L                              COMP PICTURE S9(4).
  Ø2  MAIL-F2F                                   PICTURE X.
  Ø2  MAIL-F2I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F3L                              COMP PICTURE S9(4).
  Ø2  MAIL-F3F                                   PICTURE X.
  Ø2  MAIL-F3I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F4L                              COMP PICTURE S9(4).
  Ø2  MAIL-F4F                                   PICTURE X.
  Ø2  MAIL-F4I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F5L                              COMP PICTURE S9(4).
  Ø2  MAIL-F5F                                   PICTURE X.
  Ø2  MAIL-F5I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F6L                              COMP PICTURE S9(4).
  Ø2  MAIL-F6F                                   PICTURE X.
  Ø2  MAIL-F6I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F7L                              COMP PICTURE S9(4).
  Ø2  MAIL-F7F                                   PICTURE X.
  Ø2  MAIL-F7I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F8L                              COMP PICTURE S9(4).
  Ø2  MAIL-F8F                                   PICTURE X.
  Ø2  MAIL-F8I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F9L                              COMP PICTURE S9(4).
  Ø2  MAIL-F9F                                   PICTURE X.
  Ø2  MAIL-F9I                                   PICTURE X(ØØ5).
  Ø2  MAIL-F1ØL                             COMP PICTURE S9(4).
  Ø2  MAIL-F1ØF                                  PICTURE X.
  Ø2  MAIL-F1ØI                                  PICTURE X(ØØ6).
  Ø2  MAIL-F11L                             COMP PICTURE S9(4).
  Ø2  MAIL-F11F                                  PICTURE X.
  Ø2  MAIL-F11I                                  PICTURE X(ØØ6).
  Ø2  MAIL-F12L                             COMP PICTURE S9(4).
  Ø2  MAIL-F12F                                  PICTURE X.
  Ø2  MAIL-F12I                                  PICTURE X(ØØ6).
Ø1  MAILMAPO    REDEFINES   MAILMAPI.
  Ø2  FILLER                                     PICTURE X(12).
  Ø2  FILLER                                     PICTURE X(2).
  Ø2  MAIL-DATEA                                 PICTURE X.
  Ø2  MAIL-DATEO                                 PICTURE X(Ø1Ø).
  Ø2  FILLER                                     PICTURE X(2).
  Ø2  MAIL-APPLIDA                               PICTURE X.
  Ø2  MAIL-APPLIDO                               PICTURE X(ØØ8).
  Ø2  FILLER                                     PICTURE X(2).
```

```
02  MAIL-TIMEA                              PICTURE X.
02  MAIL-TIMEO                              PICTURE X(008).
02  FILLER                                  PICTURE X(2).
02  MAIL-USERIDA                            PICTURE X.
02  MAIL-USERIDO                            PICTURE X(008).
02  FILLER                                  PICTURE X(2).
02  MAIL-TOA                                PICTURE X.
02  MAIL-TOO                                PICTURE X(040).
02  FILLER                                  PICTURE X(2).
02  MAIL-REPLYA                             PICTURE X.
02  MAIL-REPLYO                             PICTURE X(040).
02  FILLER                                  PICTURE X(2).
02  MAIL-FROMA                              PICTURE X.
02  MAIL-FROMO                              PICTURE X(020).
02  FILLER                                  PICTURE X(2).
02  MAIL-SUBJECTA                           PICTURE X.
02  MAIL-SUBJECTO                           PICTURE X(069).
02  BMSGT002-MAILMAP-MAILMAP       OCCURS  15 TIMES.
  03  FILLER                                PICTURE X(2).
  03  MAIL-BODYA                            PICTURE X.
  03  MAIL-BODYO                            PICTURE X(079).
02  FILLER                                  PICTURE X(2).
02  MAIL-F1A                                PICTURE X.
02  MAIL-F1O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F2A                                PICTURE X.
02  MAIL-F2O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F3A                                PICTURE X.
02  MAIL-F3O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F4A                                PICTURE X.
02  MAIL-F4O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F5A                                PICTURE X.
02  MAIL-F5O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F6A                                PICTURE X.
02  MAIL-F6O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F7A                                PICTURE X.
02  MAIL-F7O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F8A                                PICTURE X.
02  MAIL-F8O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F9A                                PICTURE X.
02  MAIL-F9O                                PICTURE X(005).
02  FILLER                                  PICTURE X(2).
02  MAIL-F10A                               PICTURE X.
02  MAIL-F10O                               PICTURE X(006).
```

```
           02  FILLER                                  PICTURE X(2).
           02  MAIL-F11A                               PICTURE X.
           02  MAIL-F11O                               PICTURE X(006).
           02  FILLER                                  PICTURE X(2).
           02  MAIL-F12A                               PICTURE X.
           02  MAIL-F12O                               PICTURE X(006).
```

MAILTEST.CBL

```
IDENTIFICATION DIVISION.
           SKIP1
       ****************************************************************
       *        I D E N T I F I C A T I O N      D I V I S I O N       *
       ****************************************************************
           SKIP3
        PROGRAM-ID.     MAILTEST.
           SKIP1
        AUTHOR.         DONALD GRINSELL.
           SKIP1
        INSTALLATION.   STATE OF MONTANA.
           SKIP1
        DATE-WRITTEN.   OCT 28, 1999.
           SKIP1
        DATE-COMPILED.
           SKIP2
        REMARKS.        C I C S    M A I L   T R A N S A C T I O N
           SKIP1
                        THIS PROGRAM IS USED TO TEST THE SENDMAIL PGM.
           SKIP3
       ****************************************************************
           EJECT
        ENVIRONMENT DIVISION.
           SKIP1
       ****************************************************************
       *        E N V I R O N M E N T       D I V I S I O N           *
       ****************************************************************
           SKIP3
        CONFIGURATION SECTION.
           SKIP1
        SOURCE-COMPUTER.    IBM-370.
        OBJECT-COMPUTER.    IBM-370.
           SKIP3
       ****************************************************************
           EJECT
        DATA DIVISION.
           SKIP1
       ****************************************************************
       *                D A T A         D I V I S I O N               *
       ****************************************************************
           SKIP3
```

```
 WORKING-STORAGE SECTION.
     SKIP1
*****************************************************************
*       W O R K I N G    S T O R A G E    S E C T I O N       *
*****************************************************************
 Ø1  MISC-DATA.
     Ø5  RESPONSE-CODE           PIC S9(8) COMP.
     Ø5  TERMINAL-MSG-LEN        PIC S9(4) COMP.
     Ø5  TERMINAL-MSG            PIC X(8Ø) VALUE SPACES.
     Ø5  WS-MAPSET-NAME          PIC X(8)    VALUE 'MAILMAP'.
     Ø5  WS-MAP-CONFIRM          PIC X(8)    VALUE 'CONFIRM'.
     Ø5  WS-MAP-ERROR            PIC X(8)    VALUE 'ERROR'.
     Ø5  WS-ABSTIME              PIC S9(16).
     Ø5  WS-ERROR-FLAG           PIC X       VALUE LOW-VALUES.
     Ø5  WS-INDEX                PIC 9(2).
*****************************************************************
*               C I C S     C O N S T A N T S                 *
*****************************************************************
     SKIP3
     COPY DFHAID.
     EJECT
     COPY DFHBMSCA.
     EJECT
*****************************************************************
*               C I C S     M A P S                           *
*****************************************************************
     SKIP3
     COPY MAILMAP.
     SKIP3
*****************************************************************
*               S E N D M A I L   A R E A                       *
*****************************************************************
     SKIP3
 Ø1  SEND-MAIL-PARMS.
    Ø5  SEND-MAIL-HEAD1.
        1Ø  SEND-RCPT              PIC X(4Ø).
        1Ø  SEND-REPLY1            PIC X(4Ø).
    Ø5  SEND-RESPONSE REDEFINES SEND-MAIL-HEAD1 PIC X.
        88  SEND-ERROR            VALUE HIGH-VALUES.
    Ø5  SEND-MAIL-HEAD2.
        1Ø  FILLER                 PIC X(4) VALUE 'TO:'.
        1Ø  SEND-TO                PIC X(4Ø).
        1Ø  FILLER                 PIC X(36) VALUE SPACES.
    Ø5  SEND-MAIL-HEAD3.
        1Ø  FILLER                 PIC X(6) VALUE 'FROM:'.
        1Ø  FILLER                 PIC X VALUE '"'.
        1Ø  SEND-FROM              PIC X(2Ø).
        1Ø  FILLER                 PIC X VALUE '"'.
        1Ø  FILLER                 PIC X VALUE SPACES.
        1Ø  FILLER                 PIC X VALUE '<'.
        1Ø  SEND-REPLY2            PIC X(4Ø) VALUE SPACES.
```

```
           10  FILLER                 PIC X VALUE '>'.
           10  FILLER                 PIC X(9) VALUE SPACES.
       05  SEND-MAIL-HEAD4.
           10  FILLER                 PIC X(9) VALUE 'SUBJECT:'.
           10  SEND-SUBJECT           PIC X(71).
       05  SEND-MAIL-TEXT OCCURS 15 TIMES.
           10  SEND-BODY              PIC X(80).
        SKIP3
*****************************************************************
        EJECT
 LINKAGE SECTION.
        EJECT
*****************************************************************
*            L I N K A G E       S E C T I O N                  *
*****************************************************************
        SKIP3
*****************************************************************
        EJECT
 PROCEDURE DIVISION.
        SKIP1
*****************************************************************
*            P R O C E D U R E      D I V I S I O N             *
*****************************************************************
        SKIP3
 0000-MAINLINE.
        SKIP1
        EXEC CICS
            RECEIVE MAP(WS-MAPSET-NAME)
                    MAPSET(WS-MAPSET-NAME)
                    INTO(MAILMAPI)
                    ASIS
                    RESP(RESPONSE-CODE)
        END-EXEC.
        IF EIBAID = DFHCLEAR OR
           EIBAID = DFHPF3 OR
           EIBAID = DFHPF15 GO TO 9999-EXIT-CICS.
        IF EIBAID = DFHPF12 OR
           EIBAID = DFHPF24
              MOVE LOW-VALUES TO MAILMAPO
              PERFORM 6000-BUILD-HEADER
              PERFORM 7000-SEND-ERASE
              GO TO 9000-RETURN-CICS.
        IF RESPONSE-CODE NOT = DFHRESP(NORMAL)
            MOVE LOW-VALUES TO MAILMAPO
            PERFORM 6000-BUILD-HEADER
            PERFORM 7000-SEND-ERASE
            GO TO 9000-RETURN-CICS.
*****************************************************************
*  HERE WE NEED TO CHECK THE LENGTH OF THE MAP INPUT FIELDS      *
*  AT A MINIMUM MAIL-TOL, MAIL-FROML AND MAIL-SUBJECTL MUST      *
*  BE GREATER THAN ZERO.  IF THEY ARE WE CAN SEND THE MESSAGE.   *
*  IF THEY ARE NOT WE MUST POSITION THE CURSOR BY MOVING -1 TO   *
```

```
*  THE APPROPRIATE L FIELD AND PERFORM SEND-OVER ROUTINE.        *
*  OTHERWISE WE CAN GO AHEAD AND MOVE THE MAP DATA TO THE        *
*  COMMAREA AND LINK TO SENDMAIL.                                *
****************************************************************
     IF MAIL-TOL = ZERO OR MAIL-TOI = SPACES THEN
         MOVE HIGH-VALUES TO WS-ERROR-FLAG
         MOVE -1 TO MAIL-TOL.
     IF MAIL-REPLYL = ZERO OR MAIL-REPLYI = SPACES THEN
         MOVE HIGH-VALUES TO WS-ERROR-FLAG
         MOVE -1 TO MAIL-REPLYL.
     IF MAIL-FROML = ZERO OR MAIL-FROMI = SPACES THEN
         MOVE HIGH-VALUES TO WS-ERROR-FLAG
         MOVE -1 TO MAIL-FROML.
     IF MAIL-SUBJECTL = ZERO OR MAIL-SUBJECTI = SPACES THEN
         MOVE HIGH-VALUES TO WS-ERROR-FLAG
         MOVE -1 TO MAIL-SUBJECTL.
     IF WS-ERROR-FLAG = HIGH-VALUES THEN
         PERFORM 6000-BUILD-HEADER
         PERFORM 7500-SEND-OVER
         GO TO 9000-RETURN-CICS.
     IF EIBAID = DFHPF9 OR EIBAID = DFHPF21 THEN
         PERFORM 4000-SEND-MAIL
         IF NOT SEND-ERROR THEN
             PERFORM 7700-SEND-CONFIRM
             GO TO 9000-RETURN-CICS
             ELSE
             PERFORM 8000-DUMP-TASK
             PERFORM 7800-SEND-ERROR
             GO TO 9000-RETURN-CICS
         ELSE
         PERFORM 6000-BUILD-HEADER
         MOVE -1 TO MAIL-BODYL(1)
         PERFORM 7500-SEND-OVER
         GO TO 9000-RETURN-CICS.
 4000-SEND-MAIL.
     MOVE MAIL-TOI TO SEND-RCPT SEND-TO.
     MOVE MAIL-REPLYI TO SEND-REPLY1 SEND-REPLY2.
     MOVE MAIL-FROMI TO SEND-FROM.
     MOVE MAIL-SUBJECTI TO SEND-SUBJECT.
     PERFORM 4000-MOVE-BODY VARYING WS-INDEX FROM 1 BY 1
         UNTIL WS-INDEX IS GREATER THAN 15.
     EXEC CICS LINK PROGRAM('SENDMAIL')
         COMMAREA(SEND-MAIL-PARMS)
         END-EXEC.
 4000-MOVE-BODY.
     MOVE MAIL-BODYI(WS-INDEX) TO SEND-BODY(WS-INDEX).
 6000-BUILD-HEADER.
     EXEC CICS    ASSIGN
                  APPLID(MAIL-APPLIDO)
                  USERID(MAIL-USERIDO)
                  USERNAME(MAIL-FROMO)
                  END-EXEC.
```

```
    EXEC CICS    ASKTIME
                     ABSTIME    (WS-ABSTIME)
                 END-EXEC.
    EXEC CICS    FORMATTIME
                     ABSTIME    (WS-ABSTIME)
                     MMDDYYYY   (MAIL-DATEO)
                     DATESEP
                     TIME       (MAIL-TIMEO)
                     TIMESEP
                 END-EXEC.
7000-SEND-ERASE.
    EXEC CICS    SEND
                     MAP       (WS-MAPSET-NAME)
                     MAPSET    (WS-MAPSET-NAME)
                     FROM      (MAILMAPO)
                     ERASE
                 END-EXEC.
7500-SEND-OVER.
    EXEC CICS    SEND
                     MAP       (WS-MAPSET-NAME)
                     MAPSET    (WS-MAPSET-NAME)
                     FROM      (MAILMAPO)
                     CURSOR
                 END-EXEC.
7700-SEND-CONFIRM.
    EXEC CICS    SEND
                     MAP       (WS-MAP-CONFIRM)
                     MAPSET    (WS-MAPSET-NAME)
                     MAPONLY
                 END-EXEC.
7800-SEND-ERROR.
    EXEC CICS    SEND
                     MAP       (WS-MAP-ERROR)
                     MAPSET    (WS-MAPSET-NAME)
                     MAPONLY
                 END-EXEC.
8000-DUMP-TASK.
    EXEC CICS DUMP DUMPCODE('ERR1')
                 END-EXEC.
9000-RETURN-CICS.
    EXEC CICS
              RETURN TRANSID(EIBTRNID)
    END-EXEC.
9999-EXIT-CICS.
    MOVE +80 TO TERMINAL-MSG-LEN.
    MOVE ' TRANSACTION TERMINATED / PRESS CLEAR TO CONTINUE'
      TO TERMINAL-MSG.
    EXEC CICS
              SEND FROM(TERMINAL-MSG) LENGTH(TERMINAL-MSG-LEN)
                   ERASE LAST
    END-EXEC.
    EXEC CICS
```

```
                RETURN
           END-EXEC.
           GOBACK.
```

SENDMAIL.ASM

```
TITLE 'S E N D M A I L  TCPSMTP MAIL INTERFACE'
*********************************************************************
*     S E N D M A I L  THIS PROGRAM IS USED TO SEND E-MAIL FROM CICS  *
*     USING THE CICS SPOOL INTERFACE.                                 *
*     COMMAREA FORMAT:                                                *
*         4Ø BYTES RECIPIENT EMAIL ADDRESS                            *
*         4Ø BYTES SENDER EMAIL ADDRESS                               *
*         8Ø BYTES TO:                                                *
*         8Ø BYTES FROM:                                              *
*         8Ø BYTES SUBJECT:                                           *
*         8Ø BYTES MESSAGE TEXT                                       *
*            ....                                                     *
*         8Ø BYTES END OF MESSAGE LINES                               *
*    NOTE THAT THE COMMAREA MUST BE A MULTIPLE OF 8Ø CHARACTERS       *
*    AND BE A MINIMUM OF 5*8Ø OR 4ØØ BYTES LONG AND A MAXIMUM OF      *
*    32K (CICS COMMAREA LIMIT).                                       *
*    RETURN CODES:                                                    *
*         FF  ERROR                                                   *
*    EXAMPLE:                                                         *
*     CA       DS   ØD                                               *
*     CARCPT   DC   CL4Ø'DGRINSELL@STATE.MT.US'                      *
*     CAFROM   DC   CL4Ø'CZØØ55@HLNCTR.STATE.MT.US'                  *
*              ORG  CA                                               *
*     CARESP   DC   XL1'ØØ'                                          *
*              ORG                                                   *
*     CAMSG1   DC   CL68'TO:  DONALD GRINSELL'                       *
*     CAMSG2   DC   CL8Ø'FROM: "CZØØ55 ON CICS"<DGRINSELL@STATE.MT.US>*
*     CAMSG3   DC   CL8Ø'SUBJECT: MAIL FROM CICS'                    *
*     CAMSG4   DC   CL8Ø' '                                          *
*     CAMSG5   DC   CL8Ø'THIS IS MY MESSAGE'                         *
*     CALEN    DC   Y(*-CA)                                          *
*              EXEC CICS LINK PROGRAM('SENDMAIL')                    *
*                        COMMAREA(CA) LENGTH(CALEN)                  *
*              CLI  CARESP,X'FF'    Q-ERROR RETURN CODE              *
*              BE   SENDFAIL        BIF YES, ELSE CONTINUE           *
*********************************************************************
         EJECT                                                       *
*********************************************************************
```

Editor's note: this article will be concluded in the next issue.

Donald A Grinsell
CICS Systems Programmer
State of Montana (USA)

IBM has tweaked its XML Toolkit for OS/390 with the XML Parser for OS/390, Java Edition and the XML Parser for OS/390, C++ Edition, both based on cross-platform open-source code and compliant with industry standards.

The toolkit, we're told, reduces development time by providing tested functions, promotes code re-use, and supports ASCII, EBCDIC, and Unicode data.

It is positioned to complement WebSphere Application Server for OS/390, Developer Kit for OS/390, and MQSeries.

The OS/390 XML Parser Java Edition can work with CICS TS 1.1 by exploiting the transaction system's ability to run a Java Virtual Machine within itself.

Interoperability with CICS TS 1.3 is supported via the CICS Transaction Gateway 3.1 which transmits application requests to CICS.

For further information contact your local IBM representative.
URL: http://s390.ibm.com/xml.

* * *

Dynasty Technologies has announced availability of its native support for System/390 systems via Enterprise SpA, Dynasty's development partner for System/390 platforms, which has ported its existing Dynasty foreign exchange application to a native CICS and DB2 Universal Database.

This adds to the ability to deploy application servers on NT, Windows 2000, Sun Solaris, HP, RS6000, NonStop Himalaya, and AS/400 servers.

Enterprise SpA is an Italian banking software house, which ported its PREMIA foreign exchange application.

For further information contact:
Dynasty Technologies, 101 Redwood Shores Parkway, #200 Redwood shores, CA 94065, USA.
Tel: (650) 631 5889.
URL: http://www.dynasty.com.

* * *

IBM has announced its Developer Kit for OS/390, Java 2 Technology Edition, JDK 1.3 environment for System/390s. Besides Java 2 APIs, additional functions include Java Record I/O (JRIO) and security API facilities. It runs on System/390s running OS/390 Version 2.8 (or later) with floating point instructions.

JVM implementations are intended to allow applications written in Java for OS/390 subsystems such as CICS TS, IMS, and DB2 stored procedures. IBM says it plans to add these new technologies to the existing base JVM, while maintaining the Java compatibility of existing applications in Java deployed on OS/390.

For further information contact your local IBM representative.
URL: http://www.alphaworks.ibm.com/aw.nsf/textOverviews/8638C67170C0ED5D882568DD0077C8FF.