203

CICS update

October 2002

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 December 1999 issue, are available separately to subscribers for £16.00 ($24.00) each including postage.

CICS Update on-line

Code from *CICS Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at http://www.xephon.com/cics; you will need to supply a word from the printed issue.

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

When Xephon is given copyright, 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 download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

Printed in England.

How to e-mail QMF reports (as attachments) from a QMF session

NOTE

In this article I will use the abbreviations CICS, TSO, and CMS, and refer to CICS, TSO, and CMS commands from QMF.

For more details about the CICS, TSO, and CMS commands see the *QMF Reference Guide*.

INTRODUCTION

Although QMF is a powerful tool for extracting data from DB2, it lacks some great features like e-mailing a report. One solution is QMF for Windows, but you have to install it on every PC on which you plan to use it. Another solution is to design and test your query, run it in batch, transfer the report on your workstation, and e-mail it. This could be very tedious.

There is a better solution – send the report in an e-mail from your QMF session without leaving it. It works very well in CICS, even when using the Callable Interface of QMF. I have not tested this solution in TSO or in CMS, but I think it should also be possible, without too much efforts, in those environments.

THE PRINCIPLE

To send your report directly in an e-mail from a QMF session in CICS, you have to carry out the following.

Set some global variables in QMF

You must set the global variable Q to a quote. To do this, use the following syntax:

```
SET GLOBAL (Q = (')
```

If you want to, you can also set the recipient address in another global variable. This could be very useful if you have a lot of QMF reports to send.

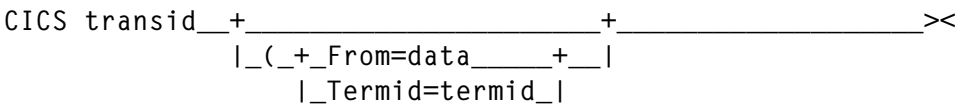
Print the report in a temporary storage queue (TS)

It is quite simple to print the report in a temporary storage queue using the QMF command PRINT. The only problem is the queue name. I have chosen the user ID although you can use any other value. In QMF, the user ID is set in the global variable DSQAO_CONNECT_ID. So the PRINT command looks like:

```
PRINT REPORT (QUEUENAME=&DSQAO_CONNECT_ID QUEUETYPE=TS W=79 SUSPEND=YES
```

Start a transaction from QMF

To start a transaction from QMF, use the command CICS. According to the *QMF Reference Guide*, the syntax is:



where:

- *transid* is the CICS transaction to be started.
- *From=data* is the data to be used. The value for data is limited to 55 characters.
- *Termid=termid* is the associated terminal. If you omit the TERMID parameter, no terminal is associated with your transaction.

If you are using a global variable in the FROM parameter, you must surround the global variable with another global variable that has a value of ' (single quote).

As you can see, you can pass only 55 characters to the started transaction. I set it to the recipient's e-mail address and it looks like:

```
CICS QMFS (FROM=&Q&EMAIL&Q
```

The started transaction runs a program that:

- Retrieves the parameter from QMF.

You will get the recipient's e-mail address using the CICS statement RETRIEVE. You have to remove all the low-values in the address.

- Submits a job that sends the e-mail.

  Our CICS administrator has defined a transient data queue (TD) that points to an internal reader. So, to submit a job, you must first open the TD queue and then write the JCL cards to it. The job is submitted when the TD is closed.

  To send the QMF report in an e-mail, you could use the TCP/IP SMTPNOTE command in batch mode and write the report printed in the TS queue as the body of the note. But I prefer XMITIP, a great freeware program written by Lionel B Dyck.

  XMITIP is a mainframe-based electronic mail application that is capable of sending electronic mail to any valid intranet or Internet address. Along with messages, XMITIP can also send mainframe files in one of several different file-attachment formats. The recipients can be on any mail system that connects to the Internet (the world) or intranet (in-house). The Simple Mail Transport Protocol (SMTP) is used for sending the mail with datasets attached using the appropriate SMTP statements.

  XMITIP can be executed as a step within a batch job, under TSO as a command, or under ISPF using a robust ISPF interface. It can also be used within an automated operations tool to generate messages related to system events.

  Some of the features of XMITIP are:

  – Send electronic mail to one or more addresses.

  – Send one or more datasets as file attachments in one of the following formats: plain text, HTML, Rich Text Format (RTF), Portable Document Format (PDF), Comma Separated Value (CSV), TSO Transmit (XMIT), and binary.

  – Supports address lists.

  – Supports CC and BCC.

– Supports priority, sensitivity, and importance.

More information is available at http://www.lbdsoftware.com/xmitip.html.

- Delete the TS queue.

  After the job submission, it's a good idea to delete the TS queue. If you don't, reports printed successively will be appended in the queue.

A QMF/XMITIP JOB EXAMPLE

Here is some sample code for a job that will send a QMF report by e-mail:

```
//USERIDX JOB USERIDX,ABC,CLASS=A,MSGCLASS=A,REGION=ØM
//XMITIP EXEC PGM=IKJEFT1B,DYNAMNBR=5Ø,COND=(Ø,NE)
//STEPLIB  DD DISP=SHR,DSN=hlq.PKZIP.LOADLIB
                                        - the name of the zip library
//SYSEXEC  DD DISP=SHR,DSN=hlq.XMITIP.CMDPROC
                                     - the name of the XMITIP library
//SYSTCPD  DD DSN=TCPIP.xxx.TCPPARMS(TCPDATA1),
                                   - the name of the TCP/IP parms dataset
//      DISP=SHR
//SYSPRINT DD SYSOUT=
//SYSTSPRT DD SYSOUT=
//SYSTSIN DD
%xmitip (your.address@your.site)
                                    - the e-mail address of the recipient
Msgdd MSG                                      - the body of the message
From userid@your.site +                      - the sender of the message
Subject 'QMF report' +
FileDD (QMFREP) +
Filename (qmfreport.pdf) +
                          - the name of QMF report attached in the e-mail
Filedesc (report_from_QMF) +
Format (PDF/Landscape/8/A4/1Ø)
                       - the format of QMF report attached in the e-mail
//MSG DD
Put the body of the message here
//QMFREP DD
Put the QMF report here
```

THE 80 CHARACTER BARRIER

The internal reader is only 80 characters wide, so to print a wider QMF report requires the report to be cut into 80-character parts, and to reblock it using IEBGENER as follows:

```
//COPY    EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=
//SYSIN    DD DUMMY
//SYSUT1   DD                                  -input file. LRECL = 80
Put your QMF report here
//SYSUT2   DD DSN=&&TEMP,DISP=(NEW,PASS),UNIT=3390,
                                 - output file. LRECL= 80, BLKSIZE=2400
//        DCB=(LRECL=80,BLKSIZE=2400,RECFM=FB),
//         SPACE=(CYL,(1,1))
//REBLOCK EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=&&TEMP,DISP=(OLD,PASS),
                                   - input file. LRECL= 80, BLKSIZE=2400
//         DCB=(LRECL=240,BLKSIZE=2400,RECFM=FB)
//SYSUT2   DD DSN=&&QMFDATA,DISP=(NEW,PASS),
                                 - output file. LRECL= 240, BLKSIZE=2400
//         DCB=(LRECL=240,BLKSIZE=2400,RECFM=FBA),
//         SPACE=(CYL,(1,1)),UNIT=3390
```

SENDING QMF REPORTS FROM TSO OR CMS

In TSO or in CMS, the print command is replaced by the EXPORT command. And it's even easier than in CICS. For example in TSO, you can write in your QMF procedure:

```
EXPORT REPORT TO USERID.QMF.DATASET
TSO SUBMIT USERID.JCL(XMITQMF)
```

USING THE CALLABLE INTERFACE

When you call QMF from, for example, a COBOL program, you can limit users' access to QMF by using report mini-sessions. In a report mini-session, QMF limits the commands that a user can issue while viewing a report. A report mini-session behaves as a nested session (a session within a session). In mini-sessions, your initial QMF session remains intact, but becomes temporarily unavailable while you are viewing a report. The mini-session becomes your current, active

session until you issue the END command (or press the *End* function key).

One of the commands that doesn't run in a QMF mini-session is RUN PROC. So, if as I did, you have written a QMF procedure to send the report by e-mail, you can't use it in a report mini-session. A solution could be to ask the user if he wants the report in his e-mail, before calling the callable interface. In that case, you have to set the global variable Q and run the procedure in the calling program.

CONCLUSIONS

Sending QMF reports directly to an e-mail is very useful. But this feature may also be integrated in your own applications. You can write the data to a TS queue and then start a task that will read the queue and submit an e-mail job with the data. Using an e-mail to fax converter, you can even send your data by fax.

But you have to be careful not to send reports that are too big, particularly in CICS. It take time and resources to print the report in a TS queue. If you want to send big reports in an e-mail, it's better to work in batch.

Pierre Delaunoy
Director
Ministère de la Communauté Française (Belgium)

Opening and closing files

We needed to give certain users the ability to open and close selected groups of files without granting them CEMT access. For this purpose, we developed a program called OPENCLOS and assigned it to transaction FCTL.

This program issues EXEC CICS SET FILE OPEN or CLOSE commands.

To control which users can command which files, we have a VSAM dataset as an information base. Each user is authorized to open or close a group of files. We can define as many file groups and as many users as we like.

The VSAM dataset is a KSDS with a keylength of 22 (the total amount of information in each line) and a logical record length of 80 bytes, in order to make it easy to REPRO its contents from an ordinary editable PDS member or sequential file. The following is an example of the file contents:

```
1OPEN *       USER0700
1OPEN GROUP1  USER0001
1OPEN GROUP2  USER0001
1OPEN GROUP3  USER0005
1CLOSE*       USER0700
1CLOSEGROUP1  USER0001
1CLOSEGROUP2  USER0001
1CLOSEGROUP3  USER0005
2OPEN GROUP1  FILE001
2OPEN GROUP1  FILE002
2OPEN GROUP1  FILE003
2OPEN GROUP1  FILE004
2OPEN GROUP2  FILE7
2OPEN GROUP3  FILE0015
2CLOSEGROUP1  FILE001
2CLOSEGROUP1  FILE002
2CLOSEGROUP1  FILE003
2CLOSEGROUP1  FILE004
2CLOSEGROUP2  FILE7
2CLOSEGROUP3  FILE0015
```

There are two types of record, indicated by the first byte.

Type 1 controls each user's authorizations. For example, user USER0700 can open or close any group of files. User USER0001 can open or close GROUP1 and GROUP2 files.

Type 2 specifies which files belong to an open or close group. In this example, GROUP1 contains four files (both for open and for close), GROUP2 and GROUP3 contain one file each. In this example, all the open and close groups are identical, both in file and in user specifications, but this need not be so. You can have a group only for open, or you can have a user who can only open a group, but cannot close it.

Basically, each line represents a command that is checked when a user

issues it, as I will explain below. Each group name can have up to eight characters (as well as the file name, of course, since it represents the name of the file under CICS). The example above shows also how the file should be constructed, with the group name occupying eight characters, padded with spaces, if necessary, and the OPEN or CLOSE directives occupying five bytes.

If you build the file image with an editor, as we do, remember to sort its contents in ascending order so that the VSAM keys are also in the correct sequence when you REPRO it.

This file must be declared under CICS with read and browse capability, and its CICS name declared in the first level-77 variable of the COBOL program. If you forget to declare it browsable (under its CEDA definition), an INVREQ condition will occur when you run the transaction.

There is no BMS map associated with the program. All terminal input is done on a line following the transaction name, and the output carries some 3270 code for positioning and also to add a little colour.

The program should be compiled to CICS and assigned a transaction. We use transaction FCTL. All commands are written after the transaction name, with a single space separating each word. You have several options at your disposal. If you simply enter the transaction name, you receive a list of the available commands.

Valid commands are:

- DISPLAY – displays existing groups.
- DISPLAY *group* – displays files in a group.
- OPEN *group* – opens files in a group.
- CLOSE *group* – closes files in a group.
- STATUS *group* – shows the status of files in a group.

If you type FCTL DISPLAY, you get a list of the group names. This list refers to the OPEN lines. If you have groups with only CLOSE commands, they will not be listed.

With FCTL DISPLAY *group*, you get the list of the files assigned to that

group. Once again, this refers to the OPEN lines only.

To open or close files, issue FCTL OPEN *group* or FCTL CLOSE *group*. The result of each command is shown below:

```
FCTL OPEN GROUP1

 FILE001   OPEN     Command complete
 FILE002   OPEN     Command complete
 FILE003   OPEN     File not found
 FILE004   OPEN     Command complete
```

You can also check the status of the files. For example:

```
FCTL STATUS GROUP1

 FILE001  STATUS   Opened       Enabled
 FILE002  STATUS   Opened       Enabled
 FILE003  STATUS   File not found
 FILE004  STATUS   Opened       Enabled
```

This way, you can check which files are opened or closed, either before or after issuing a command.

To check the status, or display the files in a group, your user needs to have OPEN authority for that group. If they do not have the authority to perform an operation, they receive a “User not authorized” message.

OPENCLOS

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. OPENCLOS.
       ENVIRONMENT DIVISION.
       DATA DIVISION.
      *============================================================*
       WORKING-STORAGE SECTION.
      *======================*
      *
       77  FNAME             PIC X(8)  VALUE 'VSAMFILE'.
       77  RDUMMYL           PIC S9(4) COMP VALUE +1.
       77  RECEIVE-AREA-L    PIC S9(4) COMP VALUE +21.
       77  FILE-KEY-L        PIC S9(4) COMP VALUE +19.
       77  OUT-MSG-L0        PIC S9(4) COMP VALUE +61.
       77  OUT-MSG-L1        PIC S9(4) COMP VALUE +85.
       77  OUT-MSG-L2        PIC S9(4) COMP VALUE +61.
       77  FILE-OPE          PIC S9(8) COMP VALUE +0.
       77  FILE-ENA          PIC S9(8) COMP VALUE +0.
       77  FILE-RC1          PIC S9(8) COMP VALUE +0.
       77  FILE-RC2          PIC S9(8) COMP VALUE +0.
```

```
 77  X                  PIC S9(8) COMP VALUE +Ø.
 77  FICODEANT         PIC X(8)  VALUE SPACES.
 77  RDUMMY            PIC X.
 77  HELPTEXTØ         PIC X(52) VALUE
     'Invalid or null command entered. Valid commands are:'.
 77  HELPTEXT1         PIC X(52) VALUE
     'DISPLAY       - Dispalys existing groups           '.
 77  HELPTEXT2         PIC X(52) VALUE
     'DISPLAY group - Displays files in a group          '.
 77  HELPTEXT3         PIC X(52) VALUE
     'OPEN group    - Opens files in a group             '.
 77  HELPTEXT4         PIC X(52) VALUE
     'CLOSE group   - Closes files in a group            '.
 77  HELPTEXT5         PIC X(52) VALUE
     'STATUS group  - Shows the status of files in a group'.
*
 Ø1  OUT-POS.
     Ø2 OUTPOS1.
       Ø4  FILLER PIC X(2) VALUE X'41D1'.
       Ø4  FILLER PIC X(2) VALUE X'42E1'.
       Ø4  FILLER PIC X(2) VALUE X'43F1'.
       Ø4  FILLER PIC X(2) VALUE X'45C1'.
       Ø4  FILLER PIC X(2) VALUE X'46D1'.
       Ø4  FILLER PIC X(2) VALUE X'47E1'.
       Ø4  FILLER PIC X(2) VALUE X'48F1'.
       Ø4  FILLER PIC X(2) VALUE X'4AC1'.
       Ø4  FILLER PIC X(2) VALUE X'4BD1'.
       Ø4  FILLER PIC X(2) VALUE X'4CE1'.
       Ø4  FILLER PIC X(2) VALUE X'4DF1'.
       Ø4  FILLER PIC X(2) VALUE X'4FC1'.
       Ø4  FILLER PIC X(2) VALUE X'5ØD1'.
       Ø4  FILLER PIC X(2) VALUE X'51E1'.
       Ø4  FILLER PIC X(2) VALUE X'52F1'.
       Ø4  FILLER PIC X(2) VALUE X'54C1'.
       Ø4  FILLER PIC X(2) VALUE X'55D1'.
       Ø4  FILLER PIC X(2) VALUE X'56E1'.
       Ø4  FILLER PIC X(2) VALUE X'57F1'.
       Ø4  FILLER PIC X(2) VALUE X'59C1'.
       Ø4  FILLER PIC X(2) VALUE X'5AD1'.
     Ø2 OUTPOS REDEFINES OUTPOS1 PIC X(2) OCCURS 21.
*
 Ø1  OUT-MESSAGE.
     Ø2 FILLER    PIC X(1)  VALUE X'11'.
     Ø2 OUT-ADDR  PIC X(2)  VALUE X'4ØCØ'.
     Ø2 FILLER    PIC X(3)  VALUE X'2842F7'.
     Ø2 OUT-FILE  PIC X(9)  VALUE SPACES.
     Ø2 FILLER    PIC X(3)  VALUE X'2842F2'.
     Ø2 OUT-CMD   PIC X(9)  VALUE SPACES.
     Ø2 FILLER    PIC X(2)  VALUE X'2842'.
     Ø2 OUT-COLOR PIC 9     VALUE 3.
```

```
     Ø2 OUT-MSG   PIC X(55) VALUE SPACES.
*
 Ø1  OUT-MESSAGEØ.
     Ø2 FILLER    PIC X(3)  VALUE X'1140C2'.
     Ø2 FILLER    PIC X(3)  VALUE X'2842F5'.
     Ø2 OUT-MSGØ  PIC X(55) VALUE SPACES.
*
 Ø1  OUT-MESSAGE2.
     Ø2 FILLER    PIC X(3)  VALUE X'115BE5'.
     Ø2 FILLER    PIC X(3)  VALUE X'2842F6'.
     Ø2 OUT-MSG2  PIC X(55) VALUE SPACES.
*
 Ø1  RECEIVE-AREA.
     Ø2    FILLER          PIC X(5).
     Ø2    RV-CMDOPEN.
       Ø4  RV-OPEN         PIC X(4).
       Ø4  FILLER          PIC X.
       Ø4  RV-OPENCODE     PIC X(8).
       Ø4  FILLER          PIC X(3).
     Ø2    RV-CMDCLOSE  REDEFINES RV-CMDOPEN.
       Ø4  RV-CLOSE        PIC X(5).
       Ø4  FILLER          PIC X.
       Ø4  RV-CLOSECODE    PIC X(8).
       Ø4  FILLER          PIC X(2).
     Ø2    RV-CMDSTATUS REDEFINES RV-CMDOPEN.
       Ø4  RV-STATUS       PIC X(6).
       Ø4  FILLER          PIC X.
       Ø4  RV-STATUSCODE   PIC X(8).
       Ø4  FILLER          PIC X.
     Ø2    RV-CMDDISP   REDEFINES RV-CMDOPEN.
       Ø4  RV-DISPLAY      PIC X(7).
       Ø4  FILLER          PIC X.
       Ø4  RV-DISPLAYCODE  PIC X(8).
     Ø2    FILLER          PIC X(2Ø).
*
 Ø1  FILE-KEY.
     Ø2    FK-TYPE         PIC X.
     Ø2    FK-COMMAND      PIC X(5).
     Ø2    FK-CODE         PIC X(8).
     Ø2    FK-TYPE1.
       Ø4  FK-USERNAME     PIC X(8).
     Ø2    FK-TYPE2 REDEFINES FK-TYPE1.
       Ø4  FK-FILENAME     PIC X(8).
*
 Ø1  FILE-AREA.
     Ø2    FI-TYPE         PIC X.
     Ø2    FI-COMMAND      PIC X(5).
     Ø2    FI-CODE         PIC X(8).
     Ø2    FI-TYPE1.
       Ø4  FI-USERNAME     PIC X(8).
```

```
       Ø4  FILLER          PIC X(36).
     Ø2    FI-TYPE2 REDEFINES FI-TYPE1.
       Ø4  FI-FILENAME     PIC X(44).
     Ø2    FILLER          PIC X(26).
*
*================================================================*
 PROCEDURE DIVISION.
*================================================================*
*
     MOVE SPACES TO FILE-AREA
                    FILE-KEY
                    RECEIVE-AREA
     EXEC CICS IGNORE CONDITION LENGERR INVREQ
     END-EXEC
     EXEC CICS RECEIVE INTO   (RECEIVE-AREA)
                       LENGTH (RECEIVE-AREA-L)
     END-EXEC
*
     EVALUATE TRUE
       WHEN RV-OPEN = 'OPEN'
            MOVE 'OPEN ' TO FK-COMMAND OUT-CMD
            MOVE RV-OPENCODE TO FK-CODE FICODEANT
            GO TO OPEN-CLOSE
       WHEN RV-CLOSE = 'CLOSE'
            MOVE 'CLOSE' TO FK-COMMAND OUT-CMD
            MOVE RV-CLOSECODE TO FK-CODE FICODEANT
            GO TO OPEN-CLOSE
       WHEN RV-STATUS = 'STATUS'
            MOVE 'OPEN' TO FK-COMMAND
            MOVE 'STATUS' TO OUT-CMD
            MOVE 'OPEN  ' TO RV-STATUS
            MOVE RV-STATUSCODE TO FK-CODE FICODEANT
            GO TO OPEN-CLOSE
       WHEN RV-DISPLAY = 'DISPLAY'
            MOVE 'OPEN' TO FK-COMMAND
            MOVE RV-DISPLAYCODE TO FK-CODE
            GO TO DISPLAY-CODES
       WHEN OTHER
            MOVE 1 TO OUT-COLOR
            MOVE HELPTEXTØ TO OUT-MSGØ
            MOVE HELPTEXT1 TO OUT-MSG
            PERFORM SEND-MSG1
            MOVE HELPTEXT2 TO OUT-MSG
            PERFORM SEND-MSG1
            MOVE HELPTEXT3 TO OUT-MSG
            PERFORM SEND-MSG1
            MOVE HELPTEXT4 TO OUT-MSG
            PERFORM SEND-MSG1
            MOVE HELPTEXT5 TO OUT-MSG
            PERFORM SEND-MSG1
```

```
            GO TO RETURN-TO-CICS
     END-EVALUATE.
*
*================================================================*
*
 OPEN-CLOSE.
*===========*
     MOVE 1 TO FK-TYPE
     MOVE '*' TO FK-USERNAME
     PERFORM READ-FILE
     IF FILE-RC2 NOT EQUAL Ø
        EXEC CICS ASSIGN USERID (FK-USERNAME)
        END-EXEC
        PERFORM READ-FILE
        IF FILE-RC2 NOT EQUAL Ø
           MOVE '* User not authorized *' TO OUT-MSGØ
           PERFORM SEND-MSGØ
           GO TO RETURN-TO-CICS
        END-IF
     END-IF

     MOVE 2 TO FK-TYPE
     MOVE 22 TO FILE-KEY-L
     MOVE SPACES TO FK-USERNAME
     EXEC CICS STARTBR FILE     (FNAME)
                       RIDFLD   (FILE-KEY)
                       KEYLENGTH(FILE-KEY-L)
                       RESP     (FILE-RC1)
                       RESP2    (FILE-RC2)
     END-EXEC.
*
 LOOP-FILE-OPEN-CLOSE.
*====================*
     EXEC CICS READNEXT FILE   (FNAME)
                        RIDFLD (FILE-KEY)
                        INTO   (FILE-AREA)
                        RESP   (FILE-RC1)
                        RESP2  (FILE-RC2)
     END-EXEC
     IF FILE-RC2   NOT EQUAL Ø
     OR FI-CODE    NOT EQUAL FICODEANT
     OR FI-COMMAND NOT EQUAL RV-CLOSE
        MOVE '>> PROGRAM TERMINATED <<' TO OUT-MSG2
        PERFORM SEND-MSG2
        GO TO RETURN-TO-CICS
     END-IF

     IF OUT-CMD = 'OPEN'
        PERFORM OPEN-FILE
     END-IF
```

```
     IF OUT-CMD = 'CLOSE'
        PERFORM CLOSE-FILE
     END-IF
     IF OUT-CMD = 'STATUS'
        PERFORM STATUS-FILE
     END-IF

     MOVE FI-FILENAME TO OUT-FILE
     PERFORM SEND-MSG1
     GO TO LOOP-FILE-OPEN-CLOSE.
*
*================================================================*
*
 DISPLAY-CODES.
*==============*
     IF RV-DISPLAYCODE EQUAL SPACES OR LOW-VALUES
        MOVE ' * DISPLAY EXISTING GROUPS *' TO OUT-MSGØ
     ELSE
        MOVE ' * * DISPLAY GROUP FILES * *' TO OUT-MSGØ
     END-IF
     PERFORM SEND-MSGØ
     MOVE 2 TO FK-TYPE
     MOVE SPACES TO OUT-CMD OUT-MSG
     MOVE 14 TO FILE-KEY-L
     EXEC CICS STARTBR FILE     (FNAME)
                       RIDFLD   (FILE-KEY)
                       KEYLENGTH(FILE-KEY-L)
                       GENERIC
                       RESP     (FILE-RC1)
                       RESP2    (FILE-RC2)
     END-EXEC.
*
 LOOP-FILE-DISPLAY.
*==================*
     EXEC CICS READNEXT FILE   (FNAME)
                        RIDFLD (FILE-KEY)
                        INTO   (FILE-AREA)
                        RESP   (FILE-RC1)
                        RESP2  (FILE-RC2)
     END-EXEC

     IF FILE-RC2   NOT EQUAL Ø
     OR FI-COMMAND NOT EQUAL FK-COMMAND
        MOVE '>> PROGRAM TERMINATED <<' TO OUT-MSG2
        PERFORM SEND-MSG2
        GO TO RETURN-TO-CICS
     END-IF

     IF RV-DISPLAYCODE EQUAL SPACES OR LOW-VALUES
        IF FI-CODE EQUAL FICODEANT
```

```
            GO TO LOOP-FILE-DISPLAY
         END-IF
         MOVE FI-CODE TO FICODEANT OUT-CMD
      ELSE
         IF FI-CODE NOT EQUAL RV-DISPLAYCODE
            MOVE '>> PROGRAM TERMINATED <<' TO OUT-MSG2
            PERFORM SEND-MSG2
            GO TO RETURN-TO-CICS
         END-IF
         MOVE FI-CODE     TO OUT-CMD
         MOVE FI-FILENAME TO OUT-MSG
      END-IF

      PERFORM SEND-MSG1
      GO TO LOOP-FILE-DISPLAY.
*
*===============================================================*
*
 SEND-MSG1.
*=========*
      ADD 1 TO X
      MOVE OUTPOS(X) TO OUT-ADDR
      IF X = 1
         PERFORM SEND-MSGØ
      END-IF
      PERFORM SEND-NORMAL
      IF X > 2Ø
         MOVE Ø TO X
         MOVE '> press any key to continue <' TO OUT-MSG2
         PERFORM SEND-MSG2
         EXEC CICS RECEIVE INTO   (RDUMMY)
                           LENGTH (RDUMMYL)
         END-EXEC
      END-IF.
*
 SEND-MSGØ.
*=========*
      EXEC CICS SEND FROM   (OUT-MESSAGEØ)
                     LENGTH (OUT-MSG-LØ)
                     ERASE
      END-EXEC.
*
 SEND-MSG2.
*=========*
      EXEC CICS SEND FROM   (OUT-MESSAGE2)
                     LENGTH (OUT-MSG-L2)
      END-EXEC.
*
 SEND-NORMAL.
*===========*
```

```
     EXEC CICS SEND FROM   (OUT-MESSAGE)
                    LENGTH (OUT-MSG-L1)
     END-EXEC.
*
 READ-FILE.
*=========*
     EXEC CICS READ FILE   (FNAME)
                    RIDFLD (FILE-KEY)
                    INTO   (FILE-AREA)
                    RESP   (FILE-RC1)
                    RESP2  (FILE-RC2)
     END-EXEC.
*
 OPEN-FILE.
*=========*
     EXEC CICS SET FILE  (FI-FILENAME)
                   OPEN
                   ENABLED
                   RESP  (FILE-RC1)
                   RESP2 (FILE-RC2)
     END-EXEC
     PERFORM CHECK-RETURNCODE.
*
 CLOSE-FILE.
*==========*
     EXEC CICS SET FILE  (FI-FILENAME)
                   CLOSED
                   DISABLED
                   RESP  (FILE-RC1)
                   RESP2 (FILE-RC2)
     END-EXEC
     PERFORM CHECK-RETURNCODE.
*
 STATUS-FILE.
*===========*
     EXEC CICS INQUIRE FILE (FI-FILENAME)
               OPENSTATUS   (FILE-OPE)
               ENABLESTATUS (FILE-ENA)
               RESP         (FILE-RC1)
               RESP2        (FILE-RC2)
     END-EXEC
     MOVE SPACES TO OUT-MSG
     MOVE 1 TO OUT-COLOR
     IF FILE-RC2 NOT EQUAL Ø
        PERFORM CHECK-RETURNCODE
     ELSE
        EVALUATE TRUE
          WHEN FILE-OPE = DFHVALUE(OPEN)
               MOVE 'Opened      ' TO OUT-MSG(1:12)
          WHEN FILE-OPE = DFHVALUE(CLOSED)
```

```
                MOVE 'Closed      ' TO OUT-MSG(1:12)
              WHEN FILE-OPE = DFHVALUE(OPENING)
                MOVE 'Opening     ' TO OUT-MSG(1:12)
              WHEN FILE-OPE = DFHVALUE(CLOSING)
                MOVE 'Closing     ' TO OUT-MSG(1:12)
              WHEN FILE-OPE = DFHVALUE(CLOSEREQUEST)
                MOVE 'Clo. request' TO OUT-MSG(1:12)
           END-EVALUATE
           EVALUATE TRUE
             WHEN FILE-ENA = DFHVALUE(ENABLED)
                MOVE 'Enabled  '    TO OUT-MSG(14:9)
             WHEN FILE-ENA = DFHVALUE(DISABLED)
                MOVE 'Disabled '    TO OUT-MSG(14:9)
             WHEN FILE-ENA = DFHVALUE(UNENABLED)
                MOVE 'Unenabled'    TO OUT-MSG(14:9)
             WHEN FILE-ENA = DFHVALUE(DISABLING)
                MOVE 'Disabling'    TO OUT-MSG(14:9)
           END-EVALUATE
        END-IF.
      *
       CHECK-RETURNCODE.
      *================*
        EVALUATE TRUE
           WHEN FILE-RC1 = DFHRESP(NORMAL)
                MOVE 'Command complete' TO OUT-MSG
                MOVE 1 TO OUT-COLOR
           WHEN FILE-RC1 = DFHRESP(FILENOTFOUND)
                MOVE 'File not found' TO OUT-MSG
                MOVE 6 TO OUT-COLOR
           WHEN FILE-RC1 = DFHRESP(INVREQ)
                MOVE 'INVREQ' TO OUT-MSG
                MOVE 4 TO OUT-COLOR
           WHEN FILE-RC1 = DFHRESP(IOERR)
                MOVE 'IOERR' TO OUT-MSG
                MOVE 4 TO OUT-COLOR
           WHEN FILE-RC1 = DFHRESP(NOTAUTH)
                MOVE 'NOT AUTH' TO OUT-MSG
                MOVE 2 TO OUT-COLOR
        END-EVALUATE.
      *
       RETURN-TO-CICS.
      *==============*
        EXEC CICS ENDBR FILE (FNAME)
        END-EXEC
        EXEC CICS RETURN
        END-EXEC
        GOBACK.
```

A poor man's CICS chargeback system – part 2

This month we conclude the code for a chargeback system that allocates costs based on two criteria – total transaction volume and transaction CPU time.

```
*-------------------------------------------------------------------
* APPLID TABLE DSECT...
*-------------------------------------------------------------------
APPLIDDS DSECT              APPLID TABLE ENTRY...
APPLNAME DS    CL8             APPLID NAME
APPLOFF  DS    H               OFFSET TO ACCUM FILEDS IN COST CENTRE
*-------------------------------------------------------------------
* COST CENTRE ACCUM TABLE DSECT...
*-------------------------------------------------------------------
CCTABDS  DSECT              COST CENTRE ACCUM TABLE ENTRY DSECT...
CCTRTAB# DS    F            TABLE HEADER: NUMBER OF TABLE ENTRIES
*                           TABLE ENTRY...
CCACCUM  DS    (APPLS*2)PL8    2 8-BYTE ACCUM FIELDS FOR EACH APPLID
CCTR     DS    ØCL5Ø           CODE AND DESCRIPTION (5Ø BYTES)...
CCCODE   DS    CL5,CL6           COST CENTRE CODE (5 DIGITS)
CCDESC   DS    CL39              COST CENTRE DESCRIPTION
CCTRTABE EQU   *-CCACCUM    LENGTH OF A TABLE ENTRY
*
KC9Ø2MP  CSECT              RESUME ORIGINAL CSECT
         EJECT
*             ******************************
*             *     PRINT OUTPUT LINES     *
*             ******************************
         SPACE 2
*      ********************************************
*      *   COST ALLOCATION REPORT TITLE LINE      *
*      ********************************************
CATITLE  DS    ØCL133
PCCCARH1 DC    CL1'1'
         DC    C'   RUN DATE:  '
SYSDATE1 DC    CL8' ',CL16' '
APPLTYPE DC    CL1Ø' '
         DC    C' CICS    COST ALLOCATION REPORT       '
FROM1    DC    CL8' '
         DC    C' THRU '
THRU1    DC    CL8' ',CL17' '
         DC    C'PAGE '
C1PAGENO DC    CL1'1'
         SPACE 1
*      ********************************************
*      *   COST ALLOCATION REPORT HEADER LINE 1   *
```

```
*      ********************************************
CAHEAD1  DS    ØCL133
         DC    C'- COST   '
CAHD1A   EQU   *+6
         DC    (CACOLS)C'____________________  '
*        DC    CL(L'CAHEAD1-(*-CAHEAD1))' '      FILLER IF NEEDED
         SPACE 1
*      ********************************************
*      *   COST ALLOCATION REPORT HEADER LINE 2    *
*      ********************************************
CAHEAD2  DS    ØCL133
         DC    CL8' CENTRE '
         DC    (CACOLS)C' TRANCOUNT   CPUTIME '
*        DC    CL(L'CAHEAD2-(*-CAHEAD2))' '      FILLER IF NEEDED
         SPACE 1
*      ********************************************
*      *   COST ALLOCATION REPORT DETAIL LINE      *
*      ********************************************
         DC    C' '
CADETAIL DS    ØCL133
CADETCTL DC    C'Ø',C' '                      ASA CARRIAGE CONTROL
CADETCC  DC    CL5' ',C' '                    COST CODE
CADET1   DC    CL(L'CADETAIL-(*-CADETAIL))' ' 1ST DETAIL COLUMN
CACOLS   EQU   6                              # OF COLUMNS/PAGE
CW       EQU   21                             COLUMN WIDTH
         SPACE 1
*      ************************************************
*      *   COST ALLOCATION REPORT TOTALS & PRINT LINES *
*      ************************************************
CATOTALS DC    (CACOLS*2)PL8'Ø'   TRAN & CPU TOTALS FOR EACH COLUMN
*
CADASHES DS    ØCL133
         DC    C' ',CL7' '
         DC    (CACOLS)C' _________ _________ '
*        DC    CL(L'CADASHES-(*-CADASHES))' '    FILLER IF NEEDED
CATOTAL1 DC    C' '               APPLID TRANCOUNT TOTAL LINE 1
CATOT1   DC    CL132' '
CATOTAL2 DC    CL133' '           APPLID CPUTIME TOTAL LINE 2
         SPACE 1
*      ****************************************************
*      *   COST CENTRE GRAND TOTALS REPORT TITLE LINE      *
*      ****************************************************
CTTITLE  DS    ØCL133
         DC    CL1'1'
         DC    C'   RUN DATE:  '
SYSDATE2 DC    CL8' '
         DC    CL25' '
         DC    C'COST CENTRE GRAND TOTALS',CL25' '
FROM2    DC    CL8' '
         DC    C' THRU '
THRU2    DC    CL8' ',CL8' '
```

```
          DC    C'PAGE 1'
          SPACE 1
*      ******************************************************
*      *   COST CENTRE GRAND TOTALS REPORT HEADER LINE 1   *
*      ******************************************************
CTHEAD1  DS    ØCL133
          DC    CL1'-'
          DC    CL9' '
          DC    C'COST'
          DC    CL7' '
          DC    C'APPLICATIONS WITHIN'
          DC    CL18' '
          DC    C'TOTAL NUMBER OF'
          DC    CL11' '
          DC    C'TOTAL CPU TIME'
          DC    CL36' '
          SPACE 1
*      ******************************************************
*      *   COST CENTRE GRAND TOTALS REPORT HEADER LINE 2   *
*      ******************************************************
CTHEAD2  DS    ØCL133
          DC    CL1' '
          DC    CL8' '
          DC    C'CENTRE'
          DC    CL6' '
          DC    C'COST CENTRE'
          DC    CL29' '
          DC    C'TRANSACTIONS'
          DC    CL13' '
          DC    C'(IN SECONDS)'
          DC    CL35' '
          SPACE 1
*      ******************************************************
*      *   COST CENTRE GRAND TOTALS REPORT DETAIL LINE     *
*      ******************************************************
CTDETAIL DS    ØCL133
CTDETCTL DC    CL1'Ø'
          DC    CL9' '
CTCSTCTR DC    CL5' '
          DC    CL6' '
CTAPLICS DC    CL4Ø' '
CTTRANS# DC    CL1Ø' '
          DC    CL12' '
CTCPUTM# DC    CL8' '
          DC    CL42' '
          SPACE 1
*      ******************************************************
*      *   COST CENTRE GRAND TOTALS TRANS TOTAL # LINE     *
*      ******************************************************
CTTRNPRT DS    ØCL133
```

```
PCCCTTRN DC    CL1'-'
         DC    CL1Ø' '
         DC    C'GRAND TOTAL: NUMBER OF TRANSACTIONS  ==>>>'
         DC    CL8' '
CTTRNTOT DC    CL1Ø' '
         DC    CL63' '
         SPACE 1
*      ****************************************************
*      *   COST CENTRE GRAND TOTALS CPUTIME TOTAL # LINE   *
*      ****************************************************
CTCPUPRT DS    ØCL133
PCCCTCPU DC    CL1'Ø'
         DC    CL1Ø' '
         DC    C'GRAND TOTAL: CPU TIME (IN SECONDS)  ==>>>'
         DC    CL28' '
CTCPUTOT DC    CL8' '
         DC    CL45' '
         SPACE 2
         TITLE '*** DATA CONTROL BLOCKS ***'
SYSPRINT DCB   DDNAME=SYSPRINT,MACRF=(PM),DSORG=PS,LRECL=133,RECFM=FA
PRODCOST DCB   DDNAME=PRODCOST,MACRF=(PM),DSORG=PS,LRECL=133,RECFM=FA
QUALCOST DCB   DDNAME=QUALCOST,MACRF=(PM),DSORG=PS,LRECL=133,RECFM=FA
TESTCOST DCB   DDNAME=TESTCOST,MACRF=(PM),DSORG=PS,LRECL=133,RECFM=FA
CCTOTPRT DCB   DDNAME=CCTOTPRT,MACRF=(PM),DSORG=PS,LRECL=133,RECFM=FA
         END
```

PROGRAM KC903MP

```
************************************************************************
*                     PROGRAM KC9Ø3MP                                  *
*  KC9Ø3MP IS AN EXTERNAL SUBROUTINE IN THE CICS MONTHLY CHARGEBACK    *
*  SYSTEM. IT IS CALLED BY KC9ØØMP AND KC9Ø1MP WHEN ERRORS OCCUR.      *
*  IT PERFORMS ALL ERROR ROUTINE PROCESSING (INCLUDING PRINTING THE    *
*  ERROR REPORT) AND RETURNS CONTROL BACK TO THE CALLER. IF 5Ø         *
*  ERRORS ARE ENCOUNTERED THEN KC9Ø3MP ABENDS WITH AN ABEND CODE       *
*  '1ØØØ'.                                                             *
************************************************************************
         TITLE 'KC9Ø3MP: ERROR ROUTINE PROCESSING'
KC9Ø3MP  CSECT                    DEFINE MODULE AS A CONTROL SECTION
         PRINT NOGEN
BEGIN    STM   R14,R12,12(R13)  SAVE MAIN PROGRAM'S REGISTERS
         USING KC9Ø3MP,R12      ESTABLISH REG 12 AS BASE REGISTER
         LR    R12,R15          LOAD PGM ENTRY POINT ADDR IN BASE REG
         LA    R11,SAVEA        LOAD ADDR OF PGM SAVEAREA IN REG 11
         ST    R13,SAVEA+4      SAVE ADDR OF SYSTEM SAVEAREA
         ST    R11,8(R13)       SAVE ADDR OF SAVEA IN SYS SAVEAREA
         LR    R13,R11          LOAD ADDR OF SAVEA IN REG 13
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
*********************************************************************
*   WHEN THIS ROUTINE GETS CONTROL REG. 1 MAY CONTAIN THESE       *
*   PARMLIST ADDRESSES:                                           *
*    REG 1 (+Ø)  ==>>> ERROR FLAG (1 BYTE HEX FIELD)              *
*    REG 1 (+4)  ==>>> DATE (SMF) (4 BYTE PACKED DEC FIELD)       *
*    REG 1 (+8)  ==>>> VTAM APPLID (8 BYTE CHAR FIELD)            *
*    REG 1 (+12) ==>>> TRANSID IN ERROR (4 BYTE CHAR FIELD)       *
*    REG 1 (+16) ==>>> COST CENTRE IN ERROR (8 BYTE CHAR FIELD)   *
*  NOTE: REG 1 WILL ALWAYS CONTAIN POINTERS TO THE ERROR FLAG,    *
*        DATE OF ERROR, AND APPLID; A POINTER TO TRANSID AND      *
*        COST CENTRE WILL ONLY EXIST IF THE APPROPRIATE ERROR     *
*        HAS OCCURRED.                                            *
*********************************************************************
ERRORRTN EQU   *                  ROUTINE TO PROCESS ERRORS
         L     R3,Ø(R1)           R3 PTS TO ADDR OF 1ST FULLWORD OF
*                                 DATA PASSED FROM KC9ØØMP
         L     R8,Ø(R3)           LOAD 1ST DATA (ERRFLAG) INTO REG 8
         L     R4,4(R1)           R4 POINTS TO 2ND DATA PASSED(ERRDATE)
         MVC   ERRDATE1(4),Ø(R4)  SAVE DATE OF ERROR
         L     R5,8(R1)           R5 POINTS TO 3RD DATA PASSED (APPLID)
         MVC   EAPPLID(8),Ø(R5)   MOVE APPLID TO ERROR PRINT LINE
         MVC   APPLID(8),Ø(R5)    SAVE APPLID
         UNPK  EDATE(5),ERRDATE1(4)  CONVERT ERR DATE TO ZONED FORMAT
         OI    EDATE+4,X'FØ'      ELIMINATE SIGN
         STCM  R8,B'1ØØØ',ERRFLAG SAVE ERROR FLAG BYTE
         TM    ERRFLAG,X'8Ø'      DID AN APPLID ERROR OCCUR?
         BC    1,APPLERR          YES: GO PROCESS IT
         TM    ERRFLAG,X'4Ø'      NO: DID A TRANSACTION ID ERROR OCCUR?
         BC    1,TRANERR          YES: GO PROCESS IT
         TM    ERRFLAG,X'2Ø'      NO: WAS THERE A COST CENTRE MISSING
*                                 FROM THE COST CENTRE ADDRESS TABLE?
         BC    1,ADDRERR          YES: GO PROCESS IT
MISCERR  MVC   EERRTYPE(46),ERRMSG4   MOVE ERROR MSG TO PRINT LINE
         MVC   EERRFLD(1),ERRFLAG MOVE ERROR FLAG TO ERR PRINTLINE
         B     ERRPRT             GO PRINT ERROR LINE
APPLERR  EQU   *
```

```
         MVC   EERRFLD(8),APPLID  MOVE APPLID TO PRINT LINE
         MVC   EERRTYPE(46),ERRMSG1   MOVE ERROR MSG TO PRINT LINE
         B     ERRPRT             GO PRINT ERROR LINE
TRANERR  EQU   *
         L     R6,12(R1)          R6 POINTS TO 4TH DATA PASSED(TRANSID)
         MVC   TRANSID(4),Ø(R6)   SAVE INVALID TRANSID
         MVC   EERRFLD(4),Ø(R6)   MOVE INVALID TRANID TO ERR PRINTLINE
         MVC   EERRTYPE(46),ERRMSG2   MOVE ERROR MSG TO PRINT LINE
         B     ERRPRT             GO PRINT ERROR LINE
ADDRERR  EQU   *
         L     R7,16(R1)          R7 PTS TO 5TH DATA PASSED (CURRCCTR)
         MVC   CURRCCTR(5),Ø(R7)  SAVE INVALID COST CENTRE
         MVC   EERRFLD(5),Ø(R7)   MOVE INVALID CCTR TO ERR PRINTLINE
         MVC   EERRTYPE(46),ERRMSG3   MOVE ERROR MSG TO PRINT LINE
ERRPRT   EQU   *                  ROUTINE TO PRINT ERROR LINE
         OPEN  (ERRPRINT,(OUTPUT)) OPEN PRINT OUTPUT FILE
         LA    R1,=A(CURRDATE)   LOAD PARM ADDR TO PASS TO KC9Ø4MP
         L     R15,=V(KC9Ø4MP)   LOAD ENTRY POINT OF KC9Ø4MP
         BALR  R14,R15           BRANCH TO KC9Ø4MP AND RETURN
         MVC   SYSDATE,CURRDATE  PLACE CURRENT DATE IN PRINT LINE
         MVI   ERRFLAG,X'ØØ'      CLEAR ERROR FLAG BYTE
         AP    ERRCOUNT,=PL1'1'   INCREMENT ERROR ACCUMULATOR
BAL   R9,ERROFLOW        PERFORM ERROR PRINT OVERFLOW RTN
         PUT   ERRPRINT,EPRTLINE  WRITE ERROR DETAIL LINE
         MVI   PCCERPT,C' '       SET FOR SINGLE SPACING
         CP    ERRCOUNT,=P'5Ø'    COMPARE ERROR COUNT TO 5Ø
         BNL   GETOUT             IF ERR CNT = OR > 5Ø THEN ABEND
         AP    ELINECNT,=PL1'1'   INCREMENT DETAIL LINE COUNT
         B     ENDOFJOB           THAT'S ALL FOLKS
ERROFLOW EQU   *                  RTN TO CHECK FOR END OF ERROR PAGE
         CP    ELINECNT,=PL2'55'  COMPARE LINE COUNT TO 25
         BNL   EHEADERS           IF = OR > 25 THEN PERFORM HEADER RTN
         BR    R9                 IF < 25 RETURN TO CALLING SECTION
EHEADERS EQU   *                  RTN TO WRITE ERROR PAGE HEADERS
         MVC   ELINECNT,=PL2'Ø'   CLEAR ERROR LINE COUNTER
         AP    EPAGECNT,=PL1'1'   INCREMENT PAGE NUMBER
         UNPK  EPAGENO,EPAGECNT   MOVE PAGE NUMBER TO HEADER LINE
         OI    EPAGENO+1,X'FØ'    ELIMINATE SIGN IN 2ND BYTE
         PUT   ERRPRINT,HDRELNE1  WRITE 1ST ERROR HEADER LINE
         PUT   ERRPRINT,HDRELNE2  WRITE 2ND ERROR HEADER LINE
         MVI   PCCERPT,C'-'       SKIP 2 LINES BEFORE DETAIL LINE PRT
         BR    R9                 RETURN TO CALLING SECTION
GETOUT   WTO   'CHARGEBACK SYSTEM TERMINATING DUE TO EXCESSIVE ERRORS',X
               ROUTCDE=14
         ABEND 1ØØØ,DUMP
ENDOFJOB L     R13,SAVEA+4
         LM    R14,R12,12(R13)
         SR    R15,R15
         BR    R14
         LTORG
```

```
         DC    C'ASSEMBLED BY MORY BINDLER &SYSDATE &SYSTIME'
ERRFLAG  DC    X'ØØ'              ERROR BYTE PASSED FROM KC9ØØMP
ERRDATE1 DC    PL4'ØØ'            DATE OF ERROR PASSED FROM KC9ØØMP
APPLID   DC    CL8' '             VTAM APPLID PASSED FROM KC9ØØMP
TRANSID  DC    CL4' '             TRANSACTION NAME PASSED FROM KC9ØØMP
CURRCCTR DC    CL8' '             COST CENTRE # PASSED FROM KC9ØØMP
CURRDATE DC    CL8' '             CURRENT DATE SUPPLIED BY KC9Ø4MP
BLANKLNE DC    CL133' '           BLANK LINE FOR REPORTS
ERRCOUNT DC    PL4'Ø'             NUMBER OF CUMULATIVE ERRORS
ELINECNT DC    PL2'65'            ERROR DETAIL LINE COUNT
EPAGECNT DC    PL2'Ø'             ERROR PAGE COUNT
ERRMSG1  DC    C'THIS APPLID DOES NOT EXIST IN THE APPLID TABLE'
ERRMSG2  DC    C'THIS TRANSACTION ID NOT DEFINED IN THE PROGRAM'
ERRMSG3  DC    C'THIS COST CENTRE DOES NOT EXIST IN ADDRESS TBL'
ERRMSG4  DC    C'MISC ERROR HAS OCCURRED: LOOK AT ERROR FLAG   '
SAVEREG1 DC    F'Ø'               SAVE REG 1 CONTENTS
SAVEA    DS    18F                REGISTER SAVE AREA
         EJECT
*      ***********************************
*      *   ERROR REPORT HEADER LINE 1    *
*      ***********************************
HDRELNE1 DS    ØCL133
PCCEHDR1 DC    CL1'1'
         DC    CL5' '
SYSDATE  DS    CL8
         DC    CL4Ø' '
         DC    C'CICS MONTHLY USAGE ERROR REPORT'
         DC    CL39' '
         DC    C'PAGE '
EPAGENO  DC    CL2' '
         DC    CL2' '
         SPACE 1
*      ***********************************
*      *   ERROR REPORT HEADER LINE 2    *
*      ***********************************
HDRELNE2 DS    ØCL133
PCCEHDR2 DC    CL1'-'
         DC    CL19' '
         DC    C'APPLID NAME'
         DC    CL9' '
         DC    C'DATE OF ERROR'
         DC    CL9' '
         DC    C'ERROR FIELD'
         DC    CL9' '
         DC    C'TYPE OF ERROR'
         DC    CL36' '
         SPACE 1
*      *********************************
*      *   ERROR REPORT DETAIL LINE   *
*      *********************************
```

```
EPRTLINE DS    ØCL133
PCCERPT  DC    CL1' '
         DC    CL2Ø' '
EAPPLID  DC    CL8' '
         DC    CL9' '
EDATE    DC    CL5' '
         DC    CL16' '
EERRFLD  DC    CL8' '
         DC    CL1Ø' '
EERRTYPE DC    CL5Ø' '
         DC    CL6' '
         SPACE 2
         TITLE '*** DATA CONTROL BLOCKS ***'
ERRPRINT DCB   DDNAME=ERRPRINT,MACRF=(PM),DSORG=PS,LRECL=133,RECFM=FA
         SPACE 1
         END   KC9Ø3MP
```

PROGRAM KC900MP

```
*************************************************************************
*  DEVELOPMENT LOG                                                      *
*************************************************************************
*               PROGRAM: KC9ØØMP                                        *
* THIS PROGRAM IS THE MAIN PROGRAM IN THE CICS MONTHLY CHARGEBACK       *
* SYSTEM. IT PERFORMS THESE FUNCTIONS:                                  *
*   (1) EXTRACTS CICS SMF MONITORING RECORDS (TYPE 11Ø, SUBTYPE ØØØ1)   *
*       AND REFORMATS THEM INTO FIXED-LENGTH 32Ø BYTE RECORDS.          *
*   (2) EXTRACTS THE VTAM APPLID AND SETS UP ADDRESSING FOR THE         *
*       MAIN TABLE THAT WILL STORE ALL THE APPROPRIATE CICS STATIS-     *
*       TICS FOR THE REPORTS.                                           *
*   (3) CALLS SUBROUTINE KC9Ø1MP TO BEGIN PROCESSING THE TRANSACTION    *
*       DATA RECORD.                                                    *
*   (4) PRINTS THE SMF DATA DICTIONARY AND MISC. TOTALS FROM THE SMF    *
*       BLOCKS PROCESSED.                                               *
*   (5) OPTIONALLY GENERATES AN OUTPUT FILE CONTAINING ALL THE TRAN-    *
*       SACTION DATA RECORDS. (THIS FACILITY NOT USED IN THE MONTHLY).*
*                                                                       *
* JCL REQUIRED TO RUN THIS PROGRAM:                                     *
*   //CICSMONR  EXEC PGM=KC9ØØMP                                       *
*   //STEPLIB   DD DSN=YOUR.LOADLIB,DISP=SHR                           *
*   //SYSUT1    DD DSN=YOUR.SMFRECS,UNIT=XXXX,DISP=OLD                 *
*   //SYSUT2    DD DSN=YOUR.OUTPUT.DATARECS,DISP=(NEW,CATLG),          *
*   //      UNIT=SYSDA,DCB=(DSORG=PS,RECFM=FB,LRECL=32Ø,BLKSIZE=64ØØ),*
*   //      SPACE=(CYL,(XXX,YY),RLSE)                                 *
*   //ERRPRINT  DD SYSOUT=*                                           *
*   //PRODCOST  DD SYSOUT=*
*   //QUALCOST  DD SYSOUT=*
*   //TESTCOST  DD SYSOUT=*
*   //CCTOTPRT  DD SYSOUT=*
```

```
*   //SYSPRINT  DD SYSOUT=*                                          *
*   //SNAPDUMP  DD SYSOUT=*                                          *
*   //SYSUDUMP  DD SYSOUT=*                                          *
*   //ABENDAID  DD SYSOUT=*                                          *
*   //ABNLHELP  DD DUMMY                                             *
*   //                                                               *
*                                                                    *
* CALLS MADE FROM KC900MP:
*   (1) KC901MP--CALLED TO PROCESS EACH TRANSACTION DATA RECORD--
*
*               FROM ROUTINE NAMED "CALLDREC"                        *
*   (2) KC901MP--CALLED AFTER ALL SMF RECORDS HAVE BEEN READ --
*
*               FROM ROUTINE NAMED "COSTREPT"                        *
*   (3) KC903MP--CALLED TO PROCESS ERROR CONDITION WHEN AN APPLID     *
*               NOT DEFINED IN THE APPLID TABLE IS ENCOUNTERED--     *
*               FROM ROUTINE NAMED "CALLERR"                         *
*                                                                    *
* MACRO LIBRARIES REQUIRED FOR ASSEMBLY:                             *
*     1. SYS1.MACLIB                                                 *
*     2. CICS410.SDFHMAC                                             *
**********************************************************************
         TITLE 'KC900MP: CICS CHARGEBACK REPORTS '
KC900MP  CSECT
         PRINT NOGEN
BEGIN    STM   R14,R12,12(13)    SAVE CONTENTS OF REG 0-12,14,15
         LR    R12,R15           LOAD PGM ENTRY POINT ADDR IN REG 12
         USING BEGIN,R12,R8      USE REGS 12 & 8 AS BASE REGISTERS
         LA    R1,1              INITIALIZE INDEX REG
         LA    R8,4095(R1,R12)   INITIALIZE 2ND BASE REG
         LA    R11,SAVEA         LOAD ADDR OF PGM SAVEAREA IN REG 11
         ST    R13,SAVEA+4       SAVE ADDRESS OF SYSTEM SAVEAREA
         ST    R11,8(R13)        SAVE ADDR OF SAVEA IN SYSTEM SAVEAREA
         LR    R13,R11           LOAD ADDR OF SAVEA IN REG 13
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
```

```
         OPEN  (SYSUT1,,SYSUT2,(OUTPUT)) OPEN INPUT AND OUTPUT FILES
         OPEN  (SYSPRINT,(OUTPUT))
READLBL  GET   SYSUT1,SMFREC      READ THE SMF RECORD
         USING MNSMFDS,R3         USE SMF HEADER/PRODUCT SECTION DEF
         LA    R3,SMFREC          MAP OVER SMF RECORD AREA
         LH    R1Ø,SMFMNLEN       LOAD LENGTH OF BLOCK INTO REG 1Ø
         LA    R9,SMFMNRVN-SMFMNLEN LOAD LENGTH OF HEADER REC INTO R9
         AP    BLKCNT,=P'1'       ADD 1 TO BLOCK COUNT
         CLC   SMFMNRTY,=X'6E'    IS THIS A CICS TYPE 11Ø RECORD?
         BNE   READLBL            NO: THEN READ NEXT BLOCK         MB1
         CLC   SMFMNSTY,=X'ØØØ1'  IS THIS A MONITORING SUBTYPE?
         BNE   READLBL            NO: THEN READ NEXT BLOCK         MB1
         CLC   SMFMNCL(2),=X'ØØØ3'  IS THIS A PERFORMANCE RECORD?
         BNE   READLBL            NO: THEN READ NEXT BLOCK
GETAPLID EQU   *                  RTN TO EXTRACT APPLID FROM SMF REC
         OC    SMFMNPRN,=8C' '    FORCE BLANK-FILL AND UPPER CASE
         MVC   APPLID(8),SMFMNPRN  SAVE VTAM APPLID
         MVC   SMFDATE(4),SMFMNDTE SAVE DATE WHEN SMF BLOCK WAS WRITTEN
         LA    R11,APPLS          REG 11: # OF ELEMENTS IN APPLID TBL
         LA    R4,APPLTABL        REG 4 POINTS TO APPLID TABLE
APPLLOOP EQU   *
         CLC   Ø(8,R4),SMFMNPRN   DOES APPLID MATCH?
         BE    DOAPPLID           YES: GO PROCESS IT
         LA    R4,APPLTABE(,R4)   BUMP TO NEXT APPLID IN TABLE
         BCT   R11,APPLLOOP       TEST FOR END OF TABLE
*
         LA    R11,TESTTAB#       REG 11: # OF ELEMENTS IN TESTID TBL
         LA    R4,TESTTABL        REG 4 POINTS TO TESTID TABLE
TESTLOOP EQU   *
         CLC   Ø(8,R4),SMFMNPRN   DOES TEST APPLID MATCH?
         BE    WRITERC            YES: WRITE THE RECORD          MB1
         LA    R4,TESTTABE(R4)    BUMP TO NEXT TESTID IN TABLE
         BCT   R11,TESTLOOP       TEST FOR END OF TABLE
*
APPLERR  EQU   *                  ERROR: AN APPLID MATCH DID NOT OCCUR
         OI    ERRFLAG,X'8Ø'      INDICATE APPLID ERROR
         MVC   ERRDATE(4),SMFMNDTE SAVE DATE WHEN SMF BLOCK WRITTEN
CALLERR  EQU   *                  SET UP CODE TO CALL ERROR SUBRTN
         LA    R1,=A(ERRFLAG,ERRDATE,APPLID)  LOAD PARMLIST ADDRESSES
         L     R15,=V(KC9Ø3MP)    LOAD ADDRESS OF ERROR SUBROUTINE
         BALR  R14,R15            INVOKE ERROR SUBROUTINE & RETURN
*DOWRITE  B     WRITERC            PERFORM WRITERC RTN
DOAPPLID EQU   *                  WE HAVE FOUND VALID APPLID
         CP    SMFMNDTE,FROMDATE
         BNL   NOTFROM
         MVC   FROMDATE,SMFMNDTE SAVE LOWEST PROCESS DATE AS FROM DATE
NOTFROM  EQU   *
         CP    SMFMNDTE,TODATE
         BNH   NOTTO
         MVC   TODATE,SMFMNDTE    SAVE HIGHEST PROCESS DATE AS TO DATE
```

```
NOTTO    EQU   *
         MVC   APPLID1,Ø(R4)     SAVE APPLID NAME
         LH    R11,8(R4)         REG 11: HAS OFFSET OF APPLID IN COST
*                                CENTRE TABLE
         STH   R11,APPLOFF       SAVE OFFSET OF APPLID IN COST CTR TBL
         B     WRITERC           PERFORM WRITERC RTN
NEXTBLK  DS    ØH
         BC    B'1111',READLBL    UNCONDITIONAL BRANCH TO READLBL
WRITERC  EQU   *
         ST    R4,SAVEREG4        SAVE REG 4 CONTENTS
         CLC   SMFMNCL,=H'3'      IS THIS A PERFORMANCE TYPE RECORD?
         BE    SKIPD              YES: THEN SKIP DICTIONARY PROCESS
         CLC   SMFMNCL,=H'1'      IS THIS A DICTIONARY RECORD?
         BNE   SKIPD              NO: THEN SKIP DICTIONARY PROCESS
         BAL   R8,PRDICT          PERFORM PRINT DICTIONARY RTN
         L     R4,SAVEREG4        RESTORE REG 4 CONTENTS
         B     READLBL            READ NEXT RECORD
SKIPD    EQU   *
         LH    R6,SMFMNDRN        SAVE COUNT OF DATA RECORDS   MB1
         CH    R6,=H'Ø'           DO ANY DATA RECORDS FOLLOW?
         BE    NEXTBLK            NO: READ ANOTHER BLOCK
         LH    R7,SMFMNDRA+2      GET OFFSET TO 1ST CICS DATA REC  MB1
         AR    R7,R3              R7 POINTS TO 1ST CICS DATA REC   MB1
         LA    R2,SMFMNLEN+Ø(R7)  LOAD ADDRESS OF FIRST DATA RECORD
         MVC   TEMPLENG,SMFMNDRL
         LH    R5,TEMPLENG        SAVE LENGTH OF DATA RECORD
WRITEAG  ST    R6,REGSIX
         ST    R7,REGSEV
         ST    R3,REGTWE
         AP    DATCNT,=P'1'       ADD 1 TO DATA RECORDS OUTPUTTED COUNT
WRITEIT  PUT   SYSUT2,Ø(R7)       WRITE THE OUTPUT RECORD
         MVC   TRANSID,Ø(R7)     SAVE TRANSACTION NAME
         CLC   TRANSID,=CL4'CARM' IS THIS A "CARM" TRANSACTION?    MB1
         BE    CALLDREC            YES: PROCESS IT                 MB1
         CLC   TRANSID,=CL4'CIGN' IS THIS A "CIGN" TRANSACTION?    MB1
         BE    CALLDREC            YES: PROCESS IT                 MB1
         CLI   TRANSID,C'C'      IS THIS AN IBM TRANSACTION?
         BE    IGNOREIT            YES: BYPASS CALL TO KC9Ø1MP
         OC    TRANSID,TRANSID   IS THIS TRANS. NAME NULLS?
         BZ    IGNOREIT            YES: BYPASS CALL TO KC9Ø1MP
         TM    TRANSID,X'FØ'     IS THIS TRANS. NAME NUMERIC?
         BO    IGNOREIT            YES: BYPASS CALL TO KC9Ø1MP
CALLDREC EQU   *                 SET UP CODE TO CALL DATAREC SUBRTN
         LA    R1,=A(SMFDATE,APPLID,APPLOFF,TRANSID) LOAD PARMLIST-MB2
         L     R15,=V(KC9Ø1MP)   LOAD ADDRESS OF DATAREC SUBROUTINE
         BALR  R14,R15           INVOKE KC9Ø1MP SUBROUTINE & RETURN
IGNOREIT EQU   *
         LA    R7,Ø(R5,R7)        BUMP TO NEXT DATA BUCKET       MB1
         BCT   R6,WRITEAG         WRITE NEXT DATA BUCKET
         B     READLBL            READ NEXT SET OF RECORDS       MB1
```

```
*****************************************************
*  SUBROUTINE TO CONVERT A JULIAN DATE TO GREGORIAN  *
*****************************************************
CALCDATE EQU   *         CONVERT JULIAN DATE (@ REG1) TO GREGORIAN
         ZAP   FEB,=P'28'         SET FEBRUARY FOR NON-LEAPYEAR
         ZAP   DAYS,2(2,R1)       GET JULIAN DATE DAYS TO WORK AREA
         ZAP   CENT,=P'Ø'         GET JULIAN DATE CENTURY
         MVC   CENT(2),Ø(R1)
         UNPK  GREGYY,CENT+1(2)
         MVC   YY,GREGYY
         TR    CENT(1),=X'192Ø'   IF CENT=ØØ TRANSLATE TO "19"
*                                  IF CENT=Ø1 TRANSLATE TO "2Ø"
         CLC   CENT(2),=X'2ØØØ'   IS YEAR 2ØØØ?
         BE    LEAPYEAR           YES: GO PROCESS LEAPYEAR
         ZAP   CENT1,CENT         PREPARE FOR DIVIDE  ("MB3')
         DP    CENT1,=PL3'4ØØØ'   DIVIDE BY 4ØØØ ("MB3")
         CP    CENT1+4(3),=PL3'Ø' IF NO REMAINDER IT'S A LEAP YEAR-MB3
         BNE   DODAYS
LEAPYEAR EQU   *
         ZAP   FEB,=P'29'         LEAP YEAR:SET FEBRUARY FOR LEAP YEAR
DODAYS   EQU   *
         LA    R5,12              SET MONTH COUNTER
         LA    R2,DAYPERMO        -> DAYS PER MONTH TABLE
         ZAP   MONTH,=P'1'
NEXTMON  EQU   *
         CP    DAYS,Ø(2,R2)
         BNH   GOTGREG            DONE
         AP    MONTH,=P'1'        INCREMENT MONTH
         SP    DAYS,Ø(2,R2)       SUBTRACT THIS MONTH FROM DAY TOTAL
         LA    R2,2(,R2)
         BCT   R5,NEXTMON
GOTGREG  EQU   *
         UNPK  GREGMM,MONTH
         OI    GREGMM+2,X'FØ'
         MVC   MM,GREGMM+1
         UNPK  GREGDD,DAYS
         OI    GREGDD+2,X'FØ'
         MVC   DD,GREGDD+1
         BR    R14                RETURN
*********************************************************************
*  SUBROUTINE TO PRINT DATA DICTIONARY... ONCE PER RUN...
PRDICT   CLI   ONCEFLG,C'1'
         BER   R8                 DO THIS ROUTINE ONLY ONCE
         MVI   ONCEFLG,C'1'
         MVC   TEMPLENG,SMFMNDRN
         LH    R6,TEMPLENG        SAVE COUNT OF DATA RECORDS
         CH    R6,=H'Ø'           SEE IF NO DATA RECORDS FOLLOW
         BE    NEXTBLK            READ ANOTHER BLOCK
         MVC   TEMPLENG,SMFMNDRA+2
         LH    R7,TEMPLENG        SAVE POINTER TO FIRST DATA RECORD
```

```
         LA    R2,SMFMNLEN+Ø(R7)  LOAD ADDRESS OF FIRST DATA RECORD
         MVC   TEMPLENG,SMFMNDRL
         LH    R5,TEMPLENG        SAVE LENGTH OF DATA RECORD
PUTAG    DS    ØH                 FORMAT OUTPUT LINE
         CP    LINES,FULLPAGE
         BL    SKIPS
         LA    R4,1                CHANGE IN DICTIONARY DESCRIPTION
         PUT   SYSPRINT,TITLE
         PUT   SYSPRINT,HEADER
         MVI   OUTLINE,C'Ø'            DOUBLE SPACE AFTER HEADING
         ZAP   LINES,=P'Ø'
         TR    PAGE,NEXTPAGE           INCREMENT PAGE NUMBER
SKIPS    MVC   OUTLINE+2(8),Ø(R2)
         MVC   OUTLINE+16(8),18(R2)
         MVC   OUTLINE+29(3),9(R2)
         MVC   OUTLINE+37(1),8(R2)
         MVC   TEMPLENG,12(R2)
         LH    R7,TEMPLENG
         CVD   R7,DOUBVAR
         MVC   OUTLINE+42(6),=X'4Ø2Ø2Ø2Ø212Ø'
         ED    OUTLINE+42(6),DOUBVAR+5
         CVD   R4,DOUBVAR
         MVC   OUTLINE+52(6),=X'4Ø2Ø2Ø2Ø212Ø'
         ED    OUTLINE+52(6),DOUBVAR+5
         AR    R4,R7
         PUT   SYSPRINT,OUTLINE
         AP    LINES,=P'1'             COUNT LINES
         MVI   OUTLINE,C' '            SINGLE SPACE
         LA    R2,Ø(R5,R2)
         BCT   R6,PUTAG          WRITE NEXT DATA RECORD
         BR    R8
*
MVCFLEID MVC   PARMS(Ø),2(R2)
NOPARMS  WTO   '*** ERROR *** NO PARM FILE-ID SPECIFIED',ROUTCDE=14
         ABEND 1                 >>>>>>>>> C R A S H <<<<<<<<<
SYSTPERR WTO   '*** ERROR *** UNEXPECTED SYSTEM TYPE-ID',ROUTCDE=14
         ABEND 2                 >>>>>>>>> C R A S H <<<<<<<<<
JRNLEND  CLOSE (SYSUT1,,SYSUT2)  CLOSE INPUT AND OUTPUT FILES
         MVC   OPARMS,PARMS
         WTO   'PARMS SPECIFIED -              XXXXXXXX',ROUTCDE=14
OPARMS   EQU   *-24,18
         MVC   OBLKCNT,=X'4Ø2Ø2Ø206B2Ø2Ø206B2Ø212Ø' MOVE EDIT PATTERN
         ED    OBLKCNT,BLKCNT    EDIT LABEL RECORD COUNT
         WTO   'PHYSICAL BLOCKS READ         999,999,999',ROUTCDE=14
OBLKCNT  EQU   *-18,12
         WTO   '                                       ',ROUTCDE=14
         MVC   ORECOVL,=X'4Ø2Ø2Ø206B2Ø2Ø206B2Ø212Ø' MOVE EDIT PATTERN
         ED    ORECOVL,RECOVL    EDIT DATA RECORD COUNT
         WTO   'NON MONITOR DATA RECORDS     999,999,999',ROUTCDE=14
ORECOVL  EQU   *-18,12
```

```
         MVC   ODATCNT,=X'4020202006B2020206B202120' MOVE EDIT PATTERN
         ED    ODATCNT,DATCNT    EDIT DATA RECORD COUNT
         WTO   'DATA RECORDS OUTPUTTED       999,999,999',ROUTCDE=14
ODATCNT  EQU   *-18,12
         WTO   '                             ----------',ROUTCDE=14
         AP    RECOVL,DATCNT     ADDITION COUNT + CHANGE COUNT
         MVC   OTOTWTE,=X'4020202006B2020206B202120' MOVE EDIT PATTERN
         ED    OTOTWTE,RECOVL    EDIT DATA RECORD COUNT
         WTO   'TOTAL RECORDS INPUT          999,999,999',ROUTCDE=14
OTOTWTE  EQU   *-18,12
COSTREPT EQU   *                 FORMAT AND PRINT CHARGEBACK REPORTS
         LA    R1,FROMDATE
         BAL   R14,CALCDATE
         MVC   FROMGREG,GREGDATE GET FROM DATE TO PASS
         LA    R1,TODATE
         BAL   R14,CALCDATE
         MVC   TOGREG,GREGDATE   GET TO DATE TO PASS
         OI    SWITCH,X'80'      INDICATE NO MORE DATA RECS TO PROCESS
         LA    R1,=A(SWITCH,FROMGREG)  LOAD ADDRESS OF PARM
         L     R15,=V(KC901MP)   LOAD ENTRY POINT OF EXT. SUBRTN
         BALR  R14,R15           BRANCH TO EXT. SUBRTN "KC901MP"
ENDOFJOB L     R13,SAVEA+4
         LM    R14,R12,12(R13)
         SR    R15,R15
         BR    R14
         LTORG
         DC    C'ASSEMBLED BY MORY BINDLER &SYSDATE &SYSTIME'
*
APPLID   DC    CL8' '            VTAM APPLID IN SMF REC (SMFMNPRN)
APPLID1  DC    CL8' '            VTAM APPLID FOUND IN APPLID TABLE
CURRCCTR DC    CL8' '            CURRENT COST CTR # IN COST CTR TBL
SMFDATE  DC    PL4'0'            DATE THE SMF BLOCK WAS WRITTEN
APPLOFF  DS    H                 OFFSET OF APPLID IN COST CENTRE TABLE
TRANSID  DS    CL4               TRANSACTION NAME
SAVEREG3 DC    F'0'              SAVE CONTENTS OF REG. 3
SAVEREG4 DC    F'0'              SAVE CONTENTS OF REG. 4
SAVEREG5 DC    F'0'              SAVE CONTENTS OF REG. 5
SAVEREG6 DC    F'0'              SAVE CONTENTS OF REG. 6
SAVEREG7 DC    F'0'              SAVE CONTENTS OF REG. 7
SAVEREG8 DC    F'0'              SAVE CONTENTS OF REG. 8
SAVEREG9 DC    F'0'              SAVE CONTENTS OF REG. 9
SAVEREGA DC    F'0'              SAVE CONTENTS OF REG. 10
SAVEREGB DC    F'0'              SAVE CONTENTS OF REG. 11
SAVEREGC DC    F'0'              SAVE CONTENTS OF REG. 12
SWITCH   DC    X'00'             FLAG BYTE USED WITH MISC FUNCTIONS
PRTFLAG  DC    X'00'             FLAG BYTE USED TO CONTROL REPT PRT
ERRFLAG  DC    X'00'             FLAG BYTE USED FOR ERROR CONDITIONS
BLANKLNE DC    CL133' '          BLANK LINE FOR REPORTS
ERRDATE  DC    PL4'0'            DATE OF ERROR
ERRCOUNT DC    PL4'0'            NUMBER OF CUMULATIVE ERRORS
```

```
ELINECNT DC    PL2'65'           ERROR DETAIL LINE COUNT
EPAGECNT DC    PL2'Ø'            ERROR PAGE COUNT
D        DS    D                 DOUBLE WORD CONVERSION WORK AREA
PARMS    DS    ØCL19             PARMS
DATE1    DC    CL5' '            STARTING DATE
DATE2    DC    CL5' '            ENDING DATE
JOBID    DC    CL8' '            JOB-ID SEARCH ARGUMENT
KEEPDT   DC    CL1' '            IF THIS IS A Y THEN MOVE DATE TO PROG.
SDATE    DC    CL5' '            STARTING DATE
TIMESTAM DC    CL8' '            TEMPORARY SAVE AREA FOR THE SMFDAT/TIM
SAVEA    DS    18F               SAVE AREA
REGTWE   DS    F                 REGISTER SAVE AREA
REGSIX   DS    F
REGSEV   DS    F
TEMPLENG DS    H                 TEMPORARY LENGTH FIELD
*
FROMDATE DC    PL4'9999999'      PROCESSING PERIOD... FROM DATE
TODATE   DC    PL4'Ø'                                 TO DATE
FROMGREG DC    CL8' '            PROCESSING PERIOD...
TOGREG   DC    CL8' '              TO BE PASSED TO KC9Ø2MP FOR REPORTS
*
GREGDATE DS    ØCL8              GREGORIAN PROCESS DATE...
MM       DC    CL2' ',C'/'
DD       DC    CL2' ',C'/'
YY       DC    CL2' '
*
CENT     DC    PL4'Ø'            WORK FIELD FOR LEAP YEAR CALC
         ORG   CENT+2
LEAP     DC    PL2'Ø'            REMAINDER FOR LEAP YEAR CALC
CENT1    DC    PL7'Ø'
CENT2    DC    PL4'Ø'
SAVECENT DS    PL4
*
MONTH    DC    PL2'Ø'
DAYS     DC    PL2'Ø'
GREGMM   DC    CL3' '            GREGORIAN CONVERSION WORK MONTH
GREGDD   DC    CL3' '                                      DAY
GREGYY   DC    CL3' '                                      YEAR
*
DAYPERMO EQU   *                 DAYS PER MONTH TABLE...
JAN      DC    PL2'31'           .
FEB      DC    PL2'28'           .
MAR      DC    PL2'31'           .
APR      DC    PL2'3Ø'           .
MAY      DC    PL2'31'           .
JUN      DC    PL2'3Ø'           .
JUL      DC    PL2'31'           .
AUG      DC    PL2'31'           .
SEP      DC    PL2'3Ø'           .
OCT      DC    PL2'31'           .
```

```
NOV      DC    PL2'3Ø'           .
DEC      DC    PL2'31'           .
*
BLKCNT   DC    PL5'Ø'            PHYSICAL BLOCKS READ COUNT
RECOVL   DC    PL5'Ø'            RECORDS OVERLOOKED
DATCNT   DC    PL5'Ø'            DATA RECORDS OUTPUTTED
ONCEFLG  DC    C'Ø'              SET TO OFF
DOUBVAR  DS    D                 DOUBLE WORD CONVERSION WORK AREA
LINES    DC    PL2'6Ø'           PRINT LINE COUNTER
FULLPAGE DC    P'6Ø'             LINES PER PAGE
NEXTPAGE EQU   *-C'1'            TRANSLATE TABLE
         DC    C'23456789'          FOR NEXT PAGE (UP TO 9)
OUTLINE  DC    CL133' '
TITLE    DS    ØCL133            TITLE LINE
         DC    C'1 ------------------  DATA DICTIONARY  -----------  PA+
               GE '
PAGE     DC    C'1'              CURRENT PAGE NUMBER
         DC    CL(L'TITLE-(*-TITLE))' '
HEADER   DC    CL133'- GROUP         FIELD       CODE  FORMAT  LENGTH  +
               POSITION'
BLANKL   DC    CL133' '
EPARMLST DC    A(ERRFLAG)        ERROR FLAG
         DC    A(ERRDATE)        DATE OF ERROR
         DC    A(APPLID)         APPLID IN ERROR
PARMLIST DC    A(SMFDATE)        DATE WHEN SMF BLOCK WAS WRITTEN
         DC    A(APPLID)         VTAM APPLID
*
         SPACE 2
*
SNAPDUMP DCB   DDNAME=SNAPDUMP,MACRF=(W),DSORG=PS,RECFM=VBA,LRECL=133, X
               BLKSIZE=133Ø
*                  **************
*                  *   TABLES   *
*                  **************
         SPACE 2
**********************
*    APPLID TABLE    *
**********************
*  THIS TABLE IS ARRANGED IN REPORT SEQUENCE...
*        NOTE: THE HALFWORD FOLLOWING EACH APPLID IS A DISPLACEMENT
*              (RELATIVE TO Ø) INTO A COST TABLE (16 BYTE ENTRIES)
         SPACE 1
         DC    C'VTAM APPLID TABLE FOLLOWS (APPLTABL):'
APPLTABL DS    ØF
         DC    CL8'CICSPAPB',Y(((*-APPLTABL)/1Ø)*16)
APPLTABE EQU   *-APPLTABL             APPLTABL ENTRY LENGTH
         DC    CL8'CICSPAPD',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSPAPF',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSPAPG',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSPAPH',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSVS  ',Y(((*-APPLTABL)/1Ø)*16)
```

```
         DC    CL8'CICSQAPB',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSQAPD',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSQAPF',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSQAPG',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSQAPH',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSQA  ',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSTAPB',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSTAPD',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSTAPF',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSTAPG',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSTAPH',Y(((*-APPLTABL)/1Ø)*16)
         DC    CL8'CICSTST ',Y(((*-APPLTABL)/1Ø)*16)
APPLS    EQU   (*-APPLTABL)/APPLTABE   NUMBER OF APPLIDS
APPLEND  EQU   *             MARK END OF APPLID TABLE
*
         SPACE 2
**************************
*    TEST APPLID TABLE    *
**************************
         SPACE 1
         DC    C'TEST APPLID TABLE FOLLOWS (TESTTABL):'
TESTTABL DS    ØF
         DC    CL8'CICSPAPE'
TESTTABE EQU   *-TESTTABL             LENGTH OF A TEST TABLE ENTRY
         DC    CL8'CICSQAPE'
         DC    CL8'CICSTAPE'
TESTTAB# EQU   (*-TESTTABL)/TESTTABE  NUMBER OF TEST ENTRIES
*
         EJECT
*
         TITLE '*** DATA CONTROL BLOCKS ***'
SYSUT1   DCB   DDNAME=SYSUT1,MACRF=(GM),EODAD=JRNLEND,DSORG=PS
SYSUT2   DCB   DDNAME=SYSUT2,MACRF=(PM),DSORG=PS,LRECL=32Ø,            X
               BUFNO=1Ø
SYSPRINT DCB   DDNAME=SYSPRINT,MACRF=(PM),DSORG=PS,LRECL=133,RECFM=FA
         SPACE 1
         TITLE '*** OUTPUT RECORD DEFINITION ***'
JXRECRD  DS    CL32Ø
         TITLE '*** INPUT  RECORD AREA       ***'
SMFREC   DS    CL32768
         TITLE '*** LAYOUT AND DESCRIPTION OF DICTIONARY  ***'
         PRINT GEN
*        DFHCMPDR DIR,,PER
         TITLE '*** SMF LABEL AND DATA RECORD DEFINITIONS ***'
         PRINT GEN
MNSMFDS  DFHMNSMF PREFIX=SMF
         END   KC9ØØMP
```

Mory Bindler
Consultant
3D Business Solutions (USA)

Set a terminal to upper or lower-case

A CICS terminal is normally set to translate all input to upper-case. However, there are products that put the terminal in lower-case mode in order to allow the user to enter some text, or for some other reason. If an abend happens at that time, the terminal may remain in CICS in lower-case mode, thus behaving differently from a standard one. For example, a transaction name entered on it will not be recognized, because upper-case translation does not take place, and an 'Invalid transaction identification xxxx' message will appear.

To set it back to upper-case translation again, it is necessary to issue the command EXEC CICS SET TERMINAL('termid') UCTRAN. This could be done in CECI. However, normal users don't have access to that transaction. For that reason, I wrote a program that puts a terminal in upper or lower-case and that can be called either by an associated transaction or from within another program via EXEC CICS LINK. The program is called TCASEUP, and I have it associated with transaction TCAS. If you want to use another transaction name, you must first set its working-storage variable to that name. This allows the program to know if it's being called directly by its own transaction or called by another program.

To set a terminal to upper-case, just type (in lower-case, of course) the transaction name without any arguments, or call the program from within another without a COMMAREA being passed.

To set it to lower-case, add an L after the transaction name, leaving one (and only one) space in between: 'TCAS L'. This will pass a COMMAREA with an L in the first (and only) byte.

TCASEUP SOURCE CODE

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. TCASEUP.
      *=============================================================*
      *                                                             *
      *   Program to set a terminal to upper-case or lower-case.    *
      *   This program is assigned to transaction TCAS, and can be  *
```

```
*   called either directly by the transaction or from within    *
*   another program via EXEC CICS LINK.                         *
*   To put the terminal in upper case, invoke this program      *
*   without any parameter / COMMAREA. To put it in lower case,  *
*   pass an "L" as transaction argument or in the first byte    *
*   of a COMMAREA.                                              *
*                                                               *
*===============================================================*
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*===============================================================*
* To distinguish whether the program was called by a transaction*
* or by another program, the variable below should be assigned  *
* the name of this program's transaction.                       *
* If this program is called by transaction, the optional        *
* argument will appear in parametro1.                           *
*===============================================================*
 Ø1  THIS-PROGRAM-TRANSACTION PIC X(4) VALUE 'TCAS'.
 Ø1  RECEIVE-AREA.
     Ø2     LRECEBE     PIC S9(4) COMP VALUE +6.
     Ø2     RECEBE.
        Ø4  TRANSNAME   PIC X(4) VALUE SPACES.
        Ø4  FILLER      PIC X    VALUE SPACES.
        Ø4  PARAMETRO1  PIC X    VALUE SPACES.
 Ø1  MSGLOW PIC X(25)   VALUE 'Terminal set to lowercase'.
 Ø1  MSGUPP PIC X(25)   VALUE 'TERMINAL SET TO UPPERCASE'.
*
 LINKAGE SECTION.
*===============================================================*
* If called by another program passing a COMMAREA (one byte),   *
* argument should appear in parametro2.                         *
*===============================================================*
 Ø1  DFHCOMMAREA.
     Ø2     PARAMETRO2  PIC X.
     Ø2     FILLER      PIC X(3Ø).
*
*===============================================================*
 PROCEDURE DIVISION.
*===============================================================*
*
     EXEC CICS HANDLE ABEND LABEL(SAIDA)
     END-EXEC
*
     IF EIBTRNID = THIS-PROGRAM-TRANSACTION
        EXEC CICS IGNORE CONDITION LENGERR
                                   ENDDATA
        END-EXEC
        EXEC CICS RECEIVE INTO(RECEBE)
                          LENGTH(LRECEBE)
```

```
            END-EXEC
         END-IF.
      *
         IF PARAMETRO1 = 'L' OR PARAMETRO2 = 'L'
         OR PARAMETRO1 = 'l' OR PARAMETRO2 = 'l'
            EXEC CICS SET TERMINAL(EIBTRMID) NOUCTRAN
            END-EXEC
         ELSE
            EXEC CICS SET TERMINAL(EIBTRMID) UCTRAN
            END-EXEC
         END-IF.
      *
         IF EIBTRNID = THIS-PROGRAM-TRANSACTION
            IF PARAMETRO1 = 'L' OR PARAMETRO1 = 'l'
               EXEC CICS SEND FROM(MSGLOW)
                              LENGTH(25)
                              ERASE
               END-EXEC
            ELSE
               EXEC CICS SEND FROM(MSGUPP)
                              LENGTH(25)
                              ERASE
               END-EXEC
            END-IF
         END-IF.
      *
          SAIDA.
      *======*
         EXEC CICS RETURN END-EXEC.
```

Systems Programmer
(Portugal)

Test the 3270 bridge via batch using MQSeries

If you are working with MQSeries you can test the 3270 bridge with the programs CIT000 and CRMQTE01. CIT000 is the CICS program and you must define this program in CICS (PPT-entry or autoinstall). You must also define the transaction T000 belonging to this program (PCT-entry). Compile the CIT000 as a command-level program. CRMQTE01 is a batch program using MQSeries. To run the program use the JCL listed below. Compile this program as a batch program using MQSeries.

For MQSeries the process CICS.SYST.BRIDGE and the queues CICS.SYST.INIT.QUEUE and CICS.SYST.ISBSTCX.FROM.BATCH .TEST.Q001 must be defined.

As soon as the following message appears in the CICSLOGs, “+The MQSeries test program was called successfully!”, you know your test was successful.

CIT000

```
         PUNCH ' MODE AMODE(31),RMODE(ANY) '
CIT000   TITLE '*** ASSEMBLER-TESTPROGRAM FOR MQSERIES ***'
*********************************************************************
* NAME:        CIT000                                               *
* FUNCTION:    THIS PROGRAM WRITES A MESSAGE INTO THE CICSLOG AFTER  *
*              IT WAS CALLED VIA THE TRANSACTION T000.              *
*              THE TRANSACTION T000 WAS STARTED VIA THE 3270 BRIDGE. *
*********************************************************************
         EJECT
*********************************************************************
*        DEFINITIONS                                                *
*********************************************************************
         SPACE
DFHEISTG DSECT
RESPONSE DS    F
         EJECT
*********************************************************************
*        MAINLINE                                                   *
*********************************************************************
CIT000   CSECT
         DFHEIENT
         SPACE
         EXEC CICS WRITE OPERATOR
*
             TEXT('The MQSeries test program was called successfully!')*
             RESP(RESPONSE)
         CLC   RESPONSE,DFHRESP(NORMAL)
         BE    RETURN
         EXEC  CICS ABEND ABCODE('WRIT')
         SPACE
RETURN   EQU   *
         EXEC  CICS RETURN
         EJECT
         EQUREG
         EJECT
         LTORG
         END
```

CRMQTE01

```
CRMQTEØ1 TITLE 'BATCH INTERFACE: TEST THE 327Ø BRIDGE VIA MQSERIES    '
*************************************************************************
* NAME:        CRMQTEØ1                                                 *
* FUNCTION:    THIS PROGRAM PUTS MESSAGES TO MQSERIES FOR LATER         *
*              PROCESSING IN CICS.                                      *
*              THE PROGRAM IS CALLED VIA THE BATCH.                     *
*              A TRANSACTION (TØØØ) IS STARTED VIA THE 327Ø BRIDGE.     *
*************************************************************************
         DSECT
         CVT     DSECT=YES,LIST=NO     COMMUNICATIONS VECTOR TABLE
         SPACE
CRMQTEØ1 CSECT
CRMQTEØ1 AMODE ANY
CRMQTEØ1 RMODE 24
         SPACE
*************************************************************************
*        INITIALIZATION                                                 *
*************************************************************************
         SPACE
         STM   R14,R12,12(R13)         SAVE CALLER'S REGISTERS
         USING CRMQTEØ1,CODEREG,CODEREG2 ESTABLISH ADDRESSABILITY
         LR    CODEREG,R15             LOAD BASE REGISTER
         LR    CODEREG2,R15            LOAD BASE REGISTER
         AH    CODEREG2,H4Ø96          LOAD BASE REGISTER
         ST    R13,SAVEAREA+4
*
         LR    R2,R13
         LA    R13,SAVEAREA
         ST    R13,8(,R2)
*
         MVC   PARM,Ø(R1)
         B     MAIN
         DC    CL22'**CRMQTEØ1**CRMQTEØ1**'
         SPACE
H4Ø96    DC    H'4Ø96'
SAVEAREA DS    18F
         SPACE
*************************************************************************
*        MAINLINE                                                       *
*************************************************************************
         SPACE
MAIN     DS    ØH
         BAS   SUBREG,READ                  READ CARD VIA SYSIN
         BAS   SUBREG,MQCONN                CONNECT MQSERIES
         BAS   SUBREG,MQOPEN                OPEN THE QUEUE
         BAS   SUBREG,MQPUT                 PUT QUEUE
         BAS   SUBREG,MQCLOSE               CLOSE THE QUEUE
         BAS   SUBREG,MQDISC                DISCONNECT FROM MQSERIES
```

```
         SPACE
MAINTERM DS    ØH
         L     R13,SAVEAREA+4
         LM    R14,R12,12(R13)              RESTORE CALLERS REGISTER
         XR    R15,R15                      CLEAR REGISTER 15
         BR    R14                          RETURN TO CALLER
         SPACE
*********************************************************************
*        READ CONTROL RECORD FROM SYSIN                             *
*********************************************************************
         SPACE
READ     EQU   *
         MVC   TRACE,=CL8'READ'
         WTO   'CRMQTEØ1: READ'
         OPEN  (SYSIN,(INPUT))
         GET   SYSIN,IOAREA
         CLC   IOAREA(4),=CL4'PROD'
         BE    READ1ØØØ
         CLC   IOAREA(4),=CL4'SYST'
         BE    READ1ØØØ
         CLC   IOAREA(4),=CL4'TEST'
         BE    READ1ØØØ
         CLC   IOAREA(4),=CL4'VPRD'
         BE    READ1ØØØ
         B     READ_ERROR
         SPACE
READ1ØØØ EQU   *
         MVC   QMGRNAME,BLANKS              USE DEFAULT QUEUE MANAGER
         MVC   MQUEUE+5(4),IOAREA           MQ QUEUE
         MVC   MQUEUER+5(4),IOAREA          MQ REPLY TO QUEUE
         CLOSE (SYSIN)
         MVI   OPEN,C'Y'
         WTOE  'SYSIN=',IOAREA
         BR    SUBREG
         SPACE
READ_ERROR     EQU *
         WTOE  'INVALID PARAMETER SYSIN=',IOAREA
         BR    SUBREG
         SPACE
SYSIN    DCB   DDNAME=MQTESTIN,DSORG=PS,MACRF=GM,EODAD=READ_ERROR
         SPACE
IOAREA   DS    CL8Ø' '
         EJECT
*********************************************************************
*        CONNECT TO MQ MANAGER                                      *
*********************************************************************
         SPACE
MQCONN   EQU   *
         MVC   TRACE,=CL8'MQCONN'
         WTO   'CRMQTEØ1: MQCONN'
         XC    HCONN,HCONN
```

```
         CALL  MQCONN,                                                 +
               (QMGRNAME,HCONN,COMPCODE,REASON),                       +
               MF=(E,CALLLIST),VL
         LA    RØ,MQCC_OK
         C     RØ,COMPCODE
         BER   SUBREG
         WTO   'CRMQTEØ1: ERROR DURING MQCONN!'
         BR    SUBREG
         EJECT
*********************************************************************
*        CLOSE A QUEUE                                              *
*********************************************************************
         SPACE
MQCLOSE  EQU   *
         MVC   TRACE,=CL8'MQCLOSE'
         WTO   'CRMQTEØ1: MQCLOSE'
         LA    RØ,MQCO_NONE
         ST    RØ,OPTIONS
         CALL  MQCLOSE,                                                +
               (HCONN,HOBJ,OPTIONS,COMPCODE,REASON),                   +
               MF=(E,CALLLIST),VL
         LA    RØ,MQCC_OK
         C     RØ,COMPCODE
         BER   SUBREG
         WTO   'CRMQTEØ1: ERROR DURING MQCLOSE!'
         BR    SUBREG
         EJECT
*********************************************************************
*        DISCONNECT FROM MQ MANAGER                                 *
*********************************************************************
         SPACE
MQDISC   EQU   *
         MVC   TRACE,=CL8'MQDISC'
         WTO   'CRMQTEØ1: MQDISC'
         CALL  MQDISC,                                                 +
               (HCONN,COMPCODE,REASON),                                +
               MF=(E,CALLLIST),VL
         LA    RØ,MQCC_OK
         C     RØ,COMPCODE
         BER   SUBREG
         WTO   'CRMQTEØ1: ERROR DURING MQDISC!'
         BR    SUBREG
         EJECT
*********************************************************************
*        GET A MESSAGE FROM A QUEUE                                 *
*********************************************************************
         SPACE
MQGET    EQU   *
         MVC   TRACE,=CL8'MQGET'
         WTO   'CRMQTEØ1: MQGET'
         LA    RØ,MYMD                 MVCL TO-OPERAND
```

```
         LA    R1,MYMD_LENGTH           MVCL TO-LENGTH(MAXIMUM)
         LA    R2,MQMD                  MVCL FROM-OPERAND
         LA    R3,MQMD_LENGTH           MVCL FROM-LENGTH
         MVCL  RØ,R2                    MVCL
         MVC   MYGMO_AREA,MQGMO_AREA
         MVC   FLENGTH,=AL4(QEND-QSTRT)
         CALL  MQGET,                                                  +
               (HCONN,HOBJ,MYMD,MYGMO,FLENGTH,QUEUECO,FLENGTH,         +
               COMPCODE,REASON),                                       +
               MF=(E,CALLLIST),VL
         LA    RØ,MQCC_OK
         C     RØ,COMPCODE
         BER   SUBREG
         WTO   'CRMQTEØ1: ERROR DURING MQGET!'
         BR    SUBREG
         EJECT
*********************************************************************
*        OPEN A QUEUE                                               *
*********************************************************************
         SPACE
MQOPEN   EQU   *
         MVC   TRACE,=CL8'MQOPEN'
         WTO   'CRMQTEØ1: MQOPEN'
         LA    RØ,MYOD                  MVCL TO-OPERAND
         LA    R1,MYOD_LENGTH           MVCL TO-LENGTH(MAXIMUM)
         LA    R2,MQOD                  MVCL FROM-OPERAND
         LA    R3,MQOD_LENGTH           MVCL FROM-LENGTH
         MVCL  RØ,R2                    MVCL
         LA    RØ,MQOO_OUTPUT
         ST    RØ,OPTIONS
         LA    RØ,MQOT_Q
         ST    RØ,MYOD_OBJECTTYPE
         MVC   OBJECTNAME,MQUEUE
         MVC   MYOD_OBJECTNAME,OBJECTNAME
         CALL  MQOPEN,                                                 +
               (HCONN,MYOD,OPTIONS,HOBJ,COMPCODE,REASON),              +
               MF=(E,CALLLIST),VL
         LA    RØ,MQCC_OK
         C     RØ,COMPCODE
         BER   SUBREG
         WTO   'CRMQTEØ1: ERROR DURING MQOPEN!'
         BR    SUBREG
         EJECT
*********************************************************************
*        PUT A MESSAGE ON A QUEUE                                   *
*********************************************************************
         SPACE
MQPUT    EQU   *
         MVC   TRACE,=CL8'MQPUT'
         WTO   'CRMQTEØ1: MQPUT'
         LA    RØ,MYMD                  MVCL TO-OPERAND
```

```
         LA    R1,MYMD_LENGTH           MVCL TO-LENGTH(MAXIMUM)
         LA    R2,MQMD                  MVCL FROM-OPERAND
         LA    R3,MQMD_LENGTH           MVCL FROM-LENGTH
         MVCL  RØ,R2                    MVCL
*
         MVC   MYPMO_AREA,MQPMO_AREA
*
         MVC   FLENGTH,=AL4(QEND-QSTRT)
*
         MVC   MYMD_CORRELID,MQCI_NEW_SESSION
         MVC   MYMD_FORMAT,=CL8'MQCICS'
*
         MVC   MYCIH_AREA,MQCIH_AREA
         MVC   MYCIH_FORMAT,=C'CSQCBDCI'
         MVC   MYCIH_LINKTYPE,MQCLT_TRANSACTION
         MVC   MYCIH_TRANSACTIONID,TESTTRAN
*
         CALL  MQPUT,                                                  +
               (HCONN,HOBJ,MYMD,MYPMO,                                 +
               FLENGTH,MYCIH_AREA,COMPCODE,REASON),                    +
               MF=(E,CALLLIST),VL
         LA    RØ,MQCC_OK
         C     RØ,COMPCODE
         BER   SUBREG
         WTO   'CRMQTEØ1: ERROR DURING MQPUT!'
         BR    SUBREG
         EJECT
*********************************************************************
*        DYNAMIC STORAGE                                            *
*********************************************************************
         SPACE
MQUEUE   DC    CL48'CICS.XXXX.ISBSTCX.FROM.BATCH.TEST.QØØ1'
MQUEUER  DC    CL48'CICS.XXXX.ISBSTCX.FROM.BATCH.TEST.QØØ1.REPLY'
CICSID   DS    CL4
         SPACE
         CMQA   LIST=YES               EQUATES FOR MQ CONSTANTS
         CMQODA DSECT=NO,LIST=YES      OBJECT DESCRIPTER
         CMQMDA DSECT=NO,LIST=YES      MESSAGE DESCRIPTOR
         CMQPMOA DSECT=NO,LIST=YES     PUT MESSAGE OPTIONS
         CMQGMOA DSECT=NO,LIST=YES     GET MESSAGE OPTIONS
         CMQCIHA DSECT=NO,LIST=YES     CICS INFORMATION HEADER
         SPACE
PARM     DS    F
LOADADDR DC    A(Ø)
         SPACE
*********************************************************************
*        PROGRAM CONSTANTS                                          *
*********************************************************************
         SPACE
BLANKS         DC   256C' '
```

```
NULLS           DC   Ø16X'ØØ'
ZEROS           DC   Ø16C'Ø'
*
DATETIME        DS   CL16
LENGTH          DS   H
FLENGTH         DS   F
TRACE           DS   CL8
WORK            DS   CL16
OPEN            DS   CL1
*
MQCI_NEW_SESSION  DC X'414D51214E45575F53455353494F4E5F434F5252454C4944X
                '
*
                DS   ØD
EYEC1           DC   CL8'**COCO**'
COMPCODE        DS   F               COMPLETION CODE
REASON          DS   F               REASON CODE QUALIFYING COMPCODE
HCONN           DS   F               CONNECTION HANDLE
HOBJ            DS   F               OBJECT HANDLE
EYEC2           DC   CL8'**CPCP**'
QMGRNAME        DS   CL48            QUEUE MANAGER NAME
OBJECTNAME      DS   CL48            OBJECT NAME (QUEUE NAME)
OPTIONS         DS   F               COMMAND OPTIONS
TESTTRAN        DC   C'TØØØ'         CICS TRANSACTION
CALLLIST        CALL ,(Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø),VL,MF=L
          SPACE
MYOD      CMQODA  LIST=YES           OBJECT DESCRIPTOR
MYMD      CMQMDA  LIST=YES           MESSAGE DESCRIPTOR
MYPMO     CMQCIHA LIST=YES           PUT MESSAGE OPTIONS FOR CICS
MYGMO     CMQCIHA LIST=YES           GET MESSAGE OPTIONS FOR CICS
          SPACE
QSTRT     DS   ØC                    "COMPLETE CICS HEADER"
MYCIH     CMQCIHA LIST=YES           CICS INFORMATION HEADER
QUEUECO   DC   C'TESTQUEUE FOR CRMQTEØ1'
QEND      DS   ØC
          SPACE
          DFHREGS
          SPACE
SUBREG    EQU  R9                    BAS REGISTER
CODEREG   EQU  R1Ø                   BASIS REGISTER
CODEREG2  EQU  R11                   BASIS REGISTER 2
          SPACE
          DC   C' '
          END  CRMQTEØ1
```

JCL

```
//USERIDX JOB ØØ2665,'YOUR NAME',NOTIFY=USERID,
JOB456Ø9
```

```
//*-------------------------------------------------------------------
//*   JOB  SUBMITTED FROM USERID.MAIN.JCL(CRMQTE01)
//*   DOC: TEST "CICS AND MQSERIES VIA THE 3270 BRIDGE"
//*   GRP:
//*   DATE: 11.06.02, TIME: 09:56
//*-------------------------------------------------------------------
//             CLASS=T,USER=USERID,MSGCLASS=X,REGION=4M,RESTART=*
//*-------------------------------------------------------------------
/*ROUTE PRINT U28
//*-----------------------------------------------------*
//CRMQTE01 EXEC PGM=CRMQTE01
//STEPLIB  DD DSN=USERID.MAIN.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//MESSAGE  DD SYSOUT=*
//MQTESTIN DD *
SYST
/*
```

Claus Reis
CICS Systems Programmer
Nuernberger Lebensversicherung AG (Germany)

CICS questions and answers

Q I know I shouldn't, but I need to address the CSA. EXEC CICS ADDRESS CSA returns a pointer to Fetch Protected Storage. How else can I get the CSA address?

A To address the CSA in CICS/ESA Version 3.2.1 and higher use the following:

```
DFHAFCD TYPE=LOCATE L R15,8(0,R15)
```

R15 now points at the CSA.

If you have any CICS-related questions, please send them in and we will do our best to find answers. Alternatively, e-mail them directly to cicsq@xephon.net.

Flynet has launched a new development, integration, and application server suite based on Intelligent Environments' Integrator technology and called iE Integrator. The single server platform simplifies the integration of numerous disparate systems into one central point, and allows Web developers access through standard object and XML interfaces, including .Net Web services.

The suite comprises a development environment, integration server, and management console optimized for building and running applications that connect to multiple environments at high speeds. An integration server is an application server with a wide range of connectivity, distributed object support, and high-volume transaction processing capabilities. It is designed to make it easy to access and encapsulate any existing enterprise data and applications. It provides high performance, robust, and secure access to mainframe and midrange systems and makes the information available to any front end, including Web browsers. This can be achieved with CICS, MQ Series, 3270/5250 scraping, and optimized DB2 calls.

In addition to connecting legacy systems to HTML Web pages, the suite can be used to access VSAM files of customer data (for example) and process the information so that Visual Basic programs can access it as an ActiveX control, Java programs as a JavaBean, and Microsoft .Net applications access via a .Net style Web service (SOAP, SMO, WSDL/WSML, etc.). Internal data can be stored in a number of formats including XML for transformation using XSLT. A Web team producing end-user services can work without needing a detailed understanding of the host systems involved.

For further information contact:
Flynet Limited, King William House, The Causeway, Burwell, Cambs CB5 0DU, UK.
Tel: (01638) 611111.
URL: http://www.flynet.co.uk.

* * *

Computer Associates has begun shipping Version 1.1 of its Advantage EDBC for ODBC and JDBC-based access to mainframe data from Linux, Unix, and Windows systems.

Enhancements to Version 1.1 include native 2.1-compliant support for JDBC, support for Windows 2000/XP, Unix, Linux, and z/OS platforms, plus improved performance, fault-tolerance, and scalability and additional support for customers' existing security facilities.

The software operates as a multi-threaded server to manage mainframe I/O requests from networked clients, interfacing with existing mainframe security to provide concurrent access to native CICS/VSAM, VSAM, IMS, DB2, Advantage CA-IDMS, and Advantage CA-Datacom data sources. For VSAM and IMS, there's an optimized mainframe SQL engine to process requests with optimum efficiency. For relational DBMS data sources, it exploits native SQL engines while negotiating any dialect differences.

It's out now for z/OS, OS/390 and MVS/ESA servers and on AIX, HP-UX, Linux, Solaris, and Windows (9*x*, NT, 2000, and XP) for the ODBC/JDBC client.

For further information contact:
Computer Associates, One Computer Associates Plaza, Islandia, NY 11749, USA.
Tel: (631) 342 6000.
URL: http://www3.ca.com/Solutions/Collateral.asp?ID=2209&PID=1272.