202

CICS update

September 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.

Fixing a hung terminal with XMEOUT – revisited

In the May 2002 issue of *CICS Update* (Issue 198, pages 3-10) the resolution of the problem of the terminals using the exit XMEOUT seems to me to be very complex. It's easier to use the CSMT queue, changing the address to another queue (eg CSSL to XZ01) for the middle parameter 'indirectname'. In this way, each message generated in the queue CSMT is sent to the queue XZ01.

This new queue, of type 'intrapartition', is defined with the parameters 'transid' (XZ01) and 'triggerlevel' (1). This indicates that the transaction XZ01 is started by each message that is generated in the queue XZ01. The program, started by the transaction, reads queue XZ01 and, depending on the message, executes an appropriate action, and writes a message to the queue CICS CSSL containing text that the CICS output extrapartition MSGUSR does not change.

We have used this system with Version 1.5 of CICS, to intercept DFHAC2236 messages from abending transactions, and, depending on the type of abend and/or transaction name, performed the appropriate action, and sent a message to the CICS master terminal operator about the transaction abend – this might be to notify to the programmers, to ignore them, or even to restart the transaction in some instances.

In the case of terminals, the program would execute an 'EXEC CICS SET TERMINAL (....) REL .....', upon intercepting a message DFHZC2411.

Definition of the CSMT queue:

```
OBJECT CHARACTERISTICS                          CICS RELEASE = 0530
 CEDA  View TDqueue( CSMT )
  TDqueue        : CSMT
  Group          : DCTMSG1
  DEscription    :
  TYPE           : INDirect          Extra | INTra | INDirect
 EXTRA PARTITION PARAMETERS
       -
       -
       -
 INDIRECT PARAMETERS
  Indirectname   : XZ01
```

```
   REMOTE PARAMETERS
         -
         -
         -
```

Definition of the XZ01 queue:

```
OBJECT CHARACTERISTICS                          CICS RELEASE = Ø53Ø
   CEDA  View TDqueue( CSMT )
    TDqueue        : XZØ1
    Group          : DCTMSG1
    DEscription    :
    TYPE           : INTra          Extra | INTra | INDirect
   EXTRA PARTITION PARAMETERS
         -
         -
         -
INTRA PARTITION PARAMETERS
    Atifacility    : File             Terminal | File | System
    RECOVstatus    : No               No | Physical | Logical
    Facilityid     :
    TRAnsid        : XZØ1
    TRIggerlevel   : ØØØØ1            Ø-32767
    Userid         : CICSDCT
         -
         -
         -
```

The program is written in COBOL. When compiling the program, the translator step must have the option ‘SP’ to permit the use of EXEC CICS SET ....

Below is the source for three programs and a copy (RESPCICS) showing how the messages appear to the operator.

The queue XZ01 with triggerlevel 1 starts the transaction XZ01 and program PXZOP40 for each message.

The program PXZOP40 reads the queue XZ01, and, if the message is an ABEND, writes it to the queues XZ02 and XZ03.

For some ABEND codes (table LL - TRANSACTIONS), it restarts the transaction immediately.

In any case, the message is written to the queue CSSL.

The transaction XZ02 starts the program PXZOP45, which sends the message to the operator’s terminal. The operator’s terminal is found

from the file VSTABLAS.

The transaction XZ03 starts the program PXZOP25, which writes the messages to a TD extrapartition queue together with messages from other applications.

We divided the application into three programs for two reasons – to avoid an enqueue, and so that the transactions XZ02 and XZ03 can be used for other types of message from other applications.

PXZOP40

```
*      POSSIBLE VALUES FOR FIELD EIBRESP
*      CICS T/S V1.3.Ø           ABRIL-2ØØØ
 Ø1  RESPCICS.
     Ø3  RESP-OK                 PIC S9(8) COMP VALUE +Ø.
*
     Ø3  RESP-ERROR              PIC S9(8) COMP VALUE +1.
     Ø3  RESP-RDATT              PIC S9(8) COMP VALUE +2.
     Ø3  RESP-WRBRK              PIC S9(8) COMP VALUE +3.
     Ø3  RESP-EOF                PIC S9(8) COMP VALUE +4.
     Ø3  RESP-EODS               PIC S9(8) COMP VALUE +5.
     Ø3  RESP-EOC                PIC S9(8) COMP VALUE +6.
     Ø3  RESP-INBFMH             PIC S9(8) COMP VALUE +7.
     Ø3  RESP-ENDINPT            PIC S9(8) COMP VALUE +8.
     Ø3  RESP-NOVAL              PIC S9(8) COMP VALUE +9.
*
     Ø3  RESP-NOSTART            PIC S9(8) COMP VALUE +1Ø.
     Ø3  RESP-TERMIDERR          PIC S9(8) COMP VALUE +11.
     Ø3  RESP-FILENOTFOUND       PIC S9(8) COMP VALUE +12.
     Ø3  RESP-NOTFND             PIC S9(8) COMP VALUE +13.
     Ø3  RESP-DUPREC             PIC S9(8) COMP VALUE +14.
     Ø3  RESP-DUPKEY             PIC S9(8) COMP VALUE +15.
     Ø3  RESP-INVREQ             PIC S9(8) COMP VALUE +16.
     Ø3  RESP-IOERR              PIC S9(8) COMP VALUE +17.
     Ø3  RESP-NOSPACE            PIC S9(8) COMP VALUE +18.
     Ø3  RESP-NOTOPEN            PIC S9(8) COMP VALUE +19.
*
     Ø3  RESP-ENDFILE            PIC S9(8) COMP VALUE +2Ø.
     Ø3  RESP-ILLOGIC            PIC S9(8) COMP VALUE +21.
     Ø3  RESP-LENGERR            PIC S9(8) COMP VALUE +22.
     Ø3  RESP-QZERO              PIC S9(8) COMP VALUE +23.
     Ø3  RESP-SIGNAL             PIC S9(8) COMP VALUE +24.
     Ø3  RESP-QBUSY              PIC S9(8) COMP VALUE +25.
     Ø3  RESP-ITEMERR            PIC S9(8) COMP VALUE +26.
     Ø3  RESP-PGMIDERR           PIC S9(8) COMP VALUE +27.
     Ø3  RESP-TRANSIDERR         PIC S9(8) COMP VALUE +28.
```

```
           03  RESP-ENDDATA            PIC S9(8) COMP VALUE +29.
      *
           03  RESP-INVTSREQ           PIC S9(8) COMP VALUE +30.
           03  RESP-EXPIRED            PIC S9(8) COMP VALUE +31.
           03  RESP-RETPAGE            PIC S9(8) COMP VALUE +32.
           03  RESP-RTEFAIL            PIC S9(8) COMP VALUE +33.
           03  RESP-RTESOME            PIC S9(8) COMP VALUE +34.
           03  RESP-TSIOERR            PIC S9(8) COMP VALUE +35.
           03  RESP-MAPFAIL            PIC S9(8) COMP VALUE +36.
           03  RESP-INVERRTERM         PIC S9(8) COMP VALUE +37.
           03  RESP-INVMPSZ            PIC S9(8) COMP VALUE +38.
           03  RESP-IGREQID            PIC S9(8) COMP VALUE +39.
      *
           03  RESP-OVERFLOW           PIC S9(8) COMP VALUE +40.
           03  RESP-INVLDC             PIC S9(8) COMP VALUE +41.
           03  RESP-NOSTG              PIC S9(8) COMP VALUE +42.
           03  RESP-JIDERR             PIC S9(8) COMP VALUE +43.
           03  RESP-QIDERR             PIC S9(8) COMP VALUE +44.
           03  RESP-NOJBUFSP           PIC S9(8) COMP VALUE +45.
           03  RESP-DSSTAT             PIC S9(8) COMP VALUE +46.
           03  RESP-SELNERR            PIC S9(8) COMP VALUE +47.
           03  RESP-FUNCERR            PIC S9(8) COMP VALUE +48.
           03  RESP-UNEXPIN            PIC S9(8) COMP VALUE +49.
      *
           03  RESP-NOPASSBKRD         PIC S9(8) COMP VALUE +50.
           03  RESP-NOPASSBKWR         PIC S9(8) COMP VALUE +51.
      *                                                  +52
           03  RESP-SYSIDERR           PIC S9(8) COMP VALUE +53.
           03  RESP-ISCINVREQ          PIC S9(8) COMP VALUE +54.
           03  RESP-ENQBUSY            PIC S9(8) COMP VALUE +55.
           03  RESP-ENVDEFERR          PIC S9(8) COMP VALUE +56.
           03  RESP-IGREQCD            PIC S9(8) COMP VALUE +57.
           03  RESP-SESSIONERR         PIC S9(8) COMP VALUE +58.
           03  RESP-SYSBUSY            PIC S9(8) COMP VALUE +59.
      *
           03  RESP-SESSBUSY           PIC S9(8) COMP VALUE +60.
           03  RESP-NOTALLOC           PIC S9(8) COMP VALUE +61.
           03  RESP-CBIDERR            PIC S9(8) COMP VALUE +62.
           03  RESP-INVEXITREQ         PIC S9(8) COMP VALUE +63.
           03  RESP-INVPARTNSET        PIC S9(8) COMP VALUE +64.
           03  RESP-INVPARTN           PIC S9(8) COMP VALUE +65.
           03  RESP-PARTNFAIL          PIC S9(8) COMP VALUE +66.
      *                                                  +67
      *                                                  +68
           03  RESP-USERIDERR          PIC S9(8) COMP VALUE +69.
      *
           03  RESP-NOTAUTH            PIC S9(8) COMP VALUE +70.
           03  RESP-VOLIDERR           PIC S9(8) COMP VALUE +71.
           03  RESP-SUPPRESSED         PIC S9(8) COMP VALUE +72.
      *                                                  +73
```

```
*                                                    +74
     Ø3  RESP-RESIDERR           PIC S9(8) COMP VALUE +75.
*                                                    +76
*                                                    +77
*                                                    +78
*                                                    +79
*
     Ø3  RESP-NOSPOOL            PIC S9(8) COMP VALUE +8Ø.
     Ø3  RESP-TERMERR            PIC S9(8) COMP VALUE +81.
     Ø3  RESP-ROLLEDBACK         PIC S9(8) COMP VALUE +82.
     Ø3  RESP-END                PIC S9(8) COMP VALUE +83.
     Ø3  RESP-DISABLED           PIC S9(8) COMP VALUE +84.
     Ø3  RESP-ALLOCERR           PIC S9(8) COMP VALUE +85.
     Ø3  RESP-STRELERR           PIC S9(8) COMP VALUE +86.
     Ø3  RESP-OPENERR            PIC S9(8) COMP VALUE +87.
     Ø3  RESP-SPOLBUSY           PIC S9(8) COMP VALUE +88.
     Ø3  RESP-SPOLERR            PIC S9(8) COMP VALUE +89.
*
     Ø3  RESP-NODEIDER           PIC S9(8) COMP VALUE +9Ø.
     Ø3  RESP-TASKIDERR          PIC S9(8) COMP VALUE +91.
     Ø3  RESP-TCIDERR            PIC S9(8) COMP VALUE +92.
     Ø3  RESP-DSNNOTFOUND        PIC S9(8) COMP VALUE +93.
     Ø3  RESP-LOADING            PIC S9(8) COMP VALUE +94.
     Ø3  RESP-MODELIDERR         PIC S9(8) COMP VALUE +95.
     Ø3  RESP-OUTDESCERR         PIC S9(8) COMP VALUE +96.
     Ø3  RESP-PARTNERIDERR       PIC S9(8) COMP VALUE +97.
     Ø3  RESP-PROFILEIDERR       PIC S9(8) COMP VALUE +98.
     Ø3  RESP-NETNAMEIDERR       PIC S9(8) COMP VALUE +99.
*
     Ø3  RESP-LOCKED             PIC S9(8) COMP VALUE +1ØØ.
     Ø3  RESP-RECORDBUSY         PIC S9(8) COMP VALUE +1Ø1.
     Ø3  RESP-UOWNOTFOUND        PIC S9(8) COMP VALUE +1Ø2.
     Ø3  RESP-UOWLNOTFOUND       PIC S9(8) COMP VALUE +1Ø3.
     Ø3  RESP-LINKABEND          PIC S9(8) COMP VALUE +1Ø4.
     Ø3  RESP-CHANGED            PIC S9(8) COMP VALUE +1Ø5.
     Ø3  RESP-PROCESSBUSY        PIC S9(8) COMP VALUE +1Ø6.
     Ø3  RESP-ACTIVITYBUSY       PIC S9(8) COMP VALUE +1Ø7.
     Ø3  RESP-PROCESSERR         PIC S9(8) COMP VALUE +1Ø8.
     Ø3  RESP-ACTIVITYERR        PIC S9(8) COMP VALUE +1Ø9.
*
     Ø3  RESP-CONTAINERERR       PIC S9(8) COMP VALUE +11Ø.
     Ø3  RESP-EVENTERR           PIC S9(8) COMP VALUE +111.
     Ø3  RESP-TOKENERR           PIC S9(8) COMP VALUE +112.
     Ø3  RESP-NOTFINISHED        PIC S9(8) COMP VALUE +113.
     Ø3  RESP-POOLERR            PIC S9(8) COMP VALUE +114.
     Ø3  RESP-TIMERERR           PIC S9(8) COMP VALUE +115.
     Ø3  RESP-SYMBOLERR          PIC S9(8) COMP VALUE +116.
     Ø3  RESP-TEMPLATERR         PIC S9(8) COMP VALUE +117.
*
```

```
       IDENTIFICATION DIVISION.
      *
      * MESSAGES  FOR THE OPERATOR
      *
       PROGRAM-ID. PXZ9Ø4Ø.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
           COPY RESPCICS.
       Ø1  VS-TREBALL.
           Ø3  VS-LLARG                PIC S999   COMP VALUE +8Ø.
           Ø3  VS-DADES      VALUE SPACES.
               Ø5  VS-CLAU                 PIC X(8).
               Ø5  VS-LINK                 PIC X(6).
               Ø5  VS-PANT                 PIC X(4).
               Ø5  FILLER                  PIC X(7).
               Ø5  VS-IMPR                 PIC X(4).
               Ø5  FILLER                  PIC X(51).
       Ø1  CAMPS-DE-DADES.
           Ø3  W-DADES                 PIC X(136)     VALUE SPACES.
      *                                 ******  CICS VERSIO 3.3.Ø *****
           Ø3  W-33Ø-DADES   REDEFINES  W-DADES.
               Ø5  W-33Ø-NUM-MISS             PIC X(9).
               Ø5  FILLER                     PIC X.
               Ø5  W-33Ø-DATA-I-HORA          PIC X(17).
               Ø5  FILLER                     PIC X.
               Ø5  W-33Ø-APLI-TRANS-I-ABEND.
                   Ø7  W-33Ø-APLICACIO        PIC X(8).
                   Ø7  FILLER                 PIC X(13).
                   Ø7  W-33Ø-COD-TRANS        PIC X(4).
                   Ø7  FILLER                 PIC X(7).
                   Ø7  W-33Ø-COD-ABEND        PIC X(4).
               Ø5  FILLER                     PIC X(72).
       Ø1  CAMPS-TREBALL.
           Ø3  W-APLI     VALUE SPACES.
               Ø5  FILLER              PIC X(5).
               Ø5  W-FI-APLI           PIC X(3).
           Ø3  STRT-CICS               PIC X(2)  VALUE SPACES.
           Ø3  SYS-CICS                PIC X(4)  VALUE SPACES.
           Ø3  W-DADES-BONES     VALUE SPACES.
               Ø5  W-NUM-MISS          PIC X(9).
               Ø5  W-APLI-TRANS-I-ABEND.
                   Ø7  W-APLI-MISS     PIC X(8).
                   Ø7  W-LLETR-TRANS   PIC X(11).
                   Ø7  W-COD-TRANS     PIC X(4).
                   Ø7  W-LLETR-ABEND   PIC X(7).
                   Ø7  W-COD-ABEND     PIC X(4).
```

```
*
     Ø3  W-LLARG                 PIC S999  COMP VALUE +Ø.
     Ø3  W-CUA                   PIC X(4)   VALUE 'XZØ1'.
     Ø3  W-CUA-2                 PIC X(4)   VALUE 'XZØ2'.
     Ø3  W-CUA-3                 PIC X(4)   VALUE 'XZØ3'.
     Ø3  W-TRANS-2               PIC X(4)   VALUE 'XZØ2'.
     Ø3  W-TRANS-3               PIC X(4)   VALUE 'XZØ3'.
     Ø3  W-START                 PIC 9      VALUE Ø.
*
* LIST OF TRANSACTIONS FOR REARRANC IN CASE OF ABENDS '-91*'
*
* TE ESPAI PER A 3Ø TRANSACCIONS.
* PER EIXAMPLAR CANVIAR EL 'OCCURS' DE BAIX
*
*
 Ø1  LLISTA-TRANSACCIONS.
     Ø3  FILLERØ1                PIC X(4)   VALUE 'EPP1'.
     Ø3  FILLERØ2                PIC X(4)   VALUE 'EPP2'.
     Ø3  FILLERØ3                PIC X(4)   VALUE 'EPP3'.
     Ø3  FILLERØ4                PIC X(4)   VALUE 'EPP4'.
     Ø3  FILLERØ5                PIC X(4)   VALUE 'EPP5'.
     Ø3  FILLERØ6                PIC X(4)   VALUE 'EPP8'.
     Ø3  FILLERØ7                PIC X(4)   VALUE 'EPP9'.
     Ø3  FILLERØ8                PIC X(4)   VALUE 'EPPG'.
     Ø3  FILLERØ9                PIC X(4)   VALUE 'EPPH'.
     Ø3  FILLER1Ø                PIC X(4)   VALUE 'EC1Ø'.
     Ø3  FILLER11                PIC X(4)   VALUE 'EC2Ø'.
     Ø3  FILLER12                PIC X(4)   VALUE 'EC25'.
     Ø3  FILLER13                PIC X(4)   VALUE 'EP3Ø'.
     Ø3  FILLER14                PIC X(4)   VALUE 'EP35'.
     Ø3  FILLER15                PIC X(4)   VALUE 'EP4Ø'.
     Ø3  FILLER16                PIC X(4)   VALUE 'EP5Ø'.
     Ø3  FILLER17                PIC X(4)   VALUE 'EP9Ø'.
     Ø3  FILLER18                PIC X(4)   VALUE 'MVDB'.
     Ø3  FILLER19                PIC X(4)   VALUE 'EM65'.
     Ø3  FILLER2Ø                PIC X(4)   VALUE 'EM66'.
     Ø3  FILLER21                PIC X(4)   VALUE 'SA24'.
     Ø3  FILLER22                PIC X(4)   VALUE '    '.
     Ø3  FILLER23                PIC X(4)   VALUE '    '.
     Ø3  FILLER24                PIC X(4)   VALUE '    '.
     Ø3  FILLER25                PIC X(4)   VALUE '    '.
     Ø3  FILLER26                PIC X(4)   VALUE '    '.
     Ø3  FILLER27                PIC X(4)   VALUE '    '.
     Ø3  FILLER28                PIC X(4)   VALUE '    '.
     Ø3  FILLER29                PIC X(4)   VALUE '    '.
     Ø3  FILLER3Ø                PIC X(4)   VALUE '    '.
 Ø1  LL-TRANSACCIONS REDEFINES LLISTA-TRANSACCIONS.
     Ø3  LL-TRANS    OCCURS 3Ø TIMES INDEXED BY T  PIC X(4).
 PROCEDURE DIVISION.
     EXEC CICS ASSIGN APPLID(W-APLI)
```

```
                    SYSID(SYS-CICS)
                    STARTCODE(STRT-CICS)
                    END-EXEC.
     EXEC CICS ENQ RESOURCE(W-CUA) LENGTH(4) END-EXEC.
     MOVE +136 TO W-LLARG.
     EXEC CICS READQ TD QUEUE(W-CUA) INTO(W-DADES)
            NOHANDLE LENGTH(W-LLARG) END-EXEC.
     PERFORM PROCES UNTIL EIBRESP NOT = RESP-OK.
     IF  W-START = 1
         PERFORM VS-VORE-TERMINALS
         EXEC CICS START TRANSID(W-TRANS-2) INTERVAL(1)
              TERMID(VS-PANT) END-EXEC
         EXEC CICS START TRANSID(W-TRANS-3) INTERVAL(1)
              END-EXEC.
     EXEC CICS RETURN END-EXEC.
     GOBACK.
*
 PROCES.
     PERFORM OBTINDRE-DADES-BONES.
     IF  W-NUM-MISS = 'DFH2236I ' OR 'DFHAC2236'
       IF  W-COD-ABEND NOT = 'AZCT'
         MOVE +136 TO W-LLARG
         EXEC CICS WRITEQ TD QUEUE(W-CUA-2) FROM(W-DADES)
                          LENGTH(W-LLARG) END-EXEC
         EXEC CICS WRITEQ TD QUEUE(W-CUA-3) FROM(W-DADES)
                          LENGTH(W-LLARG) END-EXEC
         MOVE 1 TO W-START
         IF  W-COD-ABEND = '-911' OR  W-COD-ABEND = '-913'
           PERFORM VORE-SI-REARRANC
         END-IF
       END-IF
     END-IF.
     MOVE +128 TO W-LLARG.
     EXEC CICS WRITEQ TD QUEUE('CSSL') FROM(W-DADES)
                          LENGTH(W-LLARG) END-EXEC.
     EXEC CICS SYNCPOINT END-EXEC.
*
     EXEC CICS ENQ RESOURCE(W-CUA) LENGTH(4) END-EXEC.
     MOVE SPACES TO W-DADES.
     MOVE +136 TO W-LLARG.
     EXEC CICS READQ TD QUEUE(W-CUA) INTO(W-DADES)
            NOHANDLE LENGTH(W-LLARG) END-EXEC.
*
 VORE-SI-REARRANC.
     SET T TO 1.
     SEARCH LL-TRANS
           WHEN LL-TRANS (T) = W-COD-TRANS
                EXEC CICS START TRANSID(W-COD-TRANS)
                     INTERVAL(1) END-EXEC
           WHEN LL-TRANS (T) = SPACES
```

```
                NEXT SENTENCE.
*
 OBTINDRE-DADES-BONES.
     MOVE  W-33Ø-NUM-MISS   TO  W-NUM-MISS
     UNSTRING W-33Ø-APLI-TRANS-I-ABEND  DELIMITED BY ' '
          INTO W-APLI-MISS  W-LLETR-TRANS
               W-COD-TRANS  W-LLETR-ABEND  W-COD-ABEND.
*
* OBTAIN THE NUMBER OF THE OPERATOR'S TERMINAL
*
 VS-VORE-TERMINALS.
     IF  SYS-CICS = 'CIC1'
         MOVE 'CIC1OPER' TO VS-CLAU
     ELSE
         MOVE 'CIC3OPER' TO VS-CLAU.
     EXEC CICS READ DATASET ('VSTABLAS') INTO(VS-DADES)
          RIDFLD (VS-CLAU) EQUAL
          NOHANDLE LENGTH(VS-LLARG)
          END-EXEC.
* TERMINAL PER DEFECTE
     IF EIBRESP NOT = RESP-OK
         MOVE 'V1ØØ' TO VS-PANT.
```

PXZOP45

```
 IDENTIFICATION DIVISION.
*
* MESSAGES FOR THE OPERATOR
*
 PROGRAM-ID. PXZOP45.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
     COPY RESPCICS.
 Ø1  CAMPS-TREBALL.
     Ø3  W-DADES                  PIC X(136) VALUE SPACES.
     Ø3  W-FI.
         Ø5  FILLER               PIC X(51)  VALUE
         '          <<<<<  LAST MESSAGE FOR A WHILE  >>>>> EN'.
*                                             VALOR X'1DC8'
          Ø5  LLUEN                PIC XX     VALUE 'H'.
         Ø5  W-CICS               PIC X(4) VALUE SPACES.
         Ø5  FILLER               PIC X(79)  VALUE SPACES.
     Ø3  W-LONG                   PIC S999  COMP VALUE +Ø.
     Ø3  W-CUA                    PIC X(4)   VALUE 'XZØ2'.
 PROCEDURE DIVISION.
     EXEC CICS ASSIGN SYSID(W-CICS) END-EXEC.
     PERFORM LLEGIR-CUA.
```

```
           PERFORM PROCES UNTIL EIBRESP NOT = RESP-OK
           MOVE  +136          TO  W-LONG.
           EXEC CICS SEND TEXT FROM(W-FI) LENGTH(W-LONG)
                ERASE JUSLAST FREEKB ALARM
                END-EXEC.
           EXEC CICS RETURN END-EXEC.
           GOBACK.
       PROCES.
           MOVE +136 TO W-LONG.
           EXEC CICS SEND TEXT FROM(W-DADES) LENGTH(W-LONG)
                         ERASE JUSLAST FREEKB ALARM   END-EXEC.
           EXEC CICS RECEIVE LENGTH(W-LONG) END-EXEC.
           PERFORM LLEGIR-CUA.
      *
       LLEGIR-CUA.
           MOVE +136 TO W-LONG.
           MOVE SPACES TO W-DADES.
           EXEC CICS READQ TD QUEUE(W-CUA) INTO(W-DADES)
                NOHANDLE LENGTH(W-LONG) END-EXEC.
```

PXZOP25

```
       IDENTIFICATION DIVISION.
      *
      * MESSAGES FOR THE OPERATOR
      *
       PROGRAM-ID. PXZOP25.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
           COPY RESPCICS.
       77  TEMPS     PIC S9(8) COMP VALUE ZERO.
       Ø1  CAMPS-TREBALL.
           Ø3  W-DADES                   PIC X(136) VALUE SPACES.
           Ø3  W-LONG                    PIC S999  COMP VALUE +136.
           Ø3  W-LONG-OPER               PIC S999  COMP VALUE +16Ø.
           Ø3  W-CUA                     PIC X(4)   VALUE 'XZØ3'.
           Ø3  W-CUA-OPER                PIC X(4)   VALUE 'OPER'.
           Ø3  W-LLINEA.
               Ø5  FILLER              PIC X   VALUE SPACE.
               Ø5  W-DATA              PIC X(1Ø).
               Ø5  FILLER              PIC X   VALUE SPACE.
               Ø5  W-HORA              PIC X(8).
               Ø5  FILLER              PIC X   VALUE SPACES.
               Ø5  W-SEP               PIC XX  VALUE '>>'.
               Ø5  FILLER              PIC X   VALUE SPACES.
               Ø5  W-TEXT              PIC X(136)  VALUE SPACES.
       PROCEDURE DIVISION.
```

```
           PERFORM VORE-DATA-HORA.
           MOVE SPACES TO W-DADES.
           EXEC CICS READQ TD QUEUE(W-CUA) INTO(W-DADES)
                NOHANDLE LENGTH(W-LONG) END-EXEC.
           PERFORM PROCES UNTIL EIBRESP NOT = RESP-OK.
           EXEC CICS SET TDQUEUE(W-CUA-OPER) CLOSED END-EXEC.
           EXEC CICS SET TDQUEUE(W-CUA-OPER) OPEN   END-EXEC.
           EXEC CICS RETURN END-EXEC.
           GOBACK.
       PROCES.
           MOVE SPACES  TO W-TEXT.
           MOVE W-DADES TO W-TEXT.
           EXEC CICS WRITEQ TD QUEUE(W-CUA-OPER) FROM(W-LLINEA)
                LENGTH(W-LONG-OPER)   END-EXEC.
           MOVE SPACES TO W-DADES.
           EXEC CICS READQ TD QUEUE(W-CUA) INTO(W-DADES)
                NOHANDLE LENGTH(W-LONG) END-EXEC.
       VORE-DATA-HORA.
           EXEC CICS ASKTIME ABSTIME(TEMPS) END-EXEC.
           EXEC CICS FORMATTIME ABSTIME(TEMPS)
                DDMMYYYY(W-DATA)  DATESEP('-')
                TIME(W-HORA) TIMESEP(':')
                END-EXEC.
```

Juan Tormo
Systems Manager
Sidmed SA (Spain)

Handling the printout of maps

In 1991 Felix Luke updated a module submitted by Mike Lamkins to handle printout of maps created under CICS 1.7. We recently needed to do a similar task and dug up this program and immediately ran into some problems using it with some of our maps – although others seemed to be handled just fine. After some digging we found out that BMS has gone through several changes since CICS 1.7. We found four additional changes and have updated the program to run with each of these different maps. As our shop has maps compiled under almost every CICS release known, we think that this should work for any shop. We also added the ability to handle extended attributes.

MODIFIED PROGRAM

```
         TITLE 'CONSTRUCT REPORT FROM PRINT MAP'
* FUNCTION : THIS PROGRAM ACCEPTS A COMMAREA CONTAINING THE           *
*            MAPSET AND MAP ID OF A SCREEN, THE SYMBOLIC MAP OF THE   *
*            MAP AND AN OUTPUT AREA.  THE PROGRAM LOADS THE PHYSICAL  *
*            MAP USING THE MAPSET ID SPECIFIED AND MERGES THE         *
*            DEFINITION OF THE MAP SPECIFIED IN THE MAPSET WITH THE   *
*            DATA GIVEN IN THE INPUT SYMBOLIC MAP, TO GENERATE A      *
*            24 X 8Ø REPORT IN THE OUTPUT AREA CONTAINING THE SCREEN  *
*            DATA AS IF THE MAP SPECIFIED WAS SENT OUT TO A TERMINAL  *
*            WITH THE DATA IN THE SYMBOLIC MAP.                       *
* REMARKS  : THIS PROGRAM ASSUMES THAT THE EXTENDED ATTRIBUTES ARE    *
*            NOT USED IN THE BMS MAP.                                 *
* REMARKS  : FIXED FOR EXTENDED ATTRIBUTES.                           *
* REMARKS  : WILL WORK FOR 4 VARIATIONS.                              *
* REMARKS  : WILL ERROR IF UNKNOWN VARIATION.                         *
***************************  COMMAREA  ********************************
         DFHEIGBL
COMDSECT DSECT
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
COMSCNID DS    CL8
COMMAPST DS    CL8
COMRETCD DS    CL4
BMSAREA  DS    CL192Ø
OUTSCRN  DS    CL192Ø
COMLEN   EQU   *-COMDSECT
DATAREG  EQU   R9                  DFHEISTG STORAGE
CODEREG  EQU   R1Ø                 BASE REGISTER
EIBREG   EQU   R11                 EIB BASE REGISTER
GENSCRN  DFHEIENT CODEREG=CODEREG,DATAREG=DATAREG,                    *
               EIBREG=EIBREG
         L     R12,DFHEICAP            GET ADDRESSABILITY TO COMMAREA
         USING COMDSECT,R12            TELL ASSEMBLER ABOUT IT
         MVC   COMRETCD,=CL4'****'     SET RETURN CODE TO GOOD
         CLC   COMMAPST,=CL8' '        IF INPUT MAPSET IS BLANK
```

```
         BNH   MAPERR                     ERROR
         CLC   COMSCNID,=CL8' '        IF INPUT MAP IS BLANK
         BNH   MAPERR                     ERROR
LOADMAP  DS    ØH
         EXEC  CICS LOAD PROGRAM(COMMAPST) SET(R8) NOHANDLE
         CLC   EIBRESP,DFHRESP(NORMAL) IF RESP NOT NORMAL
         BNE   LOADERR                    ERROR
         USING MSMSD,R8                TELL ASM ABOUT ADDRESSABILITY
         CLC   MSSETNM,COMMAPST        IF MAPSET IN MODULE NOT INPUT
         BNE   MAPSTERR                   MAPSET, ERROR
         LA    R2,12(R8)               POINTER R2 TO 1ST MAP IN MAPSET
         USING MSMDI,R2                TELL ASM ABOUT ADDRESSABILITY
         DROP  R8
FINDMAP  DS    ØH
         CLC   MSNAME,COMSCNID         IF CURRENT MAP = INPUT MAP
         BE    FOUNDMAP                   EXIT LOOP
         AH    R2,MSLNGTH              ELSE ADVANCE R2 TO NEXT MAP
         CLI   MSMDI,X'FF'             IF END OF MAPSET
         BE    NOMAPERR                   ERROR
         B     FINDMAP                 LOOP BACK
FOUNDMAP DS    ØH
         SR    R4,R4                   CLEAR SOURCE ADDRESS
         LA    R6,OUTSCRN              DESTINATION IS OUTPUT AREA
         L     R5,SPACES               LENGTH=ZERO; PAD CHAR=SPACE
         LA    R7,192Ø                 LENGTH OF DESTINATION = 192Ø
         MVCL  R6,R4                   CLEAR OUTPUT AREA
         LH    R3,MSDLEN               LOAD LENGTH OF MAP PREFIX
         AR    R3,R2                   BUMP PAST MAP PREFIX TO 1ST FLD
         USING MSMDF,R3                TELL ASM ABOUT ADDRESSABILITY
         LA    R8,BMSAREA+12           POINTS TO FIRST FLD IN SYMB MAP
CHKFIELD DS    ØH
         CLI   MSMDF,X'FF'             IF END OF MAP
         BE    RETURN                     RETURN
         SR    R4,R4                   CLEAR SOURCE ADDRESS
         TM    MSDESC,MSVALUE          DOES FIELD HAVE INITIAL VALUE?
         BNO   CHKVAR                  GO CHECK IF VARIABLE FIELD
         TM    MSATTR,MSDARK           DARK ATTRIBUTE ?
         BO    CHKVAR                  GO CHECK IF VARIABLE FIELD
         LA    R4,MSINIT               POINT R4 TO INITIAL VALUE
         CLC   MSIND7,MSEXTØ           NO EXTRA BYTE BEFORE FIELD?
         BE    CHKVAR                  YES, GO CHECK VAR
         CLC   MSIND7,MSEXT1           1 EXTRA BYTE BEFORE FIELD?
         BE    ADVONE                  YES, GO ADVANCE ONE
         CLC   MSIND7,MSEXT2           2 EXTRA BYTES BEFORE FIELD?
         BE    ADVTWO                  YES, GO ADVANCE TWO
         CLC   MSIND7,MSEXT4           4 EXTRA BYTES BEFORE FIELD?
         BNE   INVMPERR                NO - INVALID MAP FORMAT ERROR
ADVFOUR  LA    R4,4(R4)                ADVANCE R4 BY 4 EXTRA BYTES
         B     CHKVAR
ADVTWO   LA    R4,2(R4)                ADVANCE R4 BY 2 EXTRA BYTES
```

```
         B     CHKVAR
ADVONE   LA    R4,1(R4)              ADVANCE R4 BY 1 EXTRA BYTE
CHKVAR   DS    ØH
         TM    MSDESC,MSVAR          IS THIS A VARIABLE FIELD ?
         BO    CHKBMS                IF YES CHECK BMS AREA
         TM    MSDESC,MSVALOV        IS THIS A VARIABLE FIELD ?
         BNO   CHKPTR                IF NO, SKIP THE FOLLOWING
CHKBMS   CLI   2(R8),X'ØØ'           IS ATTRIBUTE LOW-VALUE ?
         BE    CHKVALUE              IF SO, GO CHECK DATA
         TM    2(R8),MSDARK          DARK ATTRIBUTE ?
         BO    ADVSFLD               IF SO, MOVE NO DATA
CHKVALUE CLC   MSIND7,MSEXT4         DOES MAP HAS EXTENDED ATTR?
         BNE   CHKFLDV               NO, GO CHECK
         LA    R8,4(R8)              ADVANCE R8 BY EXT. ATTR BYTES
CHKFLDV  CLI   3(R8),X'ØØ'           IS VALUE FIELD LOW-VALUE ?
         BE    ADVSFLD               IF SO, TAKE INITIAL VALUE
         LA    R4,3(R8)              ELSE POINT R4 TO VARIABLE DATA
ADVSFLD  LA    R8,3(R8)              ADVANCE R8 BY 3 BYTES (LEN+ATTR)
         AH    R8,MSFLTH             ADVANCE R8 BY FIELD LENGTH
CHKPTR   DS    ØH
         LTR   R4,R4                 IF R4 ZERO (NO DATA TO MOVE)
         BZ    DONTMOVE                 SKIP THE FOLLOWING
         LH    R5,MSFLTH             SET SOURCE LENG TO FIELD LENG
         LR    R7,R5                 SET DEST LENG TO FIELD LENG
         LH    R1,MSPOS              LOAD DISPLACEMENT OF FIELD
         LA    R6,OUTSCRN+1          ADD 1 POSITION FOR THE ATTRIBUTE
         AR    R6,R1                 POINT R6 TO CURRENT POSITION
         MVCL  R6,R4                 MOVE DATA TO OUTPUT, BE IT
*                                     CONSTANT OR VARIABLE
DONTMOVE DS    ØH
         SR    R1,R1                 CLEAR FIELD LENGTH
         TM    MSDESC,MSVALUE        DOES FIELD HAVE INITIAL VALUE?
         BO    ADDLENG               IF YES, ADD LENGTH OF FIELD
         TM    MSDESC,MSVALOV        DOES FIELD HAVE INITIAL VALUE?
         BNO   NOINIT                IF NOT, SKIP
ADDLENG  LH    R1,MSFLTH             LOAD LENGTH OF FIELD
NOINIT   DS    ØH
         CLC   MSIND7,MSEXTØ         NO EXTRA BYTE BEFORE FIELD?
         BE    ADVOTHER              YES, GO ADVANCE OTHER
         CLC   MSIND7,MSEXT1         1 EXTRA BYTE BEFORE FIELD?
         BE    ADVOTH1               YES, GO ADVANCE ONE
         CLC   MSIND7,MSEXT2         2 EXTRA BYTES BEFORE FIELD?
         BE    ADVOTH2               YES, GO ADVANCE TWO
         CLC   MSIND7,MSEXT4         4 EXTRA BYTES BEFORE FIELD?
         BNE   INVMPERR              NO - INVALID MAP FORMAT ERROR
ADVOTH4  LA    R3,4(R3)              ADVANCE R3 BY 4 EXTRA BYTES
         B     ADVOTHER
ADVOTH2  LA    R3,2(R3)              ADVANCE R3 BY 2 EXTRA BYTES
         B     ADVOTHER
ADVOTH1  LA    R3,1(R3)              ADVANCE R3 BY 1 FOR NON EXT
```

```
ADVOTHER DS    ØH
         LA    R3,8(R3)               ADVANCE 8 BYTES FOR OTHER INFO
*                                      (8 BYTES FOR NON-EXTENDED ATTR)
*                                      (12 BYTES IF EXTENDED ATTR)
         AR    R3,R1                  ADV POINTER BY FIELD LENGTH
         B     CHKFIELD               GO BACK TO CHECK NEXT FLD
*        ERROR HANDLING
MAPERR   MVC   COMRETCD,=CL4'*MAP'     NO MAPSET NOR MAP ID GIVEN
         B     RETURN
MAPSTERR MVC   COMRETCD,=CL4'*MST'     MAPSET DOES NOT MATCH THAT IN
         B     RETURN                   LOAD MODULE
LOADERR  MVC   COMRETCD,=CL4'*LOD'     ERROR LOADING MAPSET
         B     RETURN
NOMAPERR MVC   COMRETCD,=CL4'*MNF'     CANNOT FIND MAP IN MAPSET
         B     RETURN
INVMPERR MVC   COMRETCD,=CL4'*INV'     INVALID MAP FORMAT
         B     RETURN
RETURN   DS    ØH
         EXEC CICS RELEASE PROGRAM(COMMAPST) NOHANDLE
         DFHEIRET
         DS    ØF
SPACES   DC    XL4'40000000'
MSEXTØ   DC    X'00000000'    NO EXTRA BYTE BEFORE FIELD
MSEXT1   DC    X'01000000'    1 EXTRA BYTE BEFORE FIELD
MSEXT2   DC    X'01000200'    2 EXTRA BYTES BEFORE FIELD
MSEXT4   DC    X'01020304'    4 EXTRA BYTES BEFORE FIELD
         LTORG
MSMSD    DSECT
MSSETNM  DS     CL8
         DS     CL4
MSMDI    DSECT
MSDLEN   DS     AL2
         DS     CL6
MSNAME   DS     CL8
MSLNGTH  DS     AL2
         DS     CL15
MSIND    DS     AL1            MAP INDICATOR
MSEXTAT  EQU    X'40'          EXTENDED FUNCTION
         DS     CL16
MSIND7   DS     CL4            MAP INDICATOR REL 7 AND AFTER
         DS     CL8
MSMDF    DSECT
         DS     CL2
MSFLTH   DS     AL2
MSDESC   DS     AL1
MSVALOV  EQU    X'03'         INITIAL VALUE OVERRIDABLE
MSVALUE  EQU    X'02'         FIELD HAS INITIAL VALUE
MSVAR    EQU    X'01'         VARIABLE FIELD
MSATTR   DS     CL1
```

```
MSDARK   EQU    X'ØC'        DARK ATTRIBUTE
MSPOS    DS     AL2          FIELD POSITION
MSINIT   DS     ØCL1
         DFHEISTG
         DFHEIEND
         END    GENSCRN
```

Floyd Chrysler
Gerry Chung
Lombard (Canada)

A poor man's CICS chargeback system

BACKGROUND

Many years ago I was asked to develop a type of chargeback system so that my then employer could allocate costs incurred under CICS to the company's computer users. Management decided to allocate costs based on two criteria – total transaction volume and transaction CPU time. Although these criteria are probably insufficient for a really accurate workload assessment (and transaction volume is an especially inaccurate barometer of resource consumption), the data centre was satisfied with this format.

I was not asked to include financial information in the code so that actual costs could be specified. Instead, the company delivered the output from the application to the accounting and finance staff who applied their own cost values to the data. I assumed that this was done by some manual process.

DESCRIPTION OF THE APPLICATION

The application I developed contains one main program and four external subroutines. It uses SMF data (type 110 CICS records) as input and runs as a batch process. I filtered out all CICS monitoring records (subtype 01) from the daily SMF data and fed that data into the application. However, the application is designed to process

comprehensive SMF data and extract only the CICS monitoring records. If your shop doesn't write CICS SMF records, refer to the appropriate IBM documentation for procedures to accomplish this – the process is very easy to set up.

Initially my employers ran the application on a monthly basis so they could invoice the users once per month. However, later on we started running it on a weekly and daily basis so that management could get information on specific activity that had occurred the previous day. In installations that don't have many system tools, this application can be

```
RUN DATE:Ø5/Ø9/Ø2 COST CENTRE GRAND TOTALS
Ø5/Ø3/Ø2 THRU Ø5/Ø6/Ø2 PAGE 1


COST    APPLICATIONS WITHIN          TOTAL NUMBER OF  TOTAL CPU TIME
CENTER      COST CENTER               TRANSACTIONS     (IN SECONDS)

1ØØ1Ø   ABENDAID/FX                          16           .Ø25
10015   AMP DISTRIBUTION REFERRAL         2,Ø61          3.957
1ØØ2Ø   ASI                               41Ø        786
1ØØ3Ø   CARMS                             489           .938
1ØØ4Ø   CBT TRAINING                       22           .Ø42
1ØØ5Ø   CICS/CEMT                       74,555        143.141
1ØØ6Ø   CONSTRUCT                         61Ø          1.168
1ØØ7Ø   DBA                              1,118          2.144
1ØØ8Ø   EDI                                Ø        ØØØ
1ØØ9Ø   FAX NOTIFY                          Ø           .ØØØ
1Ø1ØØ   GENERAL LEDGER                     24           .Ø46
1Ø11Ø   MAINVIEW                        29,172         56.ØØØ
1Ø12Ø   MENU SYSTEM                      9,788         18.79Ø
1Ø13Ø   NATURAL                        158,37Ø        3Ø4.Ø69
1Ø14Ø   NATURAL.DEVELOPMENT              9,497         18.232
1Ø15Ø   NATURAL SECURITY                     6           .ØØ8
1Ø16Ø   OPEN OBLIGATIONS                    38           .Ø72
1Ø17Ø   ORDER.MANAGEMENT               562,22Ø      1,Ø79.46Ø
1Ø18Ø   PREDICT                              Ø           .ØØØ
1Ø19Ø   SYSTEM ARCHIVE AND RETRIEVAL         Ø           .ØØØ
1Ø2ØØ   SUPER.NATURAL                    1,Ø24          1.965
1Ø21Ø   TECH SUPPORT                      358           .684
1Ø99Ø   WHO KNOWS?                          Ø           .ØØØ

     GRAND TOTAL: NUMBER OF TRANSACTIONS  ==>>>      849,778
     GRAND TOTAL: CPU TIME (IN SECONDS)   ==>>>        1,631.527
```

Figure 1: Sample Cost Centre report

RUN DATE: 05/09/02 PRODUCTION CICS COST ALLOCATION
REPORT 05/03/02 THRU 05/06/02 PAGE 1

COST	CICSPAPB		CICSPAPD		CICSPAPF		CICSPAPG		CICSPAPH		CICSVS	
CENTER	TRANCOUNT	CPUTIME	TRANCOUNT	CPUTIME	TRANCOUNT	CPUTIME	TRANCOUNT	CPUTIME	TRANCOUNT	CPUTIME	TRANCOUNT	CPUTIME
10010	2	.000	2	.000	2	.000	2	.000	6	.000	0	.000
10015	0	.000	2,061	.039	0	.000	0	.000	0	.000	0	.000
10020	0	.000	162	.003	43	.000	0	.000	0	.000	205	.003
10030	244	.004	0	.000	0	.000	0	.000	0	.000	245	.004
10040	0	.000	11	.000	0	.000	0	.000	0	.000	11	.000
10050	6,982	.134	7,021	.134	7,005	.134	7,018	.134	7,018	.134	7,023	.134
10060	0	.000	304	.005	0	.000	0	.000	0	.000	304	.005
10070	0	.000	554	.010	0	.000	0	.000	0	.000	554	.010
10080	0	.000	0	.000	0	.000	0	.000	0	.000	0	.000
10090	0	.000	0	.000	0	.000	0	.000	0	.000	0	.000
10100	12	.000	0	.000	0	.000	0	.000	0	.000	12	.000
10110	3,544	.068	2,328	.044	3,552	.068	3,558	.068	2,326	.044	2,330	.044
10120	0	.000	0	.000	0	.000	3,366	.064	2,900	.055	3,139	.060
10130	0	.000	78,761	1.512	0	.000	0	.000	0	.000	78,745	1.511
10140	0	.000	505	.009	0	.000	0	.000	0	.000	505	.009
10150	0	.000	2	.000	0	.000	0	.000	0	.000	2	.000
10160	19	.000	0	.000	0	.000	0	.000	0	.000	19	.000
10170	0	.000	0	.000	0	.000	276,885	5.316	277,969	5.337	20	.000
10180	0	.000	0	.000	0	.000	0	.000	0	.000	0	.000
10190	0	.000	0	.000	0	.000	0	.000	0	.000	0	.000
10200	0	.000	511	.009	0	.000	0	.000	0	.000	513	.009
10210	0	.000	0	.000	2	.000	79	.001	64	.001	213	.004
10990	0	.000	0	.000	0	.000	0	.000	0	.000	0	.000
	10,803	.207	92,222	1.770	10,604	.203	290,908	5.585	290,283	5.573	93,840	1.801

Figure 1: Sample CICS usage report

used as a type of audit tool. The daily reports it produces effectively complement the daily CICS shutdown statistics since they report CICS activity by application/user groups. The reports will clarify which transactions and application groups are consuming the most CPU resources.

The environment at my shop consisted of 18 CICS regions (6 production, 6 Q/A, and 6 test/development). The first report generated by the application provides figures on resource usage for the Production regions. See Figure 1 for a sample of this report. The second and third reports are similar, with the six region names printed across the page. As indicated in Figure 1, the report contains the complete period's transaction and CPU times for each of the regions. The total transaction count for a given region should be equivalent to the number of user transactions reported in the IBM shutdown statistics report for that system.

The application or user groups are categorized into cost centres in the reports. (Later in this article I will describe what you need to do to build your own cost centres.) Figure 2 presents a sample report of cost centre statistics and specifies the application/user groups that make up each cost centre. The figures in this report are cumulative for all CICS regions in your system. Thus, for example, cost centre 10010 had only 16 transactions execute during the reporting period over all 18 regions. By looking at these reports on a frequent basis you can quickly note any changes in activity in your CICS environment and then pursue additional analysis if needed.

The application reports on various errors, such as transaction names that appear in the SMF data but are not defined in the application. You can control the level of errors you want to process by modifying one instruction in program KC903MP. The twelfth instruction following label ERRPRT is CP errcount,=p'50'. That statement implies that after 50 errors the program will terminate with an abend code '1000'.

HOW TO IMPLEMENT THE APPLICATION

How to implement the application:

1 Modify source code in KC900MP (optional) – the program ignores any IBM-supplied transaction names (those starting with C), but,

if you have any transaction names beginning with C that you want to process, add them to the code after label WRITEIT.

2 Modify source code in KC901MP (required) – the cost centres, application/user group text and transaction names must be defined in the COSTCTR macro at the end of the program. Documentation on how to do this is contained as comments at the top of the program. Then, in the TYPE=FINAL statement, specify the number of CICS regions you want to process in the APPLS operand. If you have less than 18 regions you may be able to leave the extra ones already defined in the code. That will save you some coding changes but you'll see empty values for them in the reports.

3 Modify source code in KC902MP (required) – specify your CICS region names following label APPLTABE.

4 Assemble and link edit subroutines KC901MP, KC902MP, KC903MP, and KC904MP using JCL similar to that shown below:

```
//STEP1    EXEC ASMHCL,
//* USE THIS FOR CICS/ESA 4.1.0
//         MAC1='SYS1.MACLIB',
//         MAC='SYSX.CICS410.SDFHMAC',
//         MAC3='SYS1.AMODGEN'
//ASM.SYSPUNCH DD SYSOUT=Z
//ASM.SYSIN DD DISP=SHR,DSN=YOUR.SOURCE.LIBRARY(KC90xMP)
//LKED.SYSLMOD DD DSN=YOUR.LOAD.LIBRARY,DISP=SHR
//LKED.SYSIN DD *
  NAME KC90xMP(R)
//
```

5 Assemble and link edit the main program KC900MP with JCL similar to that shown below:

```
//STEP1    EXEC ASMHCL,
//* USE THIS FOR CICS/ESA 4.1.0
//         MAC1='SYS1.MACLIB',
//         MAC='SYSX.CICS410.SDFHMAC',
//         MAC3='SYS1.AMODGEN'
//ASM.SYSPUNCH DD SYSOUT=Z
//ASM.SYSIN DD DISP=SHR,DSN=YOUR.SOURCE.LIBRARY(KC900MP)
//LKED.SYSLMOD DD DSN=YOUR.LOAD.LIBRARY,DISP=SHR
//LKED.SYSIN DD *
 INCLUDE SYSLIB(KC901MP,KC902MP,KC903MP,KC904MP)
 NAME KC900MP(R)
//LKED.SYSLIB DD DISP=SHR,DSN=YOUR.LOAD.LIBRARY
//
```

6 Execute the application using the JCL shown below:

```
//CICSRPTS  EXEC PGM=KC900MP,
//             REGION=1M
//STEPLIB   DD DSN=YOUR.LOAD.LIBRARY,DISP=SHR
//SYSUT1    DD DSN=YOUR.SMF.INPUT,DISP=SHR
//SYSUT2    DD DUMMY
//ERRPRINT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//PRODCOST  DD SYSOUT=*
//QUALCOST  DD SYSOUT=*
//TESTCOST  DD SYSOUT=*
//CCTOTPRT  DD SYSOUT=*
//SNAPDUMP  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//
```

CONCLUSION

Once you've implemented this application it is very easy to maintain. The typical things you'll do to keep it current will be to add new transactions or cost centres as your installation adds new applications. Simply follow the steps depicted above to add the changes, then reassemble KC901MP, KC902MP, and KC900MP in that order, and you should be ready to run the application again.

Much can be done to improve the application if you have the time, such as adding more resources or real financial values to the reports, or replacing KC904MP with a much simpler process. Much of this application was written as an early attempt with the Assembler language, so the code is not sophisticated. However, it has proven to be useful over the years and may be especially valuable to a smaller installation.

PROGRAM KC904MP

```
*****************************************************************
*                  PROGRAM KC904MP                              *
*   KC904MP IS AN EXTERNAL SUBROUTINE USED IN THE CICS MONTHLY   *
*   CHARGEBACK REPORTING SYSTEM. IT FORMATS THE GREGORIAN DATE   *
*   AND PASSES IT BACK TO THE CALLING PROGRAM. DATE FORMAT =     *
*   MM/DD/YY -- FOR EXAMPLE 09/10/90.                            *
*****************************************************************
KC904MP  CSECT
         PRINT NOGEN                  MACRO INSTRUCTIONS NOT GENERATED
BEGIN    STM   14,12,12(13)           SAVE CONTENTS OF REG 0-12,13,15
         USING BEGIN,12               USE REG 12 AS PGM BASE REGISTER
```

```
         LR    12,15               LOAD PGM ENTRY POINT ADDR IN REG 12
         LA    11,SAVEB            LOAD ADDR OF PGM SAVEAREA IN REG 11
         ST    13,SAVEB+4              SAVE ADDR OF SYSTEM SAVEAREA
         ST    11,8(13)            SAVE ADDRE OF SAVEA IN SYS SAVEAREA
         LR    13,11                   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
         L     R7,Ø(R1)                COPY ADDR. OF DATE FIELD PASSED
         TIME
         ST    R1,SYSDATE              COPY SYSTEM DATE TO STORAGE
         LA    R3,SYSDATE              COPY SYSTEM DATE TO REG. 3
         MVO   WSYEAR+2(2),1(1,R3)     COPY YEAR FIELD TO STORAGE
         MVC   SAVEYEAR,WSYEAR+2       SAVE YEAR FIELD
         LA    R4,NORMTBL              POINTER TO TABLE OF ACCUM DAYS
         LA    R5,12                   INIT. LIMIT FOR LOOP CONTROL
         UNPK  NORMDAYS,2(2,R3)        UNPACK DAYS FIELD
         OI    NORMDAYS+2,X'FØ'        CHANGE SIGN FOR OUTPUT
FINDDAYS CLC   NORMDAYS,2(R4)          LOOK FOR ACCUM. DAYS IN TABLE
         BNH   CONVERTN                IF A = OR LOW GO TO CONVERTN
HIGH     LA    R4,5(R4)                INCREMENT TABLE ADDRESS
         BCT   R5,FINDDAYS             DECREMENT,TEST AND BRANCH
CONVERTN MVC   GREGMON,Ø(R4)           MOVE MONTH TO GREG. DATE AREA
         S     R4,=F'5'                POINT TO PREVIOUS TBL ELEMENT
         PACK  LODAY,2(3,R4)           PACK ACC. DAYS OF PREVIOUS MO.
         MVC   HIDAY,2(R3)             MOVE JULIAN DAYS TO STORAGE
         SP    HIDAY,LODAY             SUBTRACT TO GET EXACT DAYS
         UNPK  EXACTDAY,HIDAY          UNPACK RESULT OF ABOVE STATE.
         OI    EXACTDAY+2,X'FØ'        CHANGE SIGN FOR OUTPUT
         MVC   GREGDAY,EXACTDAY+1      MOVE EXACT DAYS TO GREG. DATE
CHANGEYR UNPK  GREGYR,SAVEYEAR         UNPACK YEAR FIELD
         OI    GREGYR+2,X'FØ'          CHANGE SIGN FOR OUTPUT
         MVC   GREGYEAR,GREGYR+1       MOVE YEAR TO GREG. DATE AREA
         MVC   Ø(8,R7),GREGDATE        COPY GREG. DATE TO REG. 7
CLOSE    L     13,SAVEB+4              LOAD SAVE AREA
         LM    14,12,12(13)            RESTORE REGISTERS
         SR    15,15                   SET CONDITION CODE TO Ø
         BR    14                      RETURN TO CALLING PROGRAM
```

```
SAVEB    DS    18F
         DC    C'ASSEMBLED BY MORY BINDLER &SYSDATE &SYSTIME'
*************************
*   STORAGE CONSTANTS   *
*************************
SYSDATE  DC    PL4'0'
JULDAYS  DC    PL2'0'
JULYEAR  DC    PL1'0'
SAVEYEAR DC    PL2'0'
WSYEAR   DC    PL4'0'
NORMDAYS DC    CL3' '
GREGDATE DS    0CL8
GREGMON  DC    CL2' '
         DC    C'/'
GREGDAY  DC    CL2' '
         DC    C'/'
GREGYEAR DC    CL2' '
LODAY    DC    PL2'0'
HIDAY    DC    PL2'0'
JULDATE  DC    F'0'
EXACTDAY DC    CL3' '
GREGYR   DC    CL3' '
***********************************
*    TABLE OF ACCUMULATED DAYS    *
***********************************
NORMTBL  DC    CL5'01031'
         DC    CL5'02059'
         DC    CL5'03090'
         DC    CL5'04120'
         DC    CL5'05151'
         DC    CL5'06181'
         DC    CL5'07212'
         DC    CL5'08243'
         DC    CL5'09273'
         DC    CL5'10304'
         DC    CL5'11334'
         DC    CL5'12365'
         END
```

PROGRAM KC901MP

```
*************************************************************************
*                          PROGRAM KC901MP                              *
*  THIS IS AN EXTERNAL SUBROUTINE IN THE MONTHLY CICS CHARGEBACK        *
*  SYSTEM. KC901MP IS CALLED FROM THE MAIN PROGRAM, KC900MP, AND        *
*  PROCESSES THE TRANSACTION DATA RECORD WITHIN THE CICS SMF RECORD     *
*  (SMF RECORD TYPE 110). IF AN ERROR IS ENCOUNTERED, KC901MP CALLS     *
*  EXTERNAL SUBROUTINE KC903MP TO PERFORM ERROR PROCESSING; IF NO       *
*  ERRORS OCCUR, CONTROL IS RETURNED TO KC900MP.                        *
*************************************************************************
```

```
         TITLE 'KC901MP: PROCESS TRANSACTION DATA RECORD'
         PRINT GEN
         MACRO
         COSTCTR &TYPE=ENTRY,&APPLS=,&CC=,&NAME=,&TRANS=
.*=================================================================
.* OPERAND  VALUE      DESCRIPTION
.* ------  --------  ------------------------------------------------
.* TYPE     ENTRY      DEFAULT TYPE IS ENTRY. NEED ONE PER COST CENTRE.
.*                     REQUIRES CENTRE CODE, NAME, & TRANSID OPERANDS.
.*          FINAL      USE ONLY ONCE, AFTER ALL TYPE=ENTRY COMMANDS.
.*                     IT WILL GENERATE THE COST CENTRE ACCUM TABLE AND
.*                     TRANSACTION XREF TABLE. IT ALSO GENERATES A
.*                     DSECT FOR EACH TABLE AND AN EXECUTE COMMAND
.*                     "TRANCOMP" FOR TRANSID LOOKUPS. THE APPLS
.*                     OPERAND IS THE NUMBER OF CICS APPLIDS BEING
.*                     PROCESSED. IT MUST MATCH THE NUMBER OF TABLE
.*                     ENTRIES IN "APPLTABL" OF "KC900MP". THE APPLS
.*                     OPERAND IS REQUIRED TO DEFINE HOW MANY 8-BYTE
.*                     ACCUM FIELDS TO GENERATE PER COST CENTRE.
.*
.* CC       5 DIGITS   NUMERIC COST CENTRE CODE.
.*
.* NAME     1-39 CHAR  COST CENTRE NAME 'ENCLOSED IN APOSTROPHIES'.
.*
.* TRANS    1-4 CHAR   LIST OF ALL TRANIDS IN THIS COST CENTRE.
.*                     IF < 4 CHARS, IT IS USED AS GENERIC PREFIX. TRAN
.*                     LIST IS ENCLOSED IN PARENS & SEPARATED BY COMMAS
.*--------  ---------  ------------------------------------------------
.* EXAMPLE:
.*   COSTCTR CC=10001,NAME='ORDER MANAGEMENT SYSTEM',TRANS=(DB15,NAT)
.*   COSTCTR CC=10002,NAME='FAX NOTIFICATION',TRANS=(FAX)
.*   COSTCTR TYPE=FINAL,APPLS=18
.*=================================================================
         GBLB   &CCERROR       1 = ERROR FOUND... DO NOT GENERATE TABLE
         GBLB   &DONE          1 = FINAL MACRO HAS BEEN PROCESSED
         GBLA   &APPLNUM       NUMBER OF CICS APPLIDS
         GBLA   &CCNUM         COST CENTRE COUNTER
         GBLC   &CCTR(200)     COST CENTRE CODE ARRAY
         GBLC   &CCNM(200)     COST CENTRE DESCRIPTION ARRAY
         LCLA   &C             COST CENTRE ENTRY NUMBER
         GBLA   &TRNUM         TRANS COUNTER
         GBLC   &TRANSID(500)  TRANSID ARRAY
         GBLC   &CCPTR(200)    TRANS COST CENTRE ARRAY POINTER
         LCLA   &T             TRANS ENTRY NUMBER
         LCLA   &W             WORK NUMBER
.*
         AIF   ('&TYPE' EQ 'ENTRY').ENTRY
         AIF   ('&TYPE' EQ 'FINAL').FINAL
         MNOTE 8,'ERROR: INVALID TYPE= OPERAND... MUST BE ENTRY OR FINA+
               L.'
         MEXIT
```

```
.*-------------------------------------------------------------------
.* PROCESS TYPE=ENTRY...
.*-------------------------------------------------------------------
.ENTRY   ANOP
         AIF   (NOT &DONE).GOCC
         MNOTE 8,'ERROR: CANNOT ISSUE TYPE=ENTRY AFTER TYPE=FINAL.'
         MEXIT
.GOCC    ANOP
         AIF   (T'&CC EQ 'N' AND K'&CC EQ 5).CCOK
         MNOTE 8,'ERROR: INVALID COST CENTRE CODE. MUST BE 5 NUMERIC DI+
               GITS.'
&CCERROR SETB  1               INDICATE ERROR
         MEXIT
.CCOK    ANOP
         AIF   (K'&NAME GT 1 AND K'&NAME LT 4Ø).GOTRAN
         MNOTE 8,'ERROR: COST CENTRE NAME MISSING OR INVALID.'
&CCERROR SETB  1               INDICATE ERROR
         MEXIT
.GOTRAN  ANOP
&CCNUM   SETA  &CCNUM+1        COUNT COST CENTRES
&CCTR(&CCNUM) SETC '&CC'       SAVE COST CENTRE CODE
&CCNM(&CCNUM) SETC '&NAME'     SAVE COST CENTRE DESCRIPTION
&T       SETA  1               INIT TRANS ENTRY NUMBER
.SAVTRAN ANOP
         AIF   (NOT K'&TRANS(&T) GT 4).TRANOK
         MNOTE 8,'ERROR: INVALID TRANS ID LENGTH.'
&CCERROR SETB  1               INDICATE ERROR
         AGO   .NXTTRAN
.TRANOK  ANOP
&TRNUM   SETA  &TRNUM+1
&TRANSID(&TRNUM) SETC '&TRANS(&T)' SAVE TRANSID
&CCPTR(&TRNUM) SETC '&CCNUM'   SAVE POINTER TO COST CENTRE
.NXTTRAN ANOP
&T       SETA  &T+1
         AIF   (NOT &T GT N'&TRANS).SAVTRAN
         MEXIT                 TRANS LIST HAS BEEN PROCESSED... EXIT
.*-------------------------------------------------------------------
.* PROCESS TYPE=FINAL...
.*-------------------------------------------------------------------
.FINAL   ANOP
         AIF   (NOT &DONE).OKFINAL
         MNOTE 8,'ERROR: TYPE=FINAL HAS ALREADY BEEN PROCESSED.'
         MEXIT
.OKFINAL ANOP
         AIF   (K'&APPLS GT Ø AND T'&APPLS EQ 'N').OKAPPLS
         MNOTE 8,'ERROR: APPLS (NUMBER OF APPLIDS) OPERAND IS REQUIRED +
               AND MUST BE NUMERIC.'
         MEXIT
.OKAPPLS ANOP
         AIF   (NOT &CCERROR).NOERROR
         MNOTE 8,'ERROR: TYPE=FINAL NOT PROCESSED, DUE TO OUTSTANDING T+
```

```
              YPE=ENTRY ERROR.'
         MEXIT
.NOERROR ANOP
*--------------------------------------------------------------------
* EXECUTE STATEMENT FOR COMPARING TRANSID TO A COST CENTRE PREFIX...
*--------------------------------------------------------------------
         USING CCTRANDS,R11
COMPTRAN CLC   TRANPRE(Ø),TRANSID EXECUTE STMT TO FIND TRANSID IN XREF
         DROP R11
*--------------------------------------------------------------------
* TRANSID / COST CENTRE XREF TABLE...
*--------------------------------------------------------------------
         PRINT GEN
CCTRAN   DS    ØF            TRANSID / COST CENTRE XREF TABLE...
&T       SETA  1             INIT TRANSID ENTRY NUMBER
.GENTRAN ANOP
         AIF   (&T GT &TRNUM).ENDTRAN
&W       SETA  K'&TRANSID(&T)
         DC    A(CC&CCTR(&CCPTR(&T))),A(&W-1),CL4'&TRANSID(&T)'
&T       SETA  &T+1
         AGO   .GENTRAN
.ENDTRAN ANOP
CCTRANE  EQU   12             LENGTH OF A TABLE ENTRY
CCTRAN#  EQU   &TRNUM         NUMBER OF TRANSACTION ENTRIES
*--------------------------------------------------------------------
* COST CENTRE ACCUM TABLE...
*--------------------------------------------------------------------
CCTRTAB  DS    ØF            COST CENTRE ACCUM TABLE...
         DC    A(&CCNUM)     TABLE HEADER: NUMBER OF TABLE ENTRIES
&C       SETA  1             INIT COST CENTRE ENTRY NUMBER
.GENCC   ANOP
         AIF   (&C GT &CCNUM).ENDCC
CC&CCTR(&C)  DC    (&APPLS*2)PL8'Ø',CL11'&CCTR(&C)',CL39&CCNM(&C)
&C       SETA  &C+1
         AGO   .GENCC
.ENDCC   ANOP
*--------------------------------------------------------------------
* TRANSID / COST CENTRE XREF TABLE DSECT...
*--------------------------------------------------------------------
CCTRANDS DSECT               TRANSID / COST CENTRE XREF TABLE...
TRCCPTR  DS    AL4              POINTER TO COST CENTRE ACCUM ENTRY
TRANLEN  DS    F                LENGTH-1 OF TRANSACTION PREFIX
TRANPRE  DS    CL4              TRANSACTION PREFIX
*--------------------------------------------------------------------
* 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
*
&SYSECT  CSECT               RESUME ORIGINAL CSECT
&DONE    SETB  1             INDICATE TYPE=FINAL HAS BEEN PROCESSED
         MEND
*
KC9Ø1MP  CSECT                    DEFINE MODULE AS A CONTROL SECTION
         PRINT NOGEN
BEGIN    STM   R14,R12,12(R13)  SAVE MAIN PROGRAM'S REGISTERS
         USING KC9Ø1MP,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
DATAREC  EQU   *                  BEGIN TO PROCESS DATA RECORD
         L     R3,Ø(R1)           R3 PTS TO ADDR OF 1ST FULLWORD OF
*                                 DATA PASSED FROM KC9ØØMP (SMF DATE)
         L     R5,Ø(R3)           R5 CONTAINS SMF DATE
         STCM  R5,B'1ØØØ',SWITCH SAVE PARM PASSED FROM KC9ØØMP WHEN
*                                 ALL SMF RECS HAVE BEEN PROCESSED
         TM    SWITCH,X'8Ø'       HAVE ALL TRAN DATA RECS BEEN DONE?
         BC    1,DATADONE         YES: GO CALL KC9Ø2MP
         MVC   SMFDATE(4),Ø(R3)   NO: SAVE SMF DATE
         L     R4,4(R1)           R4 PTS TO ADDR OF 2ND FULLWORD OF
*                                 DATA PASSED FROM KC9ØØMP (APPLID)
         MVC   APPLID(8),Ø(R4)    SAVE VTAM APPLID
         L     R7,8(R1)           R7 PTS TO ADDR OF 3RD FULLWORD OF
*                                 DATA PASSED FROM KC9ØØMP (APPL OFFSET)
         MVC   APPLOFF(2),Ø(R7)   SAVE APPLID OFFSET IN COST CTR TABLE
         L     R2,12(R1)          R2 PTS TO ADDR OF 4TH FULLWORD OF
*                                 DATA PASSED FROM KC9ØØMP (TRANSID)
         MVC   TRANSID,Ø(R2)      SAVE TRANSACTION NAME
```

```
         EJECT
*
TRANCHEK EQU   *                 ROUTINE TO CHECK TRANSACTION ID
*
         ST    R11,SAVEREGB      SAVE REG 11 CONTENTS
         USING CCTRANDS,R11      R11 IS BASE FOR TRAN XREF TBL
         LA    R11,CCTRAN        -> COST CENTRE/TRANSID XREF TABLE
         LA    R4,CCTRAN#        SET LOOP COUNT = NUMBER OF TRANSID'S
CCFIND   EQU   *
         L     R15,TRANLEN       GET LENGTH OF TRAN PREFIX FOR COMPARE
         EX    R15,COMPTRAN      DOES TRANSID MATCH THE PREFIX?
         BE    GOTCC              YES - PROCESS
         LA    R11,CCTRANE(,R11)   NO - TRY NEXT
         BCT   R4,CCFIND
*
TRANERR  EQU   *                 ERROR: A TRANSID MATCH DID NOT OCCUR
         OI    ERRFLAG,X'4Ø'     INDICATE TRANSACTION NAME ERROR
         MVC   ERRDATE(4),SMFDATE  SAVE DATE WHEN SMF BLOCK WRITTEN
CALLERR  EQU   *                 SET UP CODE TO CALL ERROR SUBRTN
         LA    R1,=A(ERRFLAG,ERRDATE,APPLID,TRANSID)  LOAD PARMLIST
         L     R15,=V(KC9Ø3MP)   LOAD ADDRESS OF ERROR SUBROUTINE
         BALR  R14,R15           INVOKE ERROR SUBROUTINE & RETURN
         B     EXITPGM           BRANCH TO EXIT
GOTCC    EQU   *
         L     R11,TRCCPTR       -> COST CENTRE ACCUM TABLE ENTRY
         AH    R11,APPLOFF       INCREMENT REG 11 WITH APPLID OFFSET
         DROP  R11               STOP USING R11 BASED ON TRAN XREF TBL
         USING CCACCUM,R11       R11 IS BASE FOR CC ACCUM TBL
         AP    CCACCUM,=PL1'1'   ADD 1 TO TRANSACTION COUNT
         L     R9,412(R2)        COPY CPU TIME TO REG 9   (CHG 9/16/99)
*        DC    HL2'Ø'            FORCE A DUMP
         CVD   R9,CPUTMEDW       CONVERT CPU TIME TO PACKED DECIMAL
         AP    CCACCUM+8,CPUTMEDW ADD CPU TIME TO CUMULATIVE CPU TIME
         L     R4,SAVEREG4       RESTORE REG 4 CONTENTS
         L     R9,SAVEREG9       RESTORE REG 9 CONTENTS
         L     R1Ø,SAVEREGA      RESTORE REG 1Ø CONTENTS
         L     R11,SAVEREGB      RESTORE REG 11 CONTENTS
         DROP  R11               STOP USING R11 BASED ON CC ACCUM TBL
EXITPGM  EQU   *                 PREPARE TO RETURN TO CALLING PGM
         L     R13,SAVEA+4       LOAD CALLING PGM'S REGS
         LM    R14,R12,12(R13)   RESTORE CALLING PROGRAM'S REGS
         SR    R15,R15           ESTABLISH RETURN CODE Ø
         BR    R14               RETURN TO CALLING PROGRAM (KC9ØØMP)
DATADONE EQU   *                 PREPARE TO CALL KC9Ø2MP
         L     R3,4(R1)          GET POINTER TO PROCESSING PERIOD DATES
         MVC   PERIOD,Ø(R3)      PASS PERIOD DATES
         LA    R1,=A(CCTRTAB,PERIOD)  CCTRTABE ADDR TO BE PASSED
         L     R15,=V(KC9Ø2MP)   LOAD ADDRESS OF KC9Ø2MP
         BALR  R14,R15           INVOKE KC9Ø2MP & RETURN
         B     EXITPGM           WE'RE DONE
         LTORG
```

```
         DC    C'ASSEMBLED BY MORY BINDLER &SYSDATE &SYSTIME'
*
         SPACE 2
APPLID   DC    CL8' '            VTAM APPLID IN SMF REC (SMFPSPRN)
APPLID1  DC    CL8' '            VTAM APPLID FOUND IN APPLID TABLE
SMFDATE  DC    PL4'0'            DATE WHEN SMF BLOCK WAS WRITTEN
APPLOFF  DS    H                 OFFSET OF APPLID IN COST CENTRE TABLE
TRANSID  DS    CL4               TRANSACTION NAME
CURRCCTR DS    CL8               CURRENT COST CTR # IN COST CTR TBL
PERIOD   DS    CL16              PROCESSING DATE PERIOD (FROM-TO DATES)
SAVEREG1 DC    F'0'              SAVE CONTENTS OF REG. 1
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
LASTCCTR DC    C'99999   '       DUMMY ENTRY AT END OF TRANSACTION TBL
CPUTMEDW DC    D'0'              CPU TIME IN PACKED DEC. (FOR 1 REC)
CPUDVSOR DC    PL3'62500'        DIVISOR USED TO OBTAIN CPU TIME (SECS)
CPUTIME1 DC    PL8'0'            TOTAL CPU TIME IN SECONDS
CPUTIME2 DC    PL8'0'            CPU TIME INTERMEDIATE STORAGE
CPUTIME3 DC    PL8'0'            GRAND TOTAL CPU TIME IN SECONDS
C3TRNCUM DC    PL8'0'            GRAND TOTAL TRANS COUNT FOR 1 CCTR
C4TRNCUM DC    PL8'0'            GRAND TOTAL TRANS COUNT FOR 1 CCTR
TBPOINTR DC    F'0'              ADDR OF CURRENT TRANSID TBL ELEMENT
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    PL2'0'            ERROR PAGE COUNT
SAVEA    DS    18F               SAVE AREA
ERRMSG1  DC    C'THIS APPLID DOES NOT EXIST IN THE APPLID TABLE'
ERRMSG2  DC    C'THIS TRANSACTION ID NOT HANDLED IN THE PROGRAM'
ERRMSG3  DC    C'THIS COST CENTRE DOES NOT EXIST IN ADDRESS TBL'
ERRMSG4  DC    C'A WEIRD PROB W/ THE ERROR RTN: DISPLAY ERRFLAG'
         SPACE 2
         PRINT GEN
**********************************
*    COST CENTRE ADDRESS TABLE    *
**********************************************
*    COST CENTRE/APPLID STATISTICS TABLE     *
**********************************************
*  THIS TABLE IS ARRANGED IN PRINTED REPORT SEQUENCE, WHICH IS ALSO
*  IN COLLATING SEQUENCE.
```

```
*  NOTE! THERE ARE 2 PL8 BUCKETS FOR EACH APPLID IN THE CALLING PROG
*        KC900MP'S TABLE "APPLTABL".
*        THE NUMBER OF APPLID'S IN "APPLTABL" MUST MATCH THE "APPLS="
*        OPERAND OF THE "COSTCTR TYPE=FINAL" MACRO BELOW!
        COSTCTR CC=10010,NAME='ABENDAID/FX',                           +
                TRANS=(AADB,AADF,AADM,AAON,AARB,ERWV)
        COSTCTR CC=10015,NAME='AMP DISTRIBUTION REFERRAL',             +
                TRANS=(RSTK,R104)
        COSTCTR CC=10020,NAME='ASI',                                   +
                TRANS=(ASAP,DDA,DR,EP,FB,IC,IM,SAR)
        COSTCTR CC=10030,NAME='CARMS',                                 +
                TRANS=(AR,CARM)
        COSTCTR CC=10040,NAME='CBT TRAINING',                          +
                TRANS=(DB06)
        COSTCTR CC=10050,NAME='CICS/CEMT',                             +
                TRANS=(BM,MTP)
        COSTCTR CC=10060,NAME='CONSTRUCT',                             +
                TRANS=(DB09)
        COSTCTR CC=10070,NAME='DBA',                                   +
                TRANS=(DB04,DB05,DB08,DB09,DB13,DB14,DB21,DB22,IMD1)
        COSTCTR CC=10080,NAME='EDI',                                   +
                TRANS=(ED)
        COSTCTR CC=10090,NAME='FAX NOTIFY',                            +
                TRANS=(FAX)
        COSTCTR CC=10100,NAME='GENERAL LEDGER',                        +
                TRANS=(FB,FIP,GK,GL,GS)
        COSTCTR CC=10110,NAME='MAINVIEW',                              +
                TRANS=(FCD2,FIC2,JNL2)
        COSTCTR CC=10120,NAME='MENU SYSTEM',                           +
                TRANS=(CIGN,CSGM,MEN,SIGN,SOFF)
        COSTCTR CC=10130,NAME='NATURAL',                               +
                TRANS=(NATD,NAT0)
        COSTCTR CC=10140,NAME='NATURAL DEVELOPMENT',                   +
                TRANS=(DB02,NATX,NATY)
        COSTCTR CC=10150,NAME='NATURAL SECURITY',                      +
                TRANS=(DB10)
        COSTCTR CC=10160,NAME='OPEN OBLIGATIONS',                      +
                TRANS=(OO)
        COSTCTR CC=10170,NAME='ORDER MANAGEMENT',                      +
                TRANS=(DB15,DB17,NATG,NATH,OCSS,OE,WS)
        COSTCTR CC=10180,NAME='PREDICT',                               +
                TRANS=(DB03)
        COSTCTR CC=10190,NAME='SYSTEM ARCHIVE AND RETRIEVAL',          +
                TRANS=(SAR)
        COSTCTR CC=10200,NAME='SUPER NATURAL',                         +
                TRANS=(DB07,NAT1)
        COSTCTR CC=10210,NAME='TECH SUPPORT',                          +
                TRANS=(STAT,TSS,Z,MTP,NAMD,NM,CD,HP,EB,XM)
        COSTCTR CC=10990,NAME='WHO KNOWS?',                            +
                TRANS=(ACCT,CRMD,CRMM,PC)
*
```

```
        COSTCTR TYPE=FINAL,APPLS=18
          END
```

PROGRAM KC902MP

```
*************************************************************************
*                     PROGRAM KC9Ø2MP                                   *
*  KC9Ø2MP IS AN EXTERNAL SUBROUTINE IN THE CICS MONTHLY CHARGEBACK     *
*  SYSTEM. IT IS CALLED BY KC9Ø1MP AFTER THE COST CENTRE/CICS APPLID    *
*  TABLE HAS BEEN BUILT. THE MAIN FUNCTION OF KC9Ø2MP IS TO FORMAT      *
*  AND PRINT THE CHARGEBACK REPORTS. AT COMPLETION, IT RETURNS          *
*  CONTROL BACK TO KC9Ø1MP.                                             *
*************************************************************************
         TITLE 'KC9Ø2MP: FORMAT AND PRINT CHARGEBACK REPORTS'
KC9Ø2MP  CSECT                    DEFINE MODULE AS A CONTROL SECTION
         PRINT NOGEN
BEGIN    STM   R14,R12,12(R13)  SAVE CALLING PROGRAM'S REGISTERS
         USING BEGIN,R12,R8     USE REGS 12 & 8 AS BASE REGISTERS
         LR    R12,R15          LOAD PGM ENTRY POINT ADDR IN BASE REG
BASEREG1 L     R8,BASEREG2      EST. ADDRESSABILITY TO 2ND BASE REG
         B     NEXT             BRANCH AROUND STORAGE CONSTANT
BASEREG2 DC    A(BEGIN+4Ø96)    EST. ADDRESS OF 2ND BASE REG
NEXT     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
         ST    R1,SAVEREG1       SAVE ADDRESS OF PARMLIST PASSED TO US
         OPEN  (PRODCOST,(OUTPUT),QUALCOST,(OUTPUT),TESTCOST,(OUTPUT))
         OPEN  (CCTOTPRT,(OUTPUT))
         LA    R1,=A(CURRDATE)   LOAD ADDR OF PARM TO BE PASSED
         L     R15,=V(KC9Ø4MP)   LOAD ENTRY POINT OF KC9Ø4MP
         BALR  R14,R15           BRANCH TO KC9Ø4MP AND RETURN
         MVC   SYSDATE1,CURRDATE MOVE DATE TO TITLE LINES...
         MVC   SYSDATE2,CURRDATE
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
```

```
         EJECT
*********************************************************************
*   FORMAT AND PRINT 3 COST ALLOCATION REPORTS: PROD, QUAL, TEST    *
*   IS A REPORT OF TOTAL TRANSACTION COUNTS AND CPU TIME BY         *
*   COST CENTRE AND CICS REGION; IT INCLUDES PRODUCTION REGIONS.    *
*********************************************************************
COSTREPT EQU   *                 FORMAT AND PRINT CHARGEBACK REPORTS
         L     R1,SAVEREG1       RESTORE REG 1
         L     R3,4(R1)          -> PROCESSING PERIOD DATES
         MVC   FROM1,Ø(R3)       MOVE
         MVC   THRU1,8(R3)        DATES
         MVC   FROM2,Ø(R3)          TO
         MVC   THRU2,8(R3)           TITLE LINES
         L     R3,Ø(R1)          LOAD CCTRTAB ADDR PASSED FROM KC9Ø1
         MVC   CC#,Ø(R3)         SAVE NUMBER OF COST CENTRES
         USING CCTABDS,R3
         LA    R3,CCACCUM        R3 -> 1ST COST CENTRE ACCUM ENTRY
         USING CCACCUM,R3
         ST    R3,CCPTR          SAVE ADDRESS OF COST CENTRE TABLE
         LA    R5,APPLTABL       R5 -> APPLID/ACCUM DISPLACEMENT TABLE
         USING APPLIDDS,R5
         MVC   APPLTYPE,=CL1Ø'PRODUCTION'
         L     R1Ø,=A(PRODCOST)  R1Ø -> PROD REPORT DCB
         BAL   R4,CAREPORT       PRINT COST ALLOC REPORT FOR PROD
         MVC   APPLTYPE,=CL1Ø'   QUALITY'
         L     R1Ø,=A(QUALCOST)  R1Ø -> QUAL REPORT DCB
         BAL   R4,CAREPORT       PRINT COST ALLOC REPORT FOR QUAL
         MVC   APPLTYPE,=CL1Ø'      TEST'
         L     R1Ø,=A(TESTCOST)  R1Ø -> TEST REPORT DCB
         BAL   R4,CAREPORT       PRINT COST ALLOC REPORT FOR TEST
         EJECT
*********************************************************************
*   FORMAT AND PRINT THE COST CENTRE GRAND TOTALS REPORT: THIS      *
*   IS A REPORT OF TOTAL TRANSACTION COUNTS AND CPU TIME FOR        *
*   EACH COST CENTRE.                                               *
*********************************************************************
PUT   CCTOTPRT,CTTITLE  PRINT COST CENTRE TOTALS: TITLE
         PUT   CCTOTPRT,CTHEAD1                          : HEADING 1
         PUT   CCTOTPRT,CTHEAD2                          : HEADING 2
         L     R3,CCPTR          R3 -> 1ST COST CENTRE TABLE ENTRY
         L     R9,CC#            R9 = NUMBER OF COST CENTRES
         ZAP   COUNT3,=P'Ø'      CLEAR UNDERLINE COUNTER
CCTOTALS EQU   *                 FORMAT & PRT COST CTR GRAND TOTALS
         ZAP   CTTRNCUM,=P'Ø'    CLEAR
         ZAP   CTCPUCUM,=P'Ø'      LINE TOTALS
         MVC   CTCSTCTR(5Ø),CCTR    MOVE CCTR # & APPL INFO TO PLINE
         LA    R6,CCACCUM        R6 -> 1ST ACCUM SET FOR THIS COST CTR
         LA    R7,APPLS          R7 = NUMBER OF ACCUM SETS
CTLOOP   EQU   *
         AP    CTTRNCUM,Ø(8,R6)  ADD TRAN CNT FROM TBL TO COUNTER
         ZAP   CPUTIMEC,8(8,R6)  GET CPU TIME FROM TBL TO COUNTER
```

```
         MP    CPUTIMEC,=PL2'16' CALC CPUTIME IN MICROSECONDS
         DP    CPUTIMEC,=PL4'100'  CALC CPUTIME IN SECONDS
         AP    CTCPUCUM,CPUTIMEC(7) ACCUM TOTAL CPUTIME IN SECONDS
         LA    R6,16(,R6)        -> NEXT APPLID'S ACCUM SET
         BCT   R7,CTLOOP
         AP    CTTRNGT#,CTTRNCUM ACCUM TRANCOUNT GRAND TOTAL
         MVC   CTTRANS#(12),TRANMASK    MOVE EDIT MASK TO PRINT LINE
         ED    CTTRANS#(12),CTTRNCUM+3  EDIT GRAND TOTAL TRANS COUNT
         AP    CTCPUGT#,CTCPUCUM ACCUM CPUTIME GRAND TOTAL
         MVC   CTCPUTM#(15),CPUMASK1     MOVE EDIT MASK TO PRINT LINE
         ED    CTCPUTM#(15),CTCPUCUM+2   EDIT TOTAL FOR THIS COST CTR
         AP    COUNT3,=P'1'
         CP    COUNT3,=P'3'
         BL    NOT3RDT
         TR    CTDETAIL,UNDERLIN UNDERLINE EVERY 3RD LINE
         ZAP   COUNT3,=P'0'      CLEAR UNDERLINE COUNTER
NOT3RDT  EQU   *
         PUT   CCTOTPRT,CTDETAIL PRINT A DETAIL LINE
         MVC   CTDETAIL,CTDETAIL-1
         LA    R3,CCTRTABE(,R3)  -> NEXT COST CENTRE ENTRY IN CCTRTABL
         BCT   R9,CCTOTALS
***************************** READY TO PRINT GRAND TOTAL LINES
         MVC   CTTRNTOT(12),TRANMASK  MOVE EDIT MASK TO PRINT LINE
         ED    CTTRNTOT(12),CTTRNGT#+3  FORMAT GRAND TOTAL TRN CNT
         MVC   CTCPUTOT(18),CPUMASK2  MOVE EDIT MASK TO PRINT LINE
         ED    CTCPUTOT(18),CTCPUGT#+1  FORMAT GRAND TOTAL CPUTIME
         PUT   CCTOTPRT,CTTRNPRT  PRINT TRANS GRAND TOTALS LINE
         PUT   CCTOTPRT,CTCPUPRT  PRINT CPU TIME GRAND TOTALS LINE
         EJECT
*********************************************************************
* REPORTS ARE DONE... EXIT
*********************************************************************
EXITPGM  EQU   *
         L     R13,SAVEA+4      RESTORE ADDR OF SYSTEM SAVEAREA
         LM    R14,R12,12(R13)  RESTORE CALLER'S REGISTERS
         SR    R15,R15          SET GOOD RC
         BR    R14              RETURN TO CALLER
         EJECT
*********************************************************************
* THE CAREPORT SUBROUTINE WILL PRINT A COST ALLOC REPORT.*     UPON
ENTRY, THE FOLLOWING REGISTERS MUST BE SET:
*     R4   -> RETURN FROM THIS SUBROUTINE
*     R5   -> 1ST APPLTABL ENTRY FOR THIS REPORT
*     R10  -> PRINTER DCB FOR THIS REPORT
*     UPON EXIT, THE FOLLOWING REGISTERS WILL BE SET:
*     R5   -> 1ST APPLTABL ENTRY FOR NEXT REPORT
*********************************************************************
CAREPORT EQU   *              PRINT A COST ALLOC REPORT...
         PUT   (10),CATITLE       PRINT TITLE
         LA    R6,CAHD1A          R6 -> 1ST APPLID NAME IN HEADER LINE 1
         LA    R2,CATOTALS        R2 -> TOTAL COUNTERS
```

```
         LA    R11,CACOLS        R11 = NUMBER OF COLUMNS PER PAGE
         SLL   R11,1             MULT X 2 COUNTERS PER COLUMN
CLRCATOT EQU   *
         ZAP   Ø(8,R2),=P'Ø'     ZERO TOTAL COUNTERS...
         LA    R2,8(,R2)
         BCT   R11,CLRCATOT
         LA    R11,CACOLS        R11 = NUMBER OF COLUMNS PER PAGE
         L     R3,CCPTR          R3 -> 1ST COST CENTRE TABLE ENTRY
         L     R9,CC#            R9 = NUMBER OF COST CENTRES
         ST    R5,SAVEREG5       SAVE POINTER TO 1ST APPLID ENTRY
HEADLOOP EQU   *              PRINT COLUMN HEADINGS...
         MVC   Ø(8,R6),APPLNAME  MOVE APPLID TO HEADER
         LA    R6,CW(,R6)        -> NEXT COLUMN
         LA    R5,APPLTABE(,R5)  -> NEXT APPLTABL ENTRY
         C     R5,=A(APPLEND)    PAST END OF TABLE ?
         BNL   HEADEND            YES - END LOOP
         BCT   R11,HEADLOOP       NO - CONTINUE
HEADEND  EQU   *
         PUT   (1Ø),CAHEAD1      PRINT HEADER 1
         PUT   (1Ø),CAHEAD2      PRINT HEADER 2
         MVI   CADETCTL,C'Ø'     SET 1ST DETAIL LINE FOR DOUBLE SPACING
         ZAP   COUNT3,=P'Ø'      CLEAR UNDERLINE COUNTER
CCLOOP   EQU   *              PRINT DETAIL LINES...
         MVC   CADETCC,CCCODE    MOVE COST CENTRE CODE TO DETAIL LINE
         MVC   CADET1,CADET1-1   CLEAR DETAIL COLUMNS
         L     R5,SAVEREG5       RESTORE POINTER TO 1ST APPLID ENTRY
         LA    R6,CADET1         R6 -> 1ST COLUMN IN DETAIL LINE
         LA    R2,CATOTALS       R2 -> TOTAL COUNTERS... AGAIN
         LA    R11,CACOLS        R11 = NUMBER OF COLUMNS PER PAGE
COLLOOP  EQU   *                 BUILD COLUMN DATA...
         LR    R7,R3             R7 -> CURRENT COST CENTRE ENTRY
         AH    R7,APPLOFF        ADD OFFSET TO APPLID'S ACCUM COUNTS
         MVC   Ø(CW,R6),DETMASKS MOVE EDIT MASKS FOR THIS COLUMN
         AP    Ø(8,R2),Ø(8,R7)   ACCUM APPLID TOTAL
         OC    Ø(4,R7),Ø(R7)     ANYTHING IN HI-ORDER OF TRANCOUNT?
         BZ    EDITTRAN            NO - OK
         MVC   1(9,R6),=9C'*'     YES - SHOW OVERFLOW WITH ASTERISKS
         B     OVERTRAN
EDITTRAN ED    Ø(1Ø,R6),4(R7)    EDIT TRAN COUNT FROM LAST 4 BYTES
OVERTRAN EQU   *
         ZAP   CPUTIMEB,8(8,R7)  CONVERT CPU TIME
         MP    CPUTIMEB,=PL2'16'   CONVERT TO MICROSECONDS JJK
         DP    CPUTIMEB,=PL4'1ØØ' CONVERT TO MILLISECONDS    JJK
         AP    8(8,R2),CPUTIMEB(7)  ACCUM APPLID TOTAL
         OC    CPUTIMEB(2),CPUTIMEB  ANYTHING IN 1ST 4 BYTES?
         BZ    EDITCPU             NO - OK
         MVC   11(9,R6),=9C'*'    YES - SHOW OVERFLOW WITH ASTERISKS
         B     OVERCPU
EDITCPU  ED    1Ø(1Ø,R6),CPUTIMEB+2  EDIT CPU VALUE FROM 2ND 4 BYTES
OVERCPU  EQU   *
         LA    R6,CW(,R6)        -> NEXT COLUMN
```

```
         LA    R5,APPLTABE(,R5)  -> NEXT APPLTABL ENTRY
         C     R5,=A(APPLEND)    PAST END OF TABLE ?
         BNL   PRTDET             YES - PRINT DETAIL AS IS
         LA    R2,16(,R2)        -> NEXT SET OF TOTAL COUNTERS
         BCT   R11,COLLOOP
PRTDET   EQU   *
         AP    COUNT3,=P'1'
         CP    COUNT3,=P'3'
         BL    NOT3RD
         TR    CADET1,UNDERLIN   UNDERLINE EVERY 3RD LINE
         ZAP   COUNT3,=P'Ø'      CLEAR UNDERLINE COUNTER
NOT3RD   EQU   *
         PUT   (1Ø),CADETAIL     PRINT DETAIL LINE
         MVI   CADETCTL,C' '     SET DETAIL LINES FOR SINGLE SPACING
         LA    R3,CCTRTABE(,R3)  -> NEXT COST CENTRE ACCUM ENTRY
         BCT   R9,CCLOOP
***************************** PRINT APPLID TOTAL LINES...
         PUT   (1Ø),CADASHES     PRINT TOTALS DIVIDER LINE
         MVC   CATOT1,CATOT1-1   CLEAR
         MVC   CATOTAL2,CATOTAL2-1   TOTAL LINES
         MVI   CATOTAL2,C'+'     SET TOTAL LINE 2 TO OVERPRINT LINE 1
         LA    R2,CATOTALS       R2 -> TOTAL COUNTERS
         LA    R11,CACOLS        R11 =NUMBER OF APPLID COLUMNS
         LA    R6,CATOTAL1+6     R6 -> 1ST TRANCOUNT TOTAL
         LA    R7,CATOTAL2+16    R7 -> 1ST CPUTIME TOTAL
TOTLOOP  EQU   *
         MVC   Ø(12,R6),TOTMASK
         ED    Ø(12,R6),3(R2)    EDIT TRANCOUNT TOTAL TO LINE
         CLC   Ø(3,R6),=CL3' '   OVERFLOW COLUMN WIDTH ?
         BE    OKTRTOT             NO - OK
         MVI   CATOTAL2,C' '      YES - SET TOTAL LINE 2 NOT OVERPRINT
OKTRTOT  EQU   *
         MVC   Ø(12,R7),TOTMASKC
         ED    Ø(12,R7),1Ø(R2)   EDIT TRANCPU   TOTAL TO LINE
         CLC   Ø(3,R7),=CL3' '   OVERFLOW COLUMN WIDTH ?
         BE    OKCPTOT             NO - OK
         MVI   CATOTAL2,C' '      YES - SET TOTAL LINE 2 NOT OVERPRINT
OKCPTOT  EQU   *
         LA    R2,16(,R2)        -> NEXT TOTAL COUNTER SET
         LA    R6,CW(,R6)        -> NEXT TRANCOUNT TOTAL IN PRINT LINE
         LA    R7,CW(,R7)        -> NEXT CPUTIME TOTAL IN PRINT LINE
         BCT   R11,TOTLOOP
         PUT   (1Ø),CATOTAL1     PRINT TOTAL LINE 1
         PUT   (1Ø),CATOTAL2     PRINT TOTAL LINE 2
* ****** END OF COST ALLOC REPORT, PRINT SUBROUTINE...
         BR    R4                RETURN
         LTORG
         DC    C'ASSEMBLED BY MORY BINDLER &SYSDATE &SYSTIME'
APPLID   DS    CL8               VTAM APPLID IN SMF REC (SMFPSPRN)
APPLID1  DS    CL8               VTAM APPLID FOUND IN APPLID TABLE
TRANSID  DS    CL4               TRANSACTION NAME
```

```
CURRCCTR DS    CL8               CURRENT COST CTR # IN COST CTR TBL
TRANMASK DC    X'402020206B2020206B202120'
CPUMASK  DC    X'40206B2020206B202120'
DETMASKS DC    X'40206B2020206B202120',X'40206B2020214B202020'
TOTMASK  DC    X'402020206B2020206B202120'
TOTMASKC DC    X'402020206B2020214B202020'
CPUMASK1 DC    X'4020206B2020206B2020214B202020'  FOR DETAIL LINE
CPUMASK2 DC    X'40206B2020206B2020206B2020214B202020' FOR GR TOT LINE
CCPTR    DS    F                 ADDRESS OF 1ST COST CENTRE TABLE ENTRY
CC#      DS    F                 NUMBER OF COST CENTRES
CCTRPNTR DC    F'0'              CURR POSITION IN CCTRTABL (COST CTR)
APPLPNTR DC    F'0'              CURR POSITION IN CCTRTABL (APPLID)
SAVEREG1 DC    F'0'              SAVE CONTENTS OF REG. 1
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
LASTCCTR DC    C'99999   '       DUMMY ENTRY AT END OF TRANSACTION TBL
CURRDATE DC    CL8' '            CURRENT DATE FOR REPORTS (FROM KC904)
CPUTMEDW DC    D'0'              CPU TIME IN PACKED DEC. (FOR 1 REC)
CPUDVSOR DC    PL3'62500'        DIVISOR USED TO OBTAIN CPU TIME (SECS)
CPUTIMEB DC    PL11'0'           TOTAL CPU TIME IN BINARY
CPUTIMEC DC    PL11'0'           TOTAL CPU TIME IN SECONDS (1 COST CTR)
CPUTIMED DC    PL11'0'           TOTAL CPU TIME IN SECONDS (1 COST CTR)
PACKDEC0 DC    PL11'0'           FIELD USED TO CLEAR CPUTIMEB
CPUTIMES DC    PL8'0'            TOTAL CPU TIME IN SECONDS
CPUTIME1 DC    PL8'0'            TOTAL CPU TIME IN SECONDS
CPUTIME2 DC    PL8'0'            CPU TIME INTERMEDIATE STORAGE
CPUTIME3 DC    PL8'0'            GRAND TOT CPU TIME IN SECS (1 APPLID)
CTTRNCUM DC    PL8'0'            GRAND TOTAL TRANS COUNT FOR 1 CCTR
CTTRNGT# DC    PL8'0'            GRAND TOTAL TRN COUNT FOR ALL CCTR'S
CTCPUCUM DC    PL8'0'            GRAND TOTAL CPU TIME FOR 1 COST CTR
CTCPUGT# DC    PL8'0'            GRAND TOTAL CPU TIME FOR ALL CCTR'S
C4TRNCUM DC    PL8'0'            GRAND TOTAL TRANS COUNT FOR 1 CCTR
C4TRNGT# DC    PL8'0'            GRAND TOT TRANS CNT FOR ALL CCTR'S
C4CPUCUM DC    PL8'0'            GRAND TOTAL CPU TIME FOR 1 COST CTR
C4CPUGT# DC    PL8'0'            GRAND TOT CPU TIME IN SECS (ALL APPS)
TBPOINTR DC    F'0'              ADDR OF CURRENT TRANSID TBL ELEMENT
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
COUNT3   DC    PL1'Ø'            UNDERLINE COUNTER
UNDERLIN DC    256AL1(*-UNDERLIN)
         ORG   UNDERLIN+C' '
         DC    C'.'              TRANSLATE BLANKS TO UNDERLINES
         ORG   ,
SAVEA    DS    18F               REGISTER SAVE AREA
WORK8    DS    PL8               PACKED DECIMAL WORK AREA
         SPACE 2
*                  **************
*                  *   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
```

Editor's note: the rest of the code will be published next month.

Mory Bindler
Consultant
3D Business Solutions (USA)

SYSIN checker

Because of the volume of CICS regions we support, we recently converted all of our systems to use a base or default SIT, with a SYSIN override containing region-specific parameters.

After the successful conversion, the question was posed as to how we could check whether the SYSIN was valid before it was used.

The following REXX has been written to address this problem, and has been coded so as to minimize the user input required.

The REXX and skeleton JCL members must be placed in a library that is available to your TSO profile. When executing, all modifies and alterations are done within the SYSIN member you are editing, but no permanent changes will be made, because the SYSIN member is CANCELLED out of on exit. The REXX triggers a batch job to assemble the source with a NOTIFY in the JCL deck, which returns an RC=0 to the user if all is OK.

To run the REXX, edit the SYSIN PDS member and type **CHKSIT**.

This performs the following steps in background:

1 Removes all comment lines.

2 Checks for LPA parameters and removes them if necessary.

3 Checks for .END end of field marker (if it is missing it aborts the process).

4 Remove parameter comments if present.

5 If GMTEXT > 57 then it splits to the next line.

6 Repositions data to SIT table format.

7 Copies data to member ZZTEST (SYSIN in SIT table format).

8 Dynamically builds REXX to allow SYSIN to be added to the default SIT.

9 Saves REXX/macro to member CHKSIT*xx*.

10 Sets default SIT member value.

11 Clears down SYSIN member and copies in default SIT source.

12 Execute REXX/macro CHKSIT*xx* to insert SYSIN in the correct format.

13 Copies to member ZZZTEST (default SIT plus SYSIN in SIT table format).

14 CANCELs out of SYSIN edit member.

15 Submits a batch job to assemble:

- ZZTEST (SYSIN plus IBM default SIT settings)
- ZZZTEST (default SIT plus SYSIN in SIT table format).

CHKSIT REXX SOURCE

```
/* REXX */
/* CHECKS SYNTAX OF SYSIN PARMS  */
ADDRESS "ISREDIT"
"MACRO"
"ISREDIT ("MEM") = MEMBER"
/* REMOVE COMMENTS */
"ISREDIT EXCLUDE '*' 1 2 ALL"
"ISREDIT DELETE EXCLUDE ALL"
/* CHECK FOR LPA PARM */
"ISREDIT FIND 'PRVMOD=(' LAST"
IF RC = Ø THEN
   DO
   "ISREDIT ("LNA","CNA") = CURSOR"
   "ISREDIT FIND '),' NEXT"
   "ISREDIT ("LNB","CNB") = CURSOR"
   "ISREDIT DELETE "LNA" "LNB" "
END
DROP LNA CNA LNB CNB
/* CHECK FOR EOF MARKER */
"ISREDIT FIND '.END' LAST"
IF RC = Ø THEN
    DO
    "ISREDIT ("LNC","CNC") = CURSOR"
    #LASTLINE = LNC - 1
END
ELSE
   DO
   SAY 'ERROR : .END MARKER MISSING AT THE BOTTOM OF THE SYSIN'
```

```
   MEND
   EXIT
END
/* Remove Parm Comments */
#POINT = 1
DO #POINT = 1 TO #LASTLINE
   "ISREDIT ("#LINE") = LINE "#POINT
   #ENDPOS = POS(', ',#LINE,1)
   #CHECK = LEFT(#LINE,#ENDPOS - 1)
   #REST = RIGHT(#LINE,8Ø - #ENDPOS)
   IF SUBSTR(#CHECK,1,7) ¬= 'GMTEXT=' THEN
      DO
      "ISREDIT CHANGE '"#REST"' '' ALL"
   END
   DROP #LINE
END
DROP LNC CNC
/* CHECK FOR EOF MARKER */
"ISREDIT FIND '.END' FIRST"
IF RC = Ø THEN
   DO
   "ISREDIT ("LNC","CNC") = CURSOR"
   #LASTLINE = LNC - 1
END
/* MAIN LOOP */
#POINT = 1
DO #POINT = 1 TO #LASTLINE
   "ISREDIT ("#LINE") = LINE "#POINT
   #ENDPOS = POS(' ',#LINE)
   #CHECK = LEFT(#LINE,#ENDPOS - 1)
   IF SUBSTR(#CHECK,1,7) = 'GMTEXT=' | #ENDPOS > 57 THEN
      DO
      "ISREDIT TSPLIT "#POINT" 57"
   END
   DROP #LINE
END
"ISREDIT RESET"
"ISREDIT CHANGE ' ' 'X' 57 ALL"
DROP LNC CNC
/* CHECK FOR EOF MARKER */
"ISREDIT FIND '.END' FIRST"
IF RC = Ø THEN
   DO
   "ISREDIT ("LNC","CNC") = CURSOR"
   #LASTLINE = LNC - 1
END
/* REPOSITION DATA */
#POINT = 1
DO #POINT = 1 TO #LASTLINE
   "ISREDIT SHIFT ) "#POINT" 15"
```

```
END
"ISREDIT EXCLUDE '.END' 1 4 ALL"
"ISREDIT DELETE EXCLUDE ALL"
/* COPY SOURCE TO ZZTEST MEMBER */
"ISREDIT REPLACE ZZTEST 1 "#LASTLINE
"ISREDIT ("DSN") = DATASET"
ADDRESS TSO
"ALLOC DA('"DSN"(ZZTEST)') F(INDD) SHR REUSE"
"EXECIO * DISKR INDD (STEM INBASE. FINIS)"
"FREE F(INDD)"
/* DYNAMICALLY BUILD REXX/MACRO TO ADD SYSIN TO DEFAULT SIT SOURCE */
FLAG = Ø
#Q = "'"
I = 1
J = 1
#OUTLINE.J = '/* REXX */'
J = J + 1
#OUTLINE.J = 'ADDRESS "ISREDIT"'
J = J + 1
DO I = 1 TO INBASE.Ø
   PARSE VAR INBASE.I #CHECK #REST
      #CHECKND = #CHECK
      #LNND = POS('=',#CHECK)
      #END = #LNND + 15
      #CHECK = LEFT(#CHECK,#LNND)
      IF LEFT(#CHECK,6) = 'GMTEXT' THEN DO
         FLAG = 1
         I = I + 1
      END
      IF LEFT(#CHECK,8) = 'INITPARM' THEN DO
         FLAG = 1
         IF SUBSTR(#CHECKND,56,1) ¬= ' ' THEN I = I + 1
      END
      IF FLAG = Ø THEN
         DO
         #OUTLINE.J = '"ISREDIT CHANGE '||,
                      #Q||,
                      #CHECK||,
                      #Q||,
                      ' '||,
                      #Q||,
                      '@@@@@@'||,
                      #Q||,
                      ' 15 '||,
                      #END||,
                      ' ALL "'

         J = J + 1
      END
   FLAG = Ø
```

```
END
#OUTLINE.J = '"ISREDIT EXCLUDE '||#Q||,
             '@@@@@@'||,
             #Q||,
             ' ALL"'
J = J + 1
#OUTLINE.J = '"ISREDIT DELETE EXCLUDE ALL"'
J = J + 1
#OUTLINE.J = '"ISREDIT FIND '||#Q||,
             'TYPE=CSECT,'||,
             #Q||,
             ' FIRST"'
J = J + 1
#OUTLINE.J = '   "ISREDIT ("LN","CN") = CURSOR"'
J = J + 1
#OUTLINE.J = '   "ISREDIT COPY ZZTEST AFTER "LN'
J = J + 1
#OUTLINE.J = 'MEND'
#OUTLINE.Ø = J
DO J = 1 TO #OUTLINE.Ø
   QUEUE #OUTLINE.J
END
QUEUE ''
"ALLOC DA('TGS.A.CLIST(CHKSITXX)') F(BASEFILE) SHR REUSE"
"EXECIO * DISKW BASEFILE (FINIS)"
ADDRESS "ISREDIT"
/* SET DEFAULT SIT SOURCE MEMBER */
DEFAULT = LEFT(MEM,4)||'CIØØ'
/* CLEAR DOWN SYSIN MEMBER */
"ISREDIT DELETE NX ALL"
/* COPY DEFAULT SIT SOURCE */
"ISREDIT COPY "DEFAULT" AFTER Ø"
/* EXECUTE REXX/MACRO JUST BUILT TO APPEND SYSIN */
CALL CHKSITXX
/* COPY SOURCE TO ZZZTEST MEMBER */
"ISREDIT FIND 'END   DFHSITBA' LAST"
"ISREDIT ("LND","CND") = CURSOR"
"ISREDIT REPLACE ZZZTEST 1 "LND
/* CANCEL ALL CHANGES AGAIST THE SYSIN */
"ISREDIT CANCEL"
/* SUB JOB TO ASSEMBLE ZZTEST/ZZZTEST SOURCE */
ADDRESS TSO
LOGONID = SYSVAR(SYSUID)
"ISPEXEC FTOPEN TEMP"
"ISPEXEC FTINCL SYSINCHK"
"ISPEXEC FTCLOSE"
"ISPEXEC VGET (ZTEMPF)"
"SUBMIT '"ZTEMPF"'"
EXIT
```

SYSINCHK SKELETON JCL SOURCE

```
//&LOGONID.CK JOB 1417,'&MEM',NOTIFY=&SYSUID,
//          CLASS=7,MSGCLASS=X,MSGLEVEL=(1,1)
//*
//*****************************************************************
//* SIT ASSEMBLE CHECK UTILITY (SYSIN CARDS)
//*****************************************************************
//*
//ASM1       EXEC PGM=IEV90,REGION=4096K,
//           PARM='SYSPARM(INITIAL),DECK,NOOBJECT,TERM'
//SYSPRINT   DD  SYSOUT=*
//SYSTERM    DD  SYSOUT=*
//SYSLIB     DD  DSN=*CICS*.SDFHMAC,DISP=SHR
//           DD  DSN=*CICS*.USERMAC,DISP=SHR
//           DD  DSN=SYS*.MACLIB,DISP=SHR
//           DD  DSN=SYS*.MODGEN,DISP=SHR
//SYSUT1     DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2     DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3     DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSIN      DD  *
         DFHSIT TYPE=CSECT,
//           DD  DSN=&DSN(ZZTEST),DISP=SHR
//           DD  *
               XUSER=NO
         END   DFHSITBA
/*
//SYSPUNCH   DD  DSN=&&OBJMOD,
//           DISP=(,PASS),UNIT=SYSDA,
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=19040),
//           SPACE=(CYL,(3,1))
//*
//*
//*****************************************************************
//* SIT ASSEMBLE CHECK UTILITY (SYSIN CARDS + DEFAULT SIT)
//*****************************************************************
//*
//ASM2       EXEC PGM=IEV90,REGION=4096K,
//           PARM='SYSPARM(INITIAL),DECK,NOOBJECT,TERM'
//SYSPRINT   DD  SYSOUT=*
//SYSTERM    DD  SYSOUT=*
//SYSLIB     DD  DSN=*CICS*.SDFHMAC,DISP=SHR
//           DD  DSN=*CICS*.USERMAC,DISP=SHR
//           DD  DSN=SYS*.MACLIB,DISP=SHR
//           DD  DSN=SYS*.MODGEN,DISP=SHR
//SYSUT1     DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2     DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT3     DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSIN      DD  *
//           DD  DSN=&DSN(ZZZTEST),DISP=SHR
/*
```

```
//SYSPUNCH   DD  DSN=&&OBJMOD2,
//           DISP=(,PASS),UNIT=SYSDA,
//           DCB=(RECFM=FB,LRECL=8Ø,BLKSIZE=19Ø4Ø),
//           SPACE=(CYL,(3,1))
//*
```

EXAMPLE OF SYSIN

```
*2Ø/Ø1/Ø2  PT   >>>>>   SYSIN BUILT   <<<<<<                          *
*24/Ø2/Ø2 #79693 PT ADD PRVMOD PARM FOR LPA CHANGES   *
*
APPLID=(LQØ1CIB3),            CICS APPL ID
GMTEXT='Welcome to LQØ1CIB3 Any problems call SMC 9999 9999',
GRPLIST=(DFHLIST,COMLIST,LSTQØ1B3,SYS1Q1B3),
INITPARM=(DFHD2INI='DBAA'),
PRVMOD=(DFHAIP,DFHAPLH,DFHAPXM,DFHBRTB,DFHBRTQ,DFHDHRP,DFHDLI,
DFHEIQDH,DFHEIQOP,DFHEIQSO,DFHEISO,DFHEITHG,DFHERM,DFHJCP,DFHLIRET,
DFHPCPC2,DFHSOSE,DFHTDP,DFHXFP,DFHZBAN,IBMBPSLA,IBMBPSMA,DFHDUIO,
DADSPIØ1,DFHPSP,DFHSOL,DFHTRAO,DFHCNV,DFHTDTM,DFHAPIN),
SYSIDNT=Q1B3,                 SYSID
.END
```

EXAMPLE OF DYNAMICALLY BUILT CHKSITXX MEMBER

```
/* REXX */
ADDRESS "ISREDIT"
"ISREDIT CHANGE 'APPLID=' '@@@@@@' 15 22 ALL "
"ISREDIT CHANGE 'GRPLIST=' '@@@@@@' 15 23 ALL "
"ISREDIT CHANGE 'SYSIDNT=' '@@@@@@' 15 23 ALL "
"ISREDIT EXCLUDE '@@@@@@' ALL"
"ISREDIT DELETE EXCLUDE ALL"
"ISREDIT FIND 'TYPE=CSECT,' FIRST"
   "ISREDIT ("LN","CN") = CURSOR"
   "ISREDIT COPY ZZTEST AFTER "LN
MEND
```

EXAMPLE OF BUILT ZZTEST MEMBER

```
APPLID=(LQØ1CIB3),                                      X
GMTEXT='Welcome to LQØ1CIB3 Any problems call SMC 9999 9X
999',                                                   X
GRPLIST=(DFHLIST,COMLIST,LSTQØ1B3,SYS1Q1B3),            X
INITPARM=(DFHD2INI='DBAA'),                             X
SYSIDNT=Q1B3,                                           X
```

Phil R Taylor
CICS Systems Programmer
HSBC plc (UK)

CICS questions and answers

Q Is there any benefit from running with LPA=YES set in the SIT if you don't apply the supplied usermod to move selected read-only modules into the LPA? (We haven't done so because we are running Test and User CICS on the same LPAR and therefore wouldn't be able to manage maintenance roll-ups so easily.) We have set LPA=YES but then listed a whole selection of CICS modules in the PRVMOD parameter, which seems daft if there is a CPU benefit in setting LPA=NO and not searching through the PRVMOD list.

A LPA=YES isn't just for CICS modules; application programs (defined with LPACOPY=YES) may reside in the LPA and therefore this option is needed. If you don't need to load anything from the LPA then LPA=NO is the best option.

You do gain many benefits from having the optional modules in the LPA, eg faster CICS start-up/running. This is because the modules don't need to be loaded, freeing-up DSA storage in all CICS regions and therefore reducing the overall MVS storage requirements for all CICS.

As to your maintenance issue on a single LPAR: many sites run with LPA=YES in all CICS, and apply the usermod to move the optional modules, but then PRVMOD the optional modules in the 'Test/Development' CICS so that any changes to the modules can be tested from an RPL copy of SDFHLPA. Then when the maintenance is copied to the 'User/QA' regions (without PRVMOD) the modules are also copied to LPA.

You need to manage changes to the required SDFHLPA modules, so your maintenance 'roll-out' procedure remains the same, apart from making an RPL version of the SDFHLPA library for your 'Test/Development' systems.

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.

IBM has announced Version 6.0 of its VisualAge Smalltalk Server for OS/390 and z/OS, the run-time deployment environment for Smalltalk Server applications on OS/390 and z/OS.

With VisualAge Smalltalk Enterprise and VisualAge Server Workbench, sites can write transactions for CICS and IMS, access DB2, VSAM files, and QSAM files, interface with applications using MQSeries TCP/IP, APPC, and CPI-C, and interface with COBOL, C, and PL/I applications.

It also enables sites to run XML applications, interface with Web servers using servlets, run applications that host or invoke Web services, run applications with VisualAge Smalltalk Web Connection, and develop on Windows and OS/2.

Developers can create applications on the workstation using the same development tools familiar to VisualAge Smalltalk programmers. These client/server applications can then be deployed to CICS/ESA, IMS/ESA, z/OS, OS/390, or WebSphere Application Server for z/OS and OS/390 with VisualAge Smalltalk Server for OS/390 and z/OS.

For further information contact your local IBM representative.
URL: http://www.ibm.com/software/ad/smalltalk.

* * *

NEON Systems has announced that it has added BEA WebLogic Platform 7.0 support to its ShadowConnect and ShadowDirect products.

With this integration, the NEON and BEA solutions provide JCA, JDBC, or ODBC access to mainframe data sources and transactional environments supporting CICS, IMS, DB2,ADABAS, Natural, IDMS, and flat files.

ShadowDirect J2EE connectors for z/OS are now certified for BEA WebLogic.

For further information contact:
NEON Systems**,** 14100 Southwest Freeway, Suite 500, Sugar Land, TX 77478, USA.
Tel: (281) 491 4200.
NEON Systems UK, No. 1 High Street, Windsor, Berkshire SL4 1LD, UK.
Tel: (01753) 752800.
URL: http://www.neonsys.com.

* * *