175 CICS update

June 2000

In this issue

CICS Update

Published by

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

North American office

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

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £175.00 in the UK; $270.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £16.00 ($23.50) each including postage.

CICS Update on-line

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

Editor

Trevor Eddolls

Disclaimer

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

Contributions

Articles published in *CICS Update* are paid for at the rate of £170 ($260) per 1000 words and £100 ($160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 ($80) per 100 lines. In addition, there is a flat fee of £30 ($50) per article. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*, or you can download a copy from www.xephon.com/contnote.html.

Printed in England.

Managing CICS Shared TS Queues

Now that CICS temporary storage affinity has been resolved by the introduction of the shared temporary storage server, managing temporary storage queues in the shared server now becomes a challenge. At our company, we use temporary storage a lot. For us to get CICS 24x7 was going to be challenge, until the shared TSQ server came along.

Here is our current problem. We do different processing at different times of the month. Each application creates various numbers of temporary storage queues. As you know, programmers don't always code their programs to delete the queues when they are done with them. So an application that runs at the beginning of the month will create temporary storage queues and then not use them for the rest of the month. If you have a lot of applications written to use the TSQ server, this could eat up a lot of storage in the TSQ server, which resides in the coupling facility. And now that the regions and the TSQ server are up 24x7, these queues will sit there for a month before being used again.

We had to come up with a way to manage those queues that were inactive for a certain period of time. To do this, we wrote a CPSM program, which helps us manage the shared queues. Basically, the batch program will get all the queues for a specific CONTEXT and SCOPE and put them in storage. It will then examine each queue individually and look at the LAST USED TIME. The LAST USED TIME is just that – the last time the queue was referenced. If you add to the queue or even read the queue, this last used time is reset to zero.

The program then checks whether the time is greater than a specified time, which is eight hours in this program. If it is, it will MARK the queue for deletion, then it will examine the next queue until all the queues have been examined. After that, I run a DELETE that deletes all the MARKed queues. I then create a nice little report that shows all the queues that we examined and whether they have been deleted or not.

There may be some queues that you do not want to delete ever, no matter what the last used time may be. For this kind of situation, you

```
                SUMMARY REPORT FOR CONTEXT: CICSPLXT  SCOPE: CICSPLXT

                                         LAST USED    CPSM   RESPONSE  REASON
   TSQ NAME         REGION     POOL     IN SECONDS  DELETED  ACTION    CODE     CODE
------------      --------   --------   ----------  -------  ------  --------  ------
 *SUCCESSFUL*                                                 CONNCT
 *SUCCESSFUL*                                                  GET
 THIS IS A TEST 1  CICSTGD    CICTSQT     Ø36214       YES
 THIS IS A TEST 2  CICSTGD    CICTSQT     Ø36212       YES
 THIS IS A TEST 3  CICSTGD    CICTSQT     Ø36211       YES
 THIS IS A TEST 4  CICSTGD    CICTSQT     Ø3621Ø       YES
 XX99              CICSTGD    CICTSQT     ØØ4531       NO
 1SS9              CICSTGD    CICTSQT     ØØ4271       NO
 1X99              CICSTGD    CICTSQT     ØØ4292       NO
 THIS IS A TEST 1  CICSTGT    CICTSQT     Ø36214       YES
 THIS IS A TEST 2  CICSTGT    CICTSQT     Ø36212       YES
 THIS IS A TEST 3  CICSTGT    CICTSQT     Ø36211       YES
 THIS IS A TEST 4  CICSTGT    CICTSQT     Ø3621Ø       YES
 XX99              CICSTGT    CICTSQT     ØØ4531       NO
 1SS9              CICSTGT    CICTSQT     ØØ4271       NO
 1X99              CICSTGT    CICTSQT     ØØ4292       NO
 *SUCCESSFUL*                                                 DELETE
 *SUCCESSFUL*                                                 DISCON
```

Figure 1: Example report

will need to add some logic to the program in the MARK section. I explain in the comment section where to add this logic. You would just need to check the queue names for the ones you do not want to mark, to force it to skip the actual mark logic of the program.

I wrote this program so that you can enter scope and context as a parameter in the JCL. Doing this will allow you to be flexible enough to put in the name of a CICSPlex, a group of regions, or a single CICS region. I have put a sample of the JCL at the end of this article.

An example report is shown in Figure 1.

SYS100

```
IDENTIFICATION DIVISION.
       PROGRAM-ID.    SYS100TS.

       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.

      ****************************************************************
      * DATE: 2/10/2000
      * DESCRIPTION:  THIS PROGRAM WILL CHECK THE 'TIME LAST USED' FOR
      *               ALL SHARED TEMP STORAGE AND DELETE THEM AFTER
      *               8 HOURS OF NON-USE.
      ****************************************************************
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT PRINT-FILE-OUT   ASSIGN TO UT-S-PRINT01.
      ****************************************************************
       DATA DIVISION.
       FILE SECTION.

       FD  PRINT-FILE-OUT
           LABEL RECORDS ARE STANDARD
           RECORDING MODE IS F
           BLOCK CONTAINS 0 RECORDS.
       01  PRINT-LINE-OUT             PIC X(133).
      ****************************************************************
       WORKING-STORAGE SECTION.

       01  T1-TTILE-LINE.
           05  FILLER                 PIC X(20)   VALUE SPACES.
           05  FILLER                 PIC X(28)   VALUE
               'SUMMARY REPORT FOR CONTEXT: '.
           05  T1-CONTEXT             PIC X(8)    VALUE SPACES.
           05  FILLER                 PIC X(9)    VALUE '  SCOPE: '.
           05  T1-SCOPE               PIC X(8)    VALUE SPACES.
```

```
    05  FILLER                 PIC X(60)   VALUE SPACES.

01  H1-HEADER-LINE.
    05  FILLER                 PIC X(43)   VALUE SPACES.
    05  FILLER                 PIC X(9)    VALUE 'LAST USED'.
    05  FILLER                 PIC X(15)   VALUE SPACES.
    05  FILLER                 PIC X(4)    VALUE 'CPSM'.
    05  FILLER                 PIC X(4)    VALUE SPACES.
    05  FILLER                 PIC X(8)    VALUE 'RESPONSE'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(6)    VALUE 'REASON'.
    05  FILLER                 PIC X(41)   VALUE SPACES.

01  H2-HEADER-LINE.
    05  FILLER                 PIC X(6)    VALUE SPACES.
    05  FILLER                 PIC X(8)    VALUE 'TSQ NAME'.
    05  FILLER                 PIC X(8)    VALUE SPACES.
    05  FILLER                 PIC X(6)    VALUE 'REGION'.
    05  FILLER                 PIC X(6)    VALUE SPACES.
    05  FILLER                 PIC X(4)    VALUE 'POOL'.
    05  FILLER                 PIC X(5)    VALUE SPACES.
    05  FILLER                 PIC X(10)   VALUE 'IN SECONDS'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(7)    VALUE 'DELETED'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(6)    VALUE 'ACTION'.
    05  FILLER                 PIC X(5)    VALUE SPACES.
    05  FILLER                 PIC X(4)    VALUE 'CODE'.
    05  FILLER                 PIC X(6)    VALUE SPACES.
    05  FILLER                 PIC X(4)    VALUE 'CODE'.
    05  FILLER                 PIC X(42)   VALUE SPACES.

01  H3-HEADER-LINE.
    05  FILLER                 PIC X(2)    VALUE SPACES.
    05  FILLER                 PIC X(16)   VALUE
        '----------------'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(8)    VALUE '--------'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(8)    VALUE '--------'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(10)   VALUE '----------'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(7)    VALUE '-------'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(6)    VALUE '------'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(8)    VALUE '--------'.
    05  FILLER                 PIC X(3)    VALUE SPACES.
    05  FILLER                 PIC X(6)    VALUE '------'.
    05  FILLER                 PIC X(41)   VALUE SPACES.
```

```
Ø1  D1-DETAIL-LINE.
    Ø5  FILLER                PIC X(2)    VALUE SPACES.
    Ø5  D1-NAME               PIC X(16)   VALUE SPACES.
    Ø5  FILLER                PIC X(3)    VALUE SPACES.
    Ø5  D1-REGION             PIC X(8)    VALUE SPACES.
    Ø5  FILLER                PIC X(3)    VALUE SPACES.
    Ø5  D1-POOL               PIC X(8)    VALUE SPACES.
    Ø5  FILLER                PIC X(5)    VALUE SPACES.
    Ø5  D1-LAST-USED          PIC X(6)    VALUE ZERO.
    Ø5  FILLER                PIC X(7)    VALUE SPACES.
    Ø5  D1-DELETED            PIC X(3)    VALUE ZERO.
    Ø5  FILLER                PIC X(5)    VALUE SPACES.
    Ø5  D1-ACTION             PIC X(6)    VALUE SPACES.
    Ø5  FILLER                PIC X(5)    VALUE SPACES.
    Ø5  D1-RESP               PIC X(4)    VALUE ZERO.
    Ø5  FILLER                PIC X(6)    VALUE SPACES.
    Ø5  D1-REASON             PIC X(4)    VALUE ZERO.
    Ø5  FILLER                PIC X(42)   VALUE SPACES.

Ø1  SWITCH-SW.
    Ø5  ERROR-SW              PIC X       VALUE 'N'.

Ø1  SAVE-AREAS-SA.
    Ø5  CPSM-CONTEXT-SA       PIC X(8)    VALUE SPACES.
    Ø5  CPSM-SCOPE-SA         PIC X(8)    VALUE SPACES.
    Ø5  CPSM-OBJNAME-SA       PIC X(8)    VALUE SPACES.
    Ø5  CPSM-ACTION-SA        PIC X(12)   VALUE SPACES.
    Ø5  CPSM-CRITERIA-SA      PIC X(1ØØ)  VALUE SPACES.
    Ø5  CPSM-CRITLEN-SA       PIC S9(8)   USAGE BINARY.
    Ø5  CPSM-THREAD-SA        PIC S9(8)   USAGE BINARY.
    Ø5  CPSM-RESP-SA          PIC S9(8)   USAGE BINARY.
    Ø5  CPSM-REAS-SA          PIC S9(8)   USAGE BINARY.
    Ø5  CPSM-RESULT-SA        PIC S9(8)   USAGE BINARY.
    Ø5  CPSM-COUNT-SA         PIC S9(8)   USAGE BINARY.
    Ø5  TEMP-CRIT-SA          PIC X(7)    VALUE 'NAME=*.'.
    Ø5  HOLD-TIME-SA          PIC 9(5)    VALUE ZERO.
    Ø5  HOLD-LAST-USED        PIC 9(6)    VALUE ZERO.
    Ø5  HOLD-RESP             PIC 9(4)    VALUE ZERO.
    Ø5  HOLD-REASON           PIC 9(4)    VALUE ZERO.
    Ø5  HOLD-ACTION           PIC X(6)    VALUE SPACES.
    Ø5  DELETE-TSQ-SA.
        1Ø  FILLER            PIC X(5)    VALUE 'NAME='.
        1Ø  DELETE-Q-NAME-SA  PIC X(16)   VALUE SPACES.
        1Ø  FILLER            PIC X(1)    VALUE '.'.

Ø1  FETCH-SA.
    Ø5  FETCH-CICS-SA         PIC X(8).
    Ø5  FETCH-REL-SA          PIC X(4).
    Ø5  FETCH-EYU-RESERVED    PIC X(4).
    Ø5  FETCH-TSQ-NAME-SA     PIC X(16).
    Ø5  FETCH-POOLNAME-SA     PIC X(8).
    Ø5  FETCH-LOCATION-SA     PIC S9(8)   USAGE BINARY.
```

```
           05  FETCH-QUELENGTH        PIC S9(8)   USAGE BINARY.
           05  FETCH-MAXITEMLEN       PIC S9(4)   USAGE BINARY.
           05  FETCH-MINITEMLEN       PIC S9(4)   USAGE BINARY.
           05  FETCH-NUMITEMS         PIC S9(4)   USAGE BINARY.
           05  FETCH-EYU-RSV0009      PIC X(2).
           05  FETCH-LAST-USED-SA     PIC S9(8)   USAGE BINARY.
           02  FETCH-TRANSID          PIC X(4).
           02  FETCH-RECOVSTATUS      PIC S9(8)   USAGE BINARY.
           02  FETCH-EYU-RSV0257      PIC X(4).

       01  SUBSCRIPTS-SS.
           05  COUNT-SS               PIC 9(5)    VALUE ZERO.
           05  LINE-CNT-SS            PIC 9(2)    VALUE ZERO.

       01  PARM-OPTION.
           05  PARM-LENGTH            PIC 9(4)    COMP.
           05  PARM-VALUE             PIC X(17)   VALUE SPACES.
           05  FILLER                 PIC X(66)   VALUE SPACES.

       LINKAGE SECTION.
      *****************************************************************
       PROCEDURE DIVISION.

       000-MAIN.
           OPEN OUTPUT  PRINT-FILE-OUT.

           MOVE DFHEIBLK         TO PARM-OPTION.
           UNSTRING PARM-VALUE DELIMITED BY ','
               INTO CPSM-CONTEXT-SA CPSM-SCOPE-SA.

           PERFORM 100-PRINT-NEW-PAGE.
           PERFORM 200-PROCESS-TSQ.

           MOVE  SPACES          TO PRINT-LINE-OUT.
           WRITE PRINT-LINE-OUT.
           MOVE  'END OF REPORT' TO PRINT-LINE-OUT.
           WRITE PRINT-LINE-OUT.
           MOVE  SPACES          TO PRINT-LINE-OUT.
           WRITE PRINT-LINE-OUT.
           CLOSE PRINT-FILE-OUT.

           STOP RUN.
      *****************************************************************
      *  THIS SECTION PRINTS JUST THE HEADERS ON THE PAGE             *
      *****************************************************************
       100-PRINT-NEW-PAGE.
           MOVE  SPACES            TO PRINT-LINE-OUT.
           WRITE PRINT-LINE-OUT.
           WRITE PRINT-LINE-OUT.
           MOVE  CPSM-CONTEXT-SA   TO T1-CONTEXT.
           MOVE  CPSM-SCOPE-SA     TO T1-SCOPE.
```

```
     MOVE  T1-TTILE-LINE     TO PRINT-LINE-OUT.
     WRITE PRINT-LINE-OUT.

     MOVE  SPACES            TO PRINT-LINE-OUT.
     WRITE PRINT-LINE-OUT.
     MOVE  H1-HEADER-LINE    TO PRINT-LINE-OUT.
     WRITE PRINT-LINE-OUT.
     MOVE  H2-HEADER-LINE    TO PRINT-LINE-OUT.
     WRITE PRINT-LINE-OUT.
     MOVE  H3-HEADER-LINE    TO PRINT-LINE-OUT.
     WRITE PRINT-LINE-OUT.

     MOVE 7                  TO LINE-CNT-SS.
*****************************************************************
*  THIS IS THE MAIN PART OF THE PROGRAM.                        *
*****************************************************************
 200-PROCESS-TSQ.
     PERFORM 400-CONNECT-TO-CPSM.
     IF ERROR-SW IS EQUAL TO 'N'
        PERFORM 500-GET-TSQSHR
        IF ERROR-SW IS EQUAL TO 'N'
           PERFORM 700-DELETE-RECORD.

     PERFORM 800-DISCONNECT-FROM-CPSM.
*****************************************************************
*  THIS SECTION DOES THE ERROR CHECKING.                        *
*  A RESPONSE OF 1024 IS A NORMAL RESPONSE.                     *
*  I CHECK FOR MARK AND FETCH AND I DO NOT PRINT A RECORD       *
*  FOR EACH OF THESE.  IF I DID, THE OUTPUT FILE COULD GET      *
*  EXTREMELY LARGE.                                             *
*  A RESPONSE OF 1027 FOR A DELETE MEANS THERE IS NOTHING TO    *
*  DELETE (NONE OF THE QUEUES MET THE TIME REQUIREMENT OF       *
*  EIGHT HOURS).                                                *
*  ALL OTHER RESPONSES ARE CONSIDERED AN ERROR AND PROCESSING   *
*  STOPS.                                                       *
*****************************************************************
 300-ERROR-CHECKING.
     MOVE HOLD-ACTION        TO D1-ACTION.
     MOVE SPACES             TO D1-REGION
                                D1-POOL
                                D1-DELETED
                                D1-RESP
                                D1-REASON
                                D1-LAST-USED.

     MOVE CPSM-RESP-SA       TO HOLD-RESP
     MOVE CPSM-REAS-SA       TO HOLD-REASON
     IF HOLD-RESP IS EQUAL TO '1024'
        MOVE '  *SUCCESSFUL*'      TO D1-NAME
        MOVE  D1-DETAIL-LINE       TO PRINT-LINE-OUT
        WRITE PRINT-LINE-OUT
```

```
      ELSE
         IF HOLD-RESP IS EQUAL TO '1027'
            MOVE 'Y'                TO ERROR-SW
            IF HOLD-ACTION IS EQUAL TO 'DELETE'
               MOVE 'NOTHING 2 DELETE'
                                    TO D1-NAME
               MOVE  D1-DETAIL-LINE TO PRINT-LINE-OUT
               WRITE PRINT-LINE-OUT
            ELSE
               MOVE 'NOTHING 2 GET' TO D1-NAME
               MOVE  D1-DETAIL-LINE TO PRINT-LINE-OUT
               WRITE PRINT-LINE-OUT
         ELSE
            MOVE HOLD-RESP          TO D1-RESP
            MOVE HOLD-REASON        TO D1-REASON
            MOVE '  ***ERROR***'    TO D1-NAME
            MOVE 'Y'                TO ERROR-SW
            MOVE  D1-DETAIL-LINE    TO PRINT-LINE-OUT
            WRITE PRINT-LINE-OUT.
*****************************************************************
* THIS IS WHERE THE CONNECTION IS ESTABLISHED TO CPSM.  NOTICE *
* WE ARE RUNNING CPSM VERSION 1.4.0.                            *
*****************************************************************
 400-CONNECT-TO-CPSM.
     MOVE ZERO        TO CPSM-RESP-SA
                         CPSM-REAS-SA.

     EXEC CPSM CONNECT
               CONTEXT(CPSM-CONTEXT-SA)
               SCOPE(CPSM-SCOPE-SA)
               VERSION('0140')
               THREAD(CPSM-THREAD-SA)
               RESPONSE(CPSM-RESP-SA)
               REASON(CPSM-REAS-SA)
     END-EXEC.

     MOVE 'CONNCT'    TO HOLD-ACTION.
     PERFORM 300-ERROR-CHECKING.
*****************************************************************
* IN THIS SECTION, WE GET ALL THE QUEUE FOR THE SPECIFIED      *
* CONTEXT AND SCOPE.                                            *
* A GET RETURNS A RESULT SET CONTAINING SELECTED RESOURCE      *
* TABLE RECORDS.                                                *
* AFTER THE GET, THE RECORD POINTER IS AT THE TOP OF THE       *
* RESULT SET, WHICH IS USED IN THE NEXT SECTION.               *
*****************************************************************
 500-GET-TSQSHR.
     MOVE ZERO           TO CPSM-RESP-SA
                            CPSM-REAS-SA.

     MOVE TEMP-CRIT-SA   TO CPSM-CRITERIA-SA.
```

```
     MOVE 7              TO CPSM-CRITLEN-SA.
     MOVE 'TSQSHR'       TO CPSM-OBJNAME-SA.

     EXEC CPSM GET OBJECT(CPSM-OBJNAME-SA)
               CRITERIA  (CPSM-CRITERIA-SA)
               LENGTH    (CPSM-CRITLEN-SA)
               RESULT    (CPSM-RESULT-SA)
               COUNT     (CPSM-COUNT-SA)
               THREAD    (CPSM-THREAD-SA)
               RESPONSE  (CPSM-RESP-SA)
               REASON    (CPSM-REAS-SA)
     END-EXEC.

     MOVE ' GET  '       TO HOLD-ACTION.
     PERFORM 300-ERROR-CHECKING.

     IF ERROR-SW IS EQUAL TO 'N'
        MOVE 0 TO COUNT-SS
        PERFORM 600-FETCH-DATA
           UNTIL COUNT-SS = CPSM-COUNT-SA
              OR ERROR-SW IS EQUAL TO 'Y'.
*****************************************************************
* IN THIS SECTION, EACH RECORD IS FETCHED INTO A WORKING       *
* STORAGE FIELD.                                                *
*****************************************************************
 600-FETCH-DATA.
     ADD 1     TO COUNT-SS.
     MOVE 72   TO CPSM-CRITLEN-SA.

     MOVE ZERO TO CPSM-RESP-SA
                  CPSM-REAS-SA.

     EXEC CPSM FETCH
               INTO      (FETCH-SA)
               LENGTH    (CPSM-CRITLEN-SA)
               RESULT    (CPSM-RESULT-SA)
               THREAD    (CPSM-THREAD-SA)
               RESPONSE  (CPSM-RESP-SA)
               REASON    (CPSM-REAS-SA)
     END-EXEC.

     MOVE 'FETCH '              TO HOLD-ACTION.
     IF CPSM-RESP-SA IS NOT EQUAL TO 1024
        PERFORM 300-ERROR-CHECKING.

     IF ERROR-SW IS EQUAL TO 'N'
        PERFORM 610-PROCESS-DATA.
*****************************************************************
* IN THIS SECTION, THE RECORD IS PRINTED, WHETHER IT IS TO BE  *
* DELETED OR NOT.                                               *
* THE LAST USED TIME IS COMPARED TO 21000 (8 HOURS).  IF IT    *
```

```
* IS GREATER THAN THAT, THEN THE SECTION THAT MARKS THE        *
* RECORD IS CALLED.                                            *
****************************************************************
 610-PROCESS-DATA.
     MOVE FETCH-TSQ-NAME-SA  TO D1-NAME.
     MOVE FETCH-LAST-USED-SA TO HOLD-LAST-USED.
     MOVE HOLD-LAST-USED     TO D1-LAST-USED.
     MOVE FETCH-CICS-SA      TO D1-REGION.
     MOVE FETCH-POOLNAME-SA  TO D1-POOL.

     MOVE 21000              TO HOLD-TIME-SA.
     IF FETCH-LAST-USED-SA IS GREATER THAN HOLD-TIME-SA
        MOVE 'YES'           TO D1-DELETED
*
* ADD THIS NEXT SECTION TO SKIP A QUEUE YOU WANT TO KEEP.
*
*       IF FETCH-TSQ-NAME-SA IS EQUAL TO 'XXXXXXXXXXXXXXXX'
*          MOVE 'NO'         TO D1-DELETED
*       ELSE
*
        PERFORM 620-MARK-RECORD
     ELSE
        MOVE 'NO'            TO D1-DELETED.

     IF ERROR-SW IS EQUAL TO 'N'
        PERFORM 630-WRITE-THE-RECORD.
****************************************************************
* THIS SECTION MARKS THE CURRENT RECORD.  THE CURRENT RECORD   *
* IS WHERE THE RECORD POINTER IS CURRENTLY LOCATED.  AFTER THE *
* RECORD IS MARKED, YOU CAN DO A LOT OF DIFFERENT THINGS.  WE  *
* ARE JUST GOING TO DELETE IT IN THE NEXT SECTION.             *
****************************************************************
 620-MARK-RECORD.
     MOVE ZERO        TO CPSM-RESP-SA
                         CPSM-REAS-SA.

     EXEC CPSM MARK
               CURRENT
               REASON        (CPSM-REAS-SA)
               RESULT        (CPSM-RESULT-SA)
               THREAD        (CPSM-THREAD-SA)
               RESPONSE      (CPSM-RESP-SA)
     END-EXEC.

     MOVE ' MARK '    TO HOLD-ACTION.
     IF CPSM-RESP-SA IS NOT EQUAL TO 1024
         PERFORM 300-ERROR-CHECKING.
****************************************************************
* THIS SECTION JUST WRITES THE RECORD TO THE PRINT FILE.       *
****************************************************************
 630-WRITE-THE-RECORD.
```

```
     MOVE SPACES             TO D1-ACTION
                                D1-RESP
                                D1-REASON.

     IF LINE-CNT-SS IS GREATER THAN 65
        PERFORM 100-PRINT-NEW-PAGE.

     MOVE  D1-DETAIL-LINE    TO PRINT-LINE-OUT.
     WRITE PRINT-LINE-OUT.
     ADD 1                   TO LINE-CNT-SS.
*****************************************************************
* AFTER THE WHOLE STACK IS SEARCHED, THIS SECTION IS CALLED    *
* AND ALL THE RECORDS THAT HAVE BEEN MARKED WILL NOW BE        *
* DELETED FROM THE TEMPORARY STORAGE SERVER.                   *
*****************************************************************
 700-DELETE-RECORD.
     MOVE ZERO        TO CPSM-RESP-SA
                         CPSM-REAS-SA.
     MOVE 'DELETE'    TO CPSM-ACTION-SA.
     EXEC CPSM PERFORM SET
               ACTION        (CPSM-ACTION-SA)
               MARKED
               REASON        (CPSM-REAS-SA)
               RESULT        (CPSM-RESULT-SA)
               THREAD        (CPSM-THREAD-SA)
               RESPONSE      (CPSM-RESP-SA)
     END-EXEC.

     MOVE 'DELETE'    TO HOLD-ACTION.
     PERFORM 300-ERROR-CHECKING.
*****************************************************************
* THIS SECTION DISCONNECTS FROM CPSM.                          *
*****************************************************************
 800-DISCONNECT-FROM-CPSM.
     MOVE ZERO TO CPSM-RESP-SA
                  CPSM-REAS-SA.

     EXEC CPSM DISCONNECT
               THREAD(CPSM-THREAD-SA)
               RESPONSE(CPSM-RESP-SA)
               REASON(CPSM-REAS-SA)
     END-EXEC.

     MOVE 'DISCON'    TO HOLD-ACTION.
     PERFORM 300-ERROR-CHECKING.
```

SAMPLE JCL

This is a sample of the JCL used to execute this program:

```
//JOBCARD HERE
//*********************************************************************
//*   Parm is CONTEXT,SCOPE
//*********************************************************************
//SYS198P0   EXEC PGM=SYS100,PARM=('CICSPLXT,CICSPLXT')
//STEPLIB    DD DSN=SYS.LINK.LIB,DISP=SHR
//PRINT01    DD SYSOUT=T,OUTLIM=10000
//SYSPRINT   DD SYSOUT=T
```

Jon McCabe
CICS Systems Programmmer
CUNA Mutual Group (USA)

CICS/TS 1.3 NEWCOPY facility for DOCTEMPLATES – revisited

Since I wrote the article *CICS/TS 1.3 NEWCOPY facility for DOCTEMPLATES* , *CICS Update*, Issues 173 and 174, April and May 2000, a couple of points need to be updated in the light of experience.

CICS TS 1.3 DFHHTML library now supports concatenations and secondary extents!

With respect to the NEWCOPY facility – it is the directory records of the DFHHTML members that are being cached in memory, not the members themselves. With the NEWCOPY facility I've provided, by discarding a definition and creating a new one, the directory information is effectively updated in CICS.

Currently CICS does I/O to the PDS member each and every time that the template is referenced (potentially very expensive). Also, as supplied, CICS is using a 2KB buffer, so if your template is very large there may be multiple I/Os to load one template.

APAR PQ33080 (13/03/2000) is now available to change the buffer size to 32KB to try to reduce the number of I/Os.

David Clancy
Circle Computer Group (UK)

How to prevent CICS from opening VSAM files with wrong or inappropriate define parameters

INTRODUCTION

If CICS opens a VSAM file and detects that one or more parameters are wrong or inappropriate for teleprocessing, it sends a warning message and continues opening the file. An example warning message might be:

DFHFC0970 applid Warning. Recoverable file "filename", Opened with VSAM SHAROPT 3 or 4. "CICS cannot ensure integrity"

In this case, while opening a recoverable VSAM file for update, CICS detected that it was defined with SHAREOPTION 3 or 4 – which allows updating from multiple regions. CICS issues this message to warn you that it 'cannot ensure data integrity'.

The CICS manual recommends:

System action: CICS opens file filename and continues processing.

User response: If this integrity exposure is acceptable, no further user action is required. If this integrity exposure is unplanned and unacceptable, "cancel CICS", redefine the file filename with a different SHAREOPTION, and restart.

THE PROBLEM

Risking data integrity in mission-critical applications is unacceptable(though it may be acceptable in a test CICS environment or for a particular application), and even more unacceptable is the recommendation to terminate the CICS partition.

THE SOLUTION

We wanted to limit damage to a minimum. To that end, we needed a solution that automatically closes and disables all VSAM files and

would produce warning messages when opened. It is essential for the file to be correctly redefined, but the application, even if temporarily degraded, must continue to be active. It is necessary, therefore, for the file to be closed and disabled so that we can make the necessary operation corrections and then put them on-line to CICS. I have written an XFCSREQC file exit routine, which intercepts the warning messages from CICS when the file is opened. These messages are processed on the basis of an 'applid CICS' exclusion table and a 'filename' exclusion table. The exit writes a temporary storage queue, called Qname XFCSDISA, containing the filename to be closed and disabled. In addition to the XFCSDISA program, the PLT is started by the XFCSDPLT program via START TRANSID XFDS. It reads the temporary storage queues written by the exit, and closes and disables the file. A notification message is sent to the Transient Data Queue, CSMT.

XFCSREQC

```
*===============================================================
* EXIT PROGRAM XFCSREQC SOURCE CODE
*===============================================================
*ASM XOPTS(NOEDF)
*ASM XOPTS(NOEPILOG)
         TITLE  'MACRO DEFINITIONS'
         MACRO                       MACRO HEADER
         PGMID &MEMBER,&R=           PROTYPE STATEMENT
         AGO   .PGNAME
.PGNAME  ANOP
.*
.*       THIS VARIABLE FOR TIME AND DATE STAMPING
         LCLC  &VMTMDT               TIME/DATE STAMP
         LCLC  &RELEASE              VERSION
.*
.*
        AIF  (T'&R NE 'O').SETR
&RELEASE     SETC 'Ø1Ø1'
        AGO  .DROP
.SETR   ANOP
&RELEASE     SETC '&R'
         SPACE 1
.DROP   ANOP
         PUSH PRINT
         PRINT GEN
*********************************************************************
```

```
         DC    C'*',C' '
         DC    C'PROGRAM NAME:'
         DC    CL8'&MEMBER'  NAME
         DC    C' ',C'*',C' '
         DC    C'PROGRAM VERSION:'
         DC    CL4'&RELEASE'
         DC    C' '
         DC    C'*',C' '
         SPACE
         DC    C'ASSEMBLY TIME(HH.MM):'
&VMTMDT  SETC  '&SYSTIME'
         DC    C'&VMTMDT'          ASSEMBLY TIME (HH.MM) AND
         DC    C' '
         DC    C'ASSEMBLY DATE(MM/DD/YY):'
&VMTMDT  SETC  '&SYSDATE'
         DC    C'&VMTMDT'          DATE (MM/DD/YY) SAME AS LISTING
***********************************************************************
         POP   PRINT
         MEXIT
         MEND
* CSNAME MACRO
         MACRO
         CSNAME &NAME
         GBLC  &CSECT
         AIF ('&NAME' EQ '').NONAME
&CSECT   SETC  '&NAME'
         AGO   .SC
.NONAME  ANOP
&CSECT   SETC  '&SYSECT'
.SC      ANOP
         PUSH  PRINT
         PRINT GEN
*=======================================*
CSNAME   DC    CL8'&CSECT'
*=======================================*
         POP   PRINT
         MEND
*
* FCP EXIT:AFTER A FILE ENABLE/DISABLE OPEN/CLOSE COMMAND HAS COMPLETED
*
* REGISTER DETAIL:
* R1  ADDRESS OF EXIT PARAMETER-LIST DFHUEPAR
* R13 ADDRESS OF STD REGISTER SAVE AREA (ALSO IN FIELD UEPEPSA)
* R14 CONTAIN THE ADDRESS OF RETURN POINT
* R15 ENTRY ADDRESS OF EXIT PROGRAM JUST ENTERED
*
XFCSREQP DFHUEXIT TYPE=EP,ID=XFCSREQC
*
         COPY  DFHUEFDS
*
```

```
DFHEISTG DSECT
*
SAVER14  DS    A
SAVER15  DS    A
STATUS   DS    F
TABLE    DS    F
CRESP    DS    F
APPLID   DS    CL8
*
FILENAME DS    CL8
QNAME    DS    CL8
LTS      DS    H
ITEM     DS    H
*
DSNAME   DS    CL44   DATASET NAME
*
MSG      DS    CL12Ø MESSAGE AREA
*
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
RUEPAR   EQU   R7
EIBREG   EQU   R8
CODEREG  EQU   R9
CODEREG2 EQU   R1Ø
DATAREG  EQU   R12
*
RWKR1    EQU   R1
RWKR2    EQU   R2
RWKR3    EQU   R3
RWKR14   EQU   R14
RWKR15   EQU   R15
*
         USING DFHUEPAR,RUEPAR
*
XFCSREQP DFHEIENT CODEREG=(CODEREG),                                   *
               DATAREG=(DATAREG),                                      *
```

```
               EIBREG=(EIBREG)
XFCSREQP AMODE ANY
XFCSREQP RMODE ANY
         B     ACXID
         PGMID XFCSREQP,R=ØØØ1
ACXID    DS    ØH
         ST    RWKR14,SAVER14
         XC    FILENAME,FILENAME
         MVC   QNAME,=CL8'XFCSDISA'
         MVC   MSG(L'MSGW),MSGW
         MVC   MSG+39(6),=CL6'NA'
         LR    RUEPAR,R1 SAVE PARAMETER LIST ADDRESS
*
         EXEC  CICS ADDRESS EIB(EIBREG)
*
         EXEC  CICS ASSIGN APPLID(APPLID)
*
* If CICS Not ACTIVE : No Action
*
         EXEC  CICS INQUIRE SYSTEM CICSSTATUS(STATUS) NOHANDLE
*
         CLC   STATUS,DFHVALUE(ACTIVE) SYSTEM ACTIVE ?
         BNE   RETURN
         CLC   STATUS,DFHVALUE(STARTUP) KICKED OFF FROM PLT ?
         BE    RETURN
         ICM   RWKR1,B'1111',UEPFILE Valid Filename ?
         BZ    RETURN                ...No return
         MVC   MSG+Ø5(8),Ø(RWKR1) Move Filename in Message
         MVC   FILENAME,Ø(RWKR1)  Initialize field with filename
         ICM   RWKR1,B'1111',UEPFSREQ Valid request ?
         BZ    RETURN                 ...No return
* If Not OPEN : No Action
         CLI   Ø(RWKR1),UEPFSOPN OPEN ?
         BE    OKOPEN
         MVC   MSG+39(6),=CL6'Close'
         CLI   Ø(RWKR1),UEPFSCLS CLOSE ?
         BE    RETURN
         MVC   MSG+39(6),=CL6'Enable'
         CLI   Ø(RWKR1),UEPFSENB ENABLE ?
         BE    RETURN
         MVC   MSG+39(6),=CL6'Dsable'
         CLI   Ø(RWKR1),UEPFSDIS DISABLE ?
         BE    RETURN
         MVC   MSG+39(6),=CL6'NA'
         B     RETURN
OKOPEN   DS    ØH
* If Not Info area passed : No Action
         MVC   MSG+39(6),=CL6'NoInfo'
         ICM   RWKR1,B'1111',UEPFINFO INFO AREA ?
         BZ    RETURN                 ... NO
```

```
         USING DFHUEFDS,RWKR1
         MVC   DSNAME,UEDSNAME  save  dataset name
* If Not VSAM file : No Action
         MVC   MSG+39(6),=CL6'NA'
         TM    UEFDSACC,UEFVSAM VSAM FILE ?
         BZ    RETURN           ...No
         MVC   MSG+39(6),=CL6'No R.C.'
* If Not Return Codes: No Action
         ICM   RWKR1,B'1111',UEPFSRSP RETURN CODES ?
         BZ    RETURN
* If Not Warning R.C.: No Action
         MVC   MSG+39(6),=CL6'Normal'
         CLI   Ø(RWKR1),UEFSNORM Normal Response ?
         BE    RETURNOK
         MVC   MSG+39(6),=CL6'Fail'
         CLI   Ø(RWKR1),UEFSFAIL Fail Response ?
         BE    RETURN
         MVC   MSG+39(6),=CL6'Pending'
         CLI   Ø(RWKR1),UEFSPEND Pending Response ?
         BE    RETURN
         MVC   MSG+39(7),=CL7'Warning'
         CLI   Ø(RWKR1),UEFSWARN WARNING Response ?
         BE    PURGE
         MVC   MSG+39(6),=CL6' '
         UNPK  MSG+39(3),Ø(2,RWKR1)
         TR    MSG+39(2),TABEX-24Ø
         MVI   MSG+39+2,C' '
         B     RETURN
PURGE    DS    ØH
* Select CICS applid where exit is applicable
* If CICS in Table: No Action
         LA    RWKR15,TABCICS
CICSLOOP DS    ØH
         CLI   Ø(RWKR15),X'FF' End of Table ?
         BE    FLOOPC
         CLC   Ø(8,RWKR15),APPLID
         BE    RETURN             Bypass Purge
         LA    RWKR15,L'TABCICS(RWKR15)
         B     CICSLOOP
FLOOPC   DS    ØH
* If File in Table: No Action
         LA    RWKR15,TBFILEX
         ICM   RWKR1,B'1111',UEPFILE Filename address
LOOPF    DS    ØH
         CLI   Ø(RWKR15),X'FF' end of table ?
         BE    PURGEF
         CLC   Ø(8,RWKR15),Ø(RWKR1)
         BE    RETURN          Bypass Purge
         LA    RWKR15,L'TBFILEX(RWKR15)
         B     LOOPF
```

```
PURGEF   DS    ØH
*
         EXEC  CICS INQUIRE FILE(FILENAME) TABLE(TABLE) RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NORMAL)
         BNE   RETURN
         CLC   TABLE,DFHVALUE(CICSTABLE)
         BE    RETURN
         CLC   TABLE,DFHVALUE(USERTABLE)
         BE    RETURN
*
* Open refused
*
         MVC   MSG+L'MSGW+1(L'MSGR),MSGR
         MVC   MSG+L'MSGW+1+L'MSGR+1(L'DSNAME),DSNAME
*
         LA    RWKR15,UERCNORM Continue Processing
*        LA    RWKR15,UERCPURG Task Purged during XPI Call
         ST    RWKR15,SAVER15
         CLI   FILENAME,X'Ø'
         BE    RETURN
*
         EXEC  CICS INQUIRE TSQUEUE(QNAME) RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NORMAL)
         BNE   NEWTS
         XC    ITEM,ITEM
LOOPRQS  DS    ØH
         LH    RWKR1,ITEM
         LA    RWKR1,1(RWKR1)
         STH   RWKR1,ITEM
         MVC   LTS,=Y(L'FILENAME)
*
         EXEC  CICS READQ TS QUEUE(QNAME) SET(RWKR3)                  *
               LENGTH(LTS) ITEM(ITEM) RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NORMAL)
         BNE   NEWTS
         CLC   FILENAME,Ø(RWKR3)
         BNE   LOOPRQS
         B     RETURNØ
NEWTS    DS    ØH
         MVC   LTS,=Y(L'FILENAME)
*
         EXEC  CICS WRITEQ TS QUEUE(QNAME) FROM(FILENAME)             *
               LENGTH(LTS) AUXILIARY RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NORMAL)
         BNE   ERROR
         B     RETURNØ
```

```
RETURNOK DS    ØH
         CLI   FILENAME,X'Ø'
         BE    RETURN
         XC    ITEM,ITEM
LOOPRQ   DS    ØH
         LH    RWKR1,ITEM
         LA    RWKR1,1(RWKR1)
         STH   RWKR1,ITEM
         MVC   LTS,=Y(L'FILENAME)
*
         EXEC  CICS READQ TS QUEUE(QNAME) SET(RWKR3)                  *
               LENGTH(LTS) ITEM(ITEM) RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NORMAL)
         BNE   RETURN
         CLC   FILENAME,Ø(RWKR3)
         BNE   LOOPRQ
         MVC   Ø(L'FILENAME,RWKR3),=8X'FF'
*
         EXEC  CICS WRITEQ TS QUEUE(QNAME) FROM(Ø(RWKR3))             *
               LENGTH(LTS) ITEM(ITEM) REWRITE RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NORMAL)
         BNE   ERROR
RETURN   DS    ØH
*
         LA    RWKR15,UERCNORM Continue Processing
         ST    RWKR15,SAVER15
RETURNØ  DS    ØH
*
         EXEC  CICS WRITEQ TD QUEUE('CSMT') FROM(MSG)                 *
               LENGTH(=Y(L'MSG)) NOHANDLE
*
* Restore registers
*
         L     RWKR14,SAVER14
         L     RWKR15,SAVER15
         L     R13,UEPEPSA
*
         DFHEIRET RCREG=15
*
ERROR    DS    ØH CICS ERROR HANDLER ROUTINE
*
         EXEC  CICS IGNORE CONDITION ERROR
*
         EXEC  CICS WRITEQ TD QUEUE('CSMT') FROM(MSGE)                *
               LENGTH(=Y(L'MSGE)) NOHANDLE
*
         B     RETURN
*
```

```
         LTORG
MSGW     DC    CL46'File:xxxxxxxx - Vsam Open Return code: ?'
MSGR     DC    CL13'Open rejected'
MSGE     DC    CL18'<<< CICS error >>>'
TABEX    DC    C'Ø123456789ABCDEF'
         SPACE 5
* CICS applid Exclude Table
TABCICS  DS    ØCL8
         DC    CL8'CICS1'
         DC    CL8'CICS2'
         SPACE
         DC    X'FF'
*
         SPACE 5
* Filename Exclude Table
* Table of files with SHR <> (2,3) , (1,3).. tolerated
TBFILEX  DS    ØCL8
         SPACE
         DC    CL8'filename1'
         DC    CL8'filename2'
         SPACE
         DC    X'FF'
*
         CSNAME
*
         END   XFCSREQP
```

XFCSDISA

```
*====================================================================
* XFCSDISA PROGRAM SOURCE CODE
*====================================================================
         TITLE  'MACRO DEFINITIONS'
         MACRO                       MACRO HEADER
         PGMID &MEMBER,&R=           PROTYPE STATEMENT
         AGO   .PGNAME
.PGNAME  ANOP
.*
.*       THIS VARIABLE FOR TIME AND DATE STAMPING
         LCLC  &VMTMDT               TIME/DATE STAMP
         LCLC  &RELEASE              VERSION
.*
.*
        AIF  (T'&R NE 'O').SETR
&RELEASE     SETC 'Ø1Ø1'
        AGO  .DROP
.SETR   ANOP
&RELEASE     SETC '&R'
         SPACE 1
```

```
.DROP   ANOP
         PUSH PRINT
         PRINT GEN
*************************************************************************
         DC    C'*',C' '
         DC    C'PROGRAM NAME:'
         DC    CL8'&MEMBER'  NAME
         DC    C' ',C'*',C' '
         DC    C'PROGRAM VERSION:'
         DC    CL4'&RELEASE'
         DC    C' '
         DC    C'*',C' '
         SPACE
         DC    C'ASSEMBLY TIME(HH.MM):'
&VMTMDT  SETC  '&SYSTIME'
         DC    C'&VMTMDT'          ASSEMBLY TIME (HH.MM) AND
         DC    C' '
         DC    C'ASSEMBLY DATE(MM/DD/YY):'
&VMTMDT  SETC  '&SYSDATE'
         DC    C'&VMTMDT'          DATE (MM/DD/YY) SAME AS LISTING
*************************************************************************
         POP   PRINT
         MEXIT
         MEND
*--------------------------------------
         MACRO
*
*        PROTOTYPE STATEMENT
         CSNAME &NAME
         GBLC  &CSECT
         AIF ('&NAME' EQ '').NONAME
&CSECT   SETC  '&NAME'
         AGO   .SC
.NONAME  ANOP
&CSECT   SETC  '&SYSECT'
.SC      ANOP
         PUSH  PRINT
         PRINT GEN
*=======================================*
CSNAME   DC    CL8'&CSECT'
*=======================================*
         POP   PRINT
         MEND
         TITLE 'Program XFCSDISA'
         DFHCOVER
DFHEISTG DSECT                 DEFINE DYNAMIC STORAGE
*
*****************************************************************
CRESP    DS    F    CICS RESPONSE
OSTATUS  DS    F    OPEN STATUS
```

```
ESTATUS  DS    F    ENABLE STATUS
VOXBAL1  DS    A    RETURN ADDRESS
VOXBAL2  DS    A    RETURN ADDRESS
STD      DS    CL2  START CODE
APPLID   DS    CL8  APPLID NAME
*
REQNAME  DS    ØCL8
QNAME    DS    ØCL8 TS QUEUE NAME (THE SAME NAME OF REQNAME)
         DS    CL8
*
ITEM     DS    H    TS ITEM
LTSIOA   DS    H    TSIOA LENGTH
TSIOA    DS    CL8  TS I/O AREA
MSG      DS    CL8Ø MESSAGE TO CSMT QUEUE
*
*****************************************************************
         SPACE 5
*
***   REGISTER EQUATES
*
         SPACE
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
         SPACE
RWKR1    EQU   R1  WORK REGISTER 1
RWKR2    EQU   R2  WORK REGISTER 2
RWKR3    EQU   R3  WORK REGISTER 3
RWKR14   EQU   R14 WORK REGISTER 14
RWKR15   EQU   R15 WORK REGISTER 15
RBAL1    EQU   R1  1ST LEVEL ROUTINES
RBAL2    EQU   R2  2ND LEVEL ROUTINES
CODEREG1 EQU   R4  1ST BASE REGISTER
DATAREG1 EQU   R9  DFHEISTG BASE REGISTER
EIBREG   EQU   R1Ø EIB BASE REGISTER
         SPACE
*
```

```
*** SOURCE CODE BEGIN
*
         SPACE 5
         PRINT NOGEN
XFCSDISA DFHEIENT CODEREG=(CODEREG1),                                  *
               DATAREG=(DATAREG1),                                     *
               EIBREG=(EIBREG)
XFCSDISA AMODE ANY
XFCSDISA RMODE ANY
         B     ACXID                   BRANCH AROUND PROGRAM IDENTIFIER
         PGMID XFCSDISA,R=ØØØ1
ACXID    DS    ØH
*
* FIELDS INITIALIZATION
*
         MVC   QNAME,CSNAME             SET QUEUE NAME
*
         MVC   LTSIOA,=Y(L'TSIOA)
*
* HANDLE ABEND CONDITION
*
         EXEC  CICS HANDLE ABEND LABEL(ABEND)
*
* HANDLE UNPREDICTABLE CICS ERRORS
*
         EXEC  CICS HANDLE CONDITION ERROR(ERROR)
*
         BAS   RBAL1,GETCI                 GET CICS INFO
         XC    ITEM,ITEM                   TEMPORARY STORAGE ITEM
LOOPRQ   DS    ØH
         LH    RWKR1,ITEM                  ADD 1 TO ITEM
         LA    RWKR1,1(RWKR1)
         STH   RWKR1,ITEM
         BAS   RBAL1,READQ                 GET TS QUEUE
         CLC   CRESP,DFHRESP(NORMAL)       READ TS QUEUE OK ?
         BNE   START
         CLC   TSIOA,=8X'FF'
         BE    LOOPRQ
         BAS   RBAL1,ANALIZE               ANALYSE FILE
         B     LOOPRQ
*
* RESTART TASK AFTER 1 MINUTE
*
START    DS    ØH
*
         EXEC  CICS START TRANSID(EIBTRNID) INTERVAL(INTERVAL)         *
               REQID(REQNAME)
*
         B     RETURN                       RETURN
ANALIZE  DS    ØH
```

```
         ST    RWKR1,VOXBAL1                SAVE RETURN ADDRESS
*
         EXEC  CICS INQUIRE FILE(TSIOA) OPENSTATUS(OSTATUS)           *
               ENABLESTATUS(ESTATUS) RESP(CRESP)
*
         CLC   CRESP,DFHRESP(NORMAL)
         BNE   ANALIZEE
         CLC   ESTATUS,DFHVALUE(ENABLED)
         BE    ANALIZC
         CLC   OSTATUS,DFHVALUE(OPEN)
         BNE   ANALIZEE
ANALIZC  DS    ØH
*
         EXEC  CICS SET FILE(TSIOA) CLOSED DISABLED
*
ANALIZEE DS    ØH
         L     RWKR1,VOXBAL1                LOAD RETURN ADDRESS
         BR    RWKR1                        RETURN
         TITLE 'RETURN TO CICS'
RETURN   DS    ØH
*                                           SEND START CONFIRMATION
         EXEC  CICS RETURN
*
         TITLE 'ABEND ROUTINE'
*
* IF ABENDED RESTART TASK
*
ABEND    DS    ØH
*
         EXEC  CICS START TRANSID(EIBTRNID) INTERVAL(INTERVAL)        *
               REQID(REQNAME) NOHANDLE
*
         B     RETURN
GETCI    DS    ØH
         ST    RBAL1,VOXBAL1
*
* DELETES EXISTING ICE TO AVOID DOUBLE START
*
         EXEC  CICS CANCEL REQID(REQNAME) NOHANDLE
*
* GETS STARTING CODE AND APPLID NAME
*
         EXEC  CICS ASSIGN STARTCODE(STD) APPLID(APPLID)
*
         L     RBAL1,VOXBAL1
         BR    RBAL1
READQ    DS    ØH
         ST    RBAL1,VOXBAL1
         MVC   LTSIOA,=Y(L'TSIOA)          DATA LENGTH
*                                          ERROR
```

```
         EXEC  CICS READQ TS QUEUE(QNAME) ITEM(ITEM) RESP(CRESP)       *
               INTO(TSIOA) LENGTH(LTSIOA)
*
         L     RBAL1,VOXBAL1
         BR    RBAL1
TDWRITE  DS    ØH
         ST    RBAL2,VOXBAL2
*
         EXEC  CICS WRITEQ TD QUEUE('CSMT') FROM(MSG)                  *
               LENGTH(=Y(L'MSG))
*
         MVI   MSG,C' '
         MVC   MSG+1(L'MSG-1),MSG           RESET MSG AREA
         L     RBAL2,VOXBAL2
         BR    RBAL2
         TITLE 'ERROR ROUTINE'
ERROR    DS    ØH
*
* DISABLE HANDLE CONDITION ERROR TO AVOID ANY LOOP
*
         EXEC  CICS IGNORE CONDITION ERROR
*
         SPACE
*
         EXEC  CICS WRITEQ TD QUEUE('CSMT') FROM(MSGE)                 *
               LENGTH(=Y(L'MSGE))
*
         B     ABEND      RESTART TASK
         TITLE 'LITERALS && CONSTANTS'
         LTORG
         SPACE
* CSNAME FIELD CONTAIN CONTROL SECTION NAME
         CSNAME
         SPACE
MSGE     DC    CL18'<<< CICS Error >>>'
         TITLE 'DEFAULT PARAMETERS'
*
* DEFAULT RESTART INTERVAL
*
INTERVAL DC    XL4'ØØØØØ3ØC' DEFAULT RESTART INTERVAL
*                   ØHHMMSS+
         SPACE
         END   XFCSDISA
```

XFCSDPLT

```
*===================================================================
* XFCSDPLT PROGRAM SOURCE CODE
*===================================================================
```

```
         TITLE  'MACRO DEFINITIONS'
         MACRO                       MACRO HEADER
         PGMID &MEMBER,&R=           PROTYPE STATEMENT
         AGO   .PGNAME
.PGNAME  ANOP
.*
.*       THIS VARIABLE FOR TIME AND DATE STAMPING
         LCLC  &VMTMDT               TIME/DATE STAMP
         LCLC  &RELEASE              VERSION
.*
.*
        AIF  (T'&R NE 'O').SETR
&RELEASE     SETC 'Ø1Ø1'
        AGO  .DROP
.SETR   ANOP
&RELEASE     SETC '&R'
         SPACE 1
.DROP   ANOP
         PUSH PRINT
         PRINT GEN
*********************************************************************
         DC    C'*',C' '
         DC    C'PROGRAM NAME:'
         DC    CL8'&MEMBER'  NAME
         DC    C' ',C'*',C' '
         DC    C'PROGRAM VERSION:'
         DC    CL4'&RELEASE'
         DC    C' '
         DC    C'*',C' '
         SPACE
         DC    C'ASSEMBLY TIME(HH.MM):'
&VMTMDT  SETC  '&SYSTIME'
         DC    C'&VMTMDT'          ASSEMBLY TIME (HH.MM) AND
         DC    C' '
         DC    C'ASSEMBLY DATE(MM/DD/YY):'
&VMTMDT  SETC  '&SYSDATE'
         DC    C'&VMTMDT'          DATE (MM/DD/YY) SAME AS LISTING
*********************************************************************
         POP   PRINT
         MEXIT
         MEND
*————————————————————————————————
         MACRO
*
*        PROTOTYPE STATEMENT
         CSNAME &NAME
         GBLC  &CSECT
         AIF ('&NAME' EQ '').NONAME
&CSECT   SETC  '&NAME'
         AGO   .SC
.NONAME  ANOP
```

```
&CSECT   SETC  '&SYSECT'
.SC      ANOP
         PUSH  PRINT
         PRINT GEN
*=======================================*
CSNAME   DC    CL8'&CSECT'
*=======================================*
         POP   PRINT
         MEND
         TITLE 'Program XFCSDPLT'
         DFHCOVER
DFHEISTG DSECT                 DEFINE DYNAMIC STORAGE
*
*****************************************************************
*
WRESP    DS    F
REQNAME  DS    CL8
STRAN    DS    CL4
MSG      DS    CL8Ø
*
*****************************************************************
         SPACE 5
*
***   REGISTER EQUATES
*
         SPACE
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
         SPACE
RWKR1    EQU   R1  WORK REGISTER 1
RWKR2    EQU   R2  WORK REGISTER 2
RWKR3    EQU   R3  WORK REGISTER 3
RWKR14   EQU   R14 WORK REGISTER 14
RWKR15   EQU   R15 WORK REGISTER 15
RBAL1    EQU   R1  1ST LEVEL ROUTINES
RBAL2    EQU   R2  2ND LEVEL ROUTINES
CODEREG1 EQU   R4  1ST BASE REGISTER
```

```
DATAREG1 EQU   R9  DFHEISTG BASE REGISTER
EIBREG   EQU   R1Ø EIB BASE REGISTER
         SPACE
*
*** SOURCE CODE BEGIN
*
         SPACE 5
         PRINT NOGEN
XFCSDPLT DFHEIENT CODEREG=(CODEREG1),                                  *
               DATAREG=(DATAREG1),                                     *
               EIBREG=(EIBREG)
XFCSDPLT AMODE ANY
XFCSDPLT RMODE ANY
         B     ACXID                   BRANCH AROUND PROGRAM IDENTIFIER
         PGMID XFCSDPLT,R=ØØØ1
ACXID    DS    ØH
         MVC   STRAN,TRANSID
         MVC   REQNAME,=CL8'XFCSDISA'
*
* DELETES EXISTING ICE TO AVOID DOUBLE START
*
         EXEC  CICS CANCEL REQID(REQNAME) NOHANDLE
*
         EXEC  CICS START TRANSID(STRAN) INTERVAL(Ø)                    *
               REQID(REQNAME) RESP(WRESP)
*
         CLC   WRESP,DFHRESP(NORMAL)
         BE    OKSTART
         MVC   MSG,MSGE
         MVC   MSG+12(4),STRAN
         B     RETURN
OKSTART  DS    ØH
         MVC   MSG,MSGW
         MVC   MSG+12(4),STRAN
RETURN   DS    ØH
*
         EXEC  CICS RETURN
*
MSGW     DC    CL8Ø'Transaction xxxx successfully started'
MSGE     DC    CL8Ø'Transaction xxxx NOT started !!!!'
*
* DEFAULT TRANSID
*
TRANSID  DC    CL4'XFDS'
         CSNAME
         END   XFCSDPLT
```

Giuseppe Rallo
Senior Technical Analyst
Consulent (Italy)

Getting rid of null-use resources – part 2

This month we conclude the code used to identify and remove obsolete CSD resources.

```
*****************************************************************
 LINKAGE SECTION.
*****************************************************************

*****************************************************************
 PROCEDURE DIVISION.
*****************************************************************

*******************
 0000-MAIN SECTION.
*******************
     PERFORM P-INITIALISE.
     PERFORM P-PROCESS.
     PERFORM P-UPDATE-CONTROL-FILE-RECORD.
     PERFORM P-DISPLAY-TOTALS.
     PERFORM P-CLEANUP.
 0000-RETURN.
     MOVE W-RETURN-CODE-PIC TO RETURN-CODE.
     GOBACK.
 0000-EXIT.
     EXIT.

*****************************************************************
 P-DISPLAY-TOTALS.
*****************************************************************
     MOVE W-RECORD-COUNT TO W-WORK-PIC.
     DISPLAY ' '.
     DISPLAY 'TOTALS'.
     DISPLAY '******'.
     DISPLAY 'RECORDS PROCESSED = ' W-WORK-PIC.
     DISPLAY ' '.
     MOVE W-NULLUSE-PROG TO         W-WORK-PIC.
     DISPLAY 'NULLUSE PROGRAMS        = ' W-WORK-PIC.
     MOVE W-NULLUSE-TRAN TO         W-WORK-PIC.
     DISPLAY 'NULLUSE TRANSACTIONS    = ' W-WORK-PIC.
     MOVE W-NULLUSE-FILE TO         W-WORK-PIC.
     DISPLAY 'NULLUSE FILES           = ' W-WORK-PIC.
     ADD W-NULLUSE-FILE, W-NULLUSE-PROG, W-NULLUSE-TRAN
        GIVING W-WORK-PIC.
     DISPLAY 'TOTAL NULLUSE RESOURCES = ' W-WORK-PIC.
     DISPLAY ' '.
     MOVE W-NULLUSE-NEW TO          W-WORK-PIC.
     DISPLAY 'NEW RECORDS             = ' W-WORK-PIC.
```

```
     MOVE W-NULLUSE-OLD TO          W-WORK-PIC.
     DISPLAY 'UPDATED RECORDS         = ' W-WORK-PIC.
     ADD W-NULLUSE-NEW, W-NULLUSE-OLD GIVING W-WORK-PIC.
     DISPLAY 'TOTAL NEW + UPDATED     = ' W-WORK-PIC.

*****************************************************************
 P-PROCESS.
*****************************************************************
 PERFORM WITH TEST AFTER UNTIL W-SORTFILE-STATUS NOT = '00'
    READ SORTFILE
    IF W-SORTFILE-STATUS = '00' THEN
       ADD +1 TO W-RECORD-COUNT
       IF SW-VALID-RESOURCE THEN
          IF F-SORTFILE-READ-BUFFER = W-PREVIOUS-READ-BUFFER
             PERFORM P-BUMP-DUPLICATE-COUNT
          ELSE
            MOVE F-SORTFILE-READ-BUFFER TO W-PREVIOUS-READ-BUFFER
            MOVE +1 TO W-RESOURCE-COUNT
          END-IF
       END-IF
    END-IF
 END-PERFORM.

*****************************************************************
 P-BUMP-DUPLICATE-COUNT.
*****************************************************************
     ADD +1 TO W-RESOURCE-COUNT.
**** UPDATE NULLFILE IF RESOURCE NOT USED IN ANY GENERATION
     IF W-RESOURCE-COUNT >= W-NO-GENS THEN
        PERFORM P-UPDATE-NULLFILE
     END-IF.

*****************************************************************
 P-UPDATE-CONTROL-FILE-RECORD.
*****************************************************************
**** SET UP NULLFILE KEY (LOW-VALUES)
     MOVE LOW-VALUES TO F-NULLFILE-KEY.
**** OPEN NULLFILE I-O IF IT IS IN OPEN EXTEND MODE
     IF NOT SW-NULLFILE-OPEN-IO THEN
        CLOSE NULLFILE
        PERFORM P-OPEN-NULLFILE
        IF NOT SW-NULLFILE-OPEN-IO THEN
           MOVE 'NULLFILE OPEN LOGIC ERROR' TO W-ERROR-MSG
           PERFORM P-VSAM-ERROR
           GO TO 0000-RETURN
        END-IF
     END-IF.
     READ NULLFILE
        KEY IS F-NULLFILE-KEY
        INVALID KEY MOVE 08 TO W-VSAM-RETURN-CODE
     END-READ.
```

```
      EVALUATE TRUE
         WHEN W-VSAM-RETURN-CODE = Ø8
            PERFORM P-ADD-RECORD
         WHEN W-VSAM-RETURN-CODE = ØØ OR LOW-VALUES
            PERFORM P-UPDATE-RECORD
         WHEN OTHER
            MOVE 'NULLFILE READ ERROR' TO W-ERROR-MSG
            PERFORM P-VSAM-ERROR
            GO TO ØØØØ-RETURN
      END-EVALUATE.
*****************************************************************
 P-UPDATE-NULLFILE.
*****************************************************************
**** SET UP NULLFILE KEY
      MOVE LOW-VALUES            TO F-NULLFILE-KEY.
      MOVE F-SORTFILE-RESOURCE-ID TO F-NULLFILE-RESOURCE-TYPE.
      MOVE F-SORTFILE-RESOURCE-NAME TO F-NULLFILE-RESOURCE-NAME.
      IF SW-NULLFILE-OPEN-OUTPUT THEN
         PERFORM P-ADD-RECORD
      ELSE
         READ NULLFILE
            KEY IS F-NULLFILE-KEY
            INVALID KEY MOVE Ø8 TO W-VSAM-RETURN-CODE
         END-READ
         EVALUATE TRUE
            WHEN W-VSAM-RETURN-CODE = Ø8
               PERFORM P-ADD-RECORD
            WHEN W-VSAM-RETURN-CODE = ØØ OR LOW-VALUES
               PERFORM P-UPDATE-RECORD
            WHEN OTHER
               MOVE 'NULLFILE READ ERROR' TO W-ERROR-MSG
               PERFORM P-VSAM-ERROR
               GO TO ØØØØ-RETURN
         END-EVALUATE
      END-IF.
      EVALUATE F-SORTFILE-RESOURCE-ID
         WHEN 'PROG' ADD +1 TO W-NULLUSE-PROG
         WHEN 'FILE' ADD +1 TO W-NULLUSE-FILE
         WHEN 'TRAN' ADD +1 TO W-NULLUSE-TRAN
         WHEN OTHER
            MOVE +2Ø TO W-RETURN-CODE-PIC
            DISPLAY 'RESOURCE LOGIC ERROR 1'
            GO TO ØØØØ-RETURN
      END-EVALUATE.
*****************************************************************
 P-UPDATE-RECORD.
*****************************************************************
      ADD +1 TO F-NULLFILE-NULLUSE-COUNT.
      IF F-PARMFILE-LAST-DATE > F-NULLFILE-LAST-DATE OR
         (F-NULLFILE-LAST-DATE = LOW-VALUES OR SPACES) THEN
         MOVE F-PARMFILE-LAST-DATE TO F-NULLFILE-LAST-DATE
```

```
           END-IF.
           IF F-NULLFILE-START-DATE = LOW-VALUES OR SPACES THEN
              MOVE F-PARMFILE-START-DATE TO F-NULLFILE-START-DATE
           END-IF.
           REWRITE F-NULLFILE-BUFFER.
           IF W-VSAM-RETURN-CODE NOT = ØØ AND
              W-VSAM-RETURN-CODE NOT = LOW-VALUES THEN
              MOVE 'NULLFILE REWRITE ERROR' TO W-ERROR-MSG
              PERFORM P-VSAM-ERROR
              GO TO ØØØØ-RETURN
           END-IF.
           IF F-NULLFILE-KEY NOT = LOW-VALUES THEN
              ADD +1 TO W-NULLUSE-OLD
           END-IF.
      ****************************************************************
       P-CHECK-DATES.
      ****************************************************************
           IF F-PARMFILE-LAST-DATE = SPACES OR LOW-VALUES THEN
              MOVE +2Ø TO W-RETURN-CODE-PIC
              DISPLAY 'LAST DATE PARM MISSING'
              GO TO ØØØØ-RETURN
           END-IF.
           IF F-PARMFILE-START-DATE = SPACES OR LOW-VALUES THEN
              MOVE +2Ø TO W-RETURN-CODE-PIC
              DISPLAY 'START DATE PARM MISSING'
              GO TO ØØØØ-RETURN
           END-IF.
      ****************************************************************
       P-ADD-RECORD.
      ****************************************************************
           MOVE SPACES TO F-NULLFILE-DATA.
           MOVE +1 TO F-NULLFILE-NULLUSE-COUNT.
           MOVE F-PARMFILE-LAST-DATE TO F-NULLFILE-LAST-DATE.
           MOVE F-PARMFILE-START-DATE TO F-NULLFILE-START-DATE.
           WRITE F-NULLFILE-BUFFER.
           IF W-VSAM-RETURN-CODE NOT = ØØ AND
              W-VSAM-RETURN-CODE NOT = LOW-VALUES THEN
              MOVE 'NULLFILE WRITE ERROR' TO W-ERROR-MSG
              PERFORM P-VSAM-ERROR
              GO TO ØØØØ-RETURN
           END-IF.
           IF F-NULLFILE-KEY NOT = LOW-VALUES THEN
              ADD +1 TO W-NULLUSE-NEW
           END-IF.
      ****************************************************************
       P-INITIALIZE.
      ****************************************************************
           PERFORM P-OPEN-WORKFILES.
           PERFORM P-OPEN-NULLFILE.
      **** PARMFILE FILE HAS THE INPUT PARAMETERS
           READ PARMFILE.
```

```
 IF W-PARMFILE-STATUS NOT = '00' THEN
    MOVE +20 TO W-RETURN-CODE-PIC
    DISPLAY 'READ OF PARMFILE FILE FAILED'
    DISPLAY 'STATUS CODE=' W-PARMFILE-STATUS
    GO TO 0000-RETURN
 END-IF.
 DISPLAY 'STATPRG1 INPUT PARMS = ' F-PARMFILE-READ-BUFFER.
 IF (F-PARMFILE-NO-GENS = '000000') OR
    (F-PARMFILE-NO-GENS NOT NUMERIC) THEN
    MOVE +20 TO W-RETURN-CODE-PIC
    DISPLAY 'NO. GENERATIONS PARM MISSING OR INCORRECT'
    DISPLAY '(' F-PARMFILE-NO-GENS ')'
    GO TO 0000-RETURN
 END-IF.
 PERFORM P-CHECK-DATES.
 DISPLAY ' '.
 DISPLAY 'NUMBER OF GENERATIONS PROCESSED = ' F-PARMFILE-NO-GENS.
 DISPLAY 'START DATE = ' F-PARMFILE-START-DATE.
 DISPLAY 'END DATE   = ' F-PARMFILE-LAST-DATE.
 DISPLAY ' '.
 MOVE F-PARMFILE-NO-GENS TO W-NO-GENS-PIC.
 MOVE W-NO-GENS-PIC   TO W-NO-GENS.
*****************************************************************
 P-OPEN-WORKFILES.
*****************************************************************
     OPEN INPUT SORTFILE.
     IF W-SORTFILE-STATUS NOT = '00' THEN
        MOVE +20 TO W-RETURN-CODE-PIC
        DISPLAY 'OPEN OF SORTFILE FILE FAILED'
        DISPLAY 'STATUS CODE=' W-SORTFILE-STATUS
        GO TO 0000-RETURN
     END-IF.
     SET SW-SORTFILE-OPEN TO TRUE.
     OPEN INPUT PARMFILE.
     IF W-PARMFILE-STATUS NOT = '00' THEN
        MOVE +20 TO W-RETURN-CODE-PIC
        DISPLAY 'OPEN OF PARMFILE FILE FAILED'
        DISPLAY 'STATUS CODE=' W-PARMFILE-STATUS
        GO TO 0000-RETURN
     END-IF.
     SET SW-PARMFILE-OPEN TO TRUE.
*****************************************************************
 P-OPEN-NULLFILE.
*****************************************************************
     OPEN I-O NULLFILE.
     EVALUATE TRUE
        WHEN W-NULLFILE-STATUS = '00'
           SET SW-NULLFILE-OPEN-IO TO TRUE
******* FILE IS EMPTY, OPEN FOR OUTPUT
        WHEN W-NULLFILE-STATUS = '35'
           OPEN OUTPUT NULLFILE
```

```
                IF W-NULLFILE-STATUS = '00' THEN
                   SET SW-NULLFILE-OPEN-OUTPUT TO TRUE
                ELSE
                   MOVE 'OPEN OF NULLFILE FAILED' TO W-ERROR-MSG
                   PERFORM P-VSAM-ERROR
                   GO TO 0000-RETURN
                END-IF
             WHEN OTHER
                MOVE 'OPEN OF NULLFILE FAILED' TO W-ERROR-MSG
                PERFORM P-VSAM-ERROR
                GO TO 0000-RETURN
          END-EVALUATE.
          SET SW-NULLFILE-OPEN TO TRUE.
     *****************************************************************
      P-CLEANUP.
     *****************************************************************
          IF SW-SORTFILE-OPEN THEN
             CLOSE SORTFILE
             SET SW-SORTFILE-CLOSED TO TRUE
          END-IF.
          IF SW-PARMFILE-OPEN THEN
             CLOSE PARMFILE
             SET SW-PARMFILE-CLOSED TO TRUE
          END-IF.
          IF SW-NULLFILE-OPEN THEN
             CLOSE NULLFILE
             SET SW-NULLFILE-CLOSED TO TRUE
          END-IF.
     *****************************************************************
      P-VSAM-ERROR.
     *****************************************************************
          MOVE W-VSAM-RETURN-CODE TO W-RETURN-CODE-PIC.
          DISPLAY W-ERROR-MSG.
          DISPLAY 'STATUS CODE      =' W-NULLFILE-STATUS.
          DISPLAY 'VSAM RETURN CODE =' W-VSAM-RETURN-CODE.
          DISPLAY 'VSAM REASON CODE =' W-VSAM-REASON-CODE.
          DISPLAY 'RECORD KEY       =' F-NULLFILE-KEY.
          PERFORM P-CLEANUP.
```

STATPRG3

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID.    STATPRG3.
     *****************************************************************
     * REPORT NULLUSE CICS RESOURCES
     *****************************************************************
     * FUNCTION : REPORT NULLUSE CICS RESOURCES ON PERMANENT
     * 'NULLUSE' FILE
     *
     *INPUT FILES  : NULLFILE (NULLUSE FILE) - VSAM KSDS
```

```
*OUTPUT FILES : (DISPLAY), REPORTT
*
*PARMS       : RESOURCE TYPE - 'FILE', 'TRAN', 'PROG' OR '*'
*              ('*' IS DEFAULT)
*
*IF THE 'LAST DATE' OF A RECORD = THE CONTROL RECORD 'LAST DATE',
*THIS RECORD IS WRITTEN TO THE REPORT.
*
*IF THE 'LAST DATE' OF A RECORD > THE CONTROL RECORD 'LAST DATE',
*THIS IS A LOGIC ERROR.
*
*IF THE 'LAST DATE' OF A RECORD < THE CONTROL RECORD 'LAST DATE',
*THIS MEANS THAT THE RESOURCE HAS BEEN USED AGAIN, AND IT IS
*THEREFORE DELETED FROM THE NULLUSE FILE.
*****************************************************************
 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT NULLFILE ASSIGN TO NULLFILE
     RECORD KEY IS F-NULLFILE-KEY
     FILE STATUS IS W-NULLFILE-STATUS W-VSAM-CODE
     ORGANIZATION IS INDEXED ACCESS IS DYNAMIC.
     SELECT REPORTT ASSIGN TO SYSØØ2-DA-339Ø-S-REPORTT
     FILE STATUS IS W-REPORTT-STATUS
     ORGANIZATION IS SEQUENTIAL ACCESS IS SEQUENTIAL.
 DATA DIVISION.
 FILE SECTION.
     COPY NULLFILE.
*
 FD REPORTT
     RECORD CONTAINS 133 CHARACTERS
     BLOCK  CONTAINS Ø RECORDS
     RECORDING MODE IS F
     LABEL RECORDS ARE STANDARD.
 Ø1 F-REPORTT-BUFFER.
    Ø2 F-PRINT-CONTROL-CHAR                    PIC X.
    Ø2 F-REPORTT-DATA                          PIC X(132).
*************************
 WORKING-STORAGE SECTION.
*************************
 Ø1  C-CONSTANTS.
     Ø2 C-EYECATCHER     VALUE '*START OF WORKING STORAGE*'
                                               PIC X(26).
     Ø2 C-PROGRAM-ID     VALUE 'STATPRG3'      PIC X(8).
     Ø2 C-VERSION        VALUE 'Ø1.ØØ'         PIC X(5).
     Ø2 C-MAX-LINE-LENGTH      VALUE +11Ø      PIC S9(4) COMP.
 Ø1  W-SWITCHES.
     Ø2 W-NULLFILE-OPEN-SWITCH     VALUE SPACE  PIC X.
        88 SW-NULLFILE-OPEN         VALUE 'O'.
        88 SW-NULLFILE-CLOSED       VALUE SPACE.
     Ø2 W-REPORTT-OPEN-SWITCH      VALUE SPACE  PIC X.
```

```
          88 SW-REPORTT-OPEN         VALUE 'O'.
          88 SW-REPORTT-CLOSED       VALUE SPACE.
 Ø1  W-WORK-FIELDS.
     Ø2 W-WORK-PIC           VALUE Ø            PIC 9(8).
     Ø2 W-VSAM-CODE.
        Ø3 W-VSAM-RETURN-CODE       VALUE ØØ    PIC 9(2).
        Ø3 W-VSAM-COMPONENT-CODE    VALUE Ø     PIC 9(1).
        Ø3 W-VSAM-REASON-CODE       VALUE ØØØ   PIC 9(3).
     Ø2 W-NULLFILE-STATUS    VALUE 'ØØ'         PIC X(2).
     Ø2 W-REPORTT-STATUS     VALUE 'ØØ'         PIC X(2).
     Ø2 W-WORK4              VALUE SPACES       PIC X(4).
     Ø2 W-WORK8              VALUE SPACES       PIC X(8).
     Ø2 W-RETURN-CODE-SAVE                      PIC S9(8) COMP.
     Ø2 W-RETURN-CODE-PIC    VALUE Ø            PIC 9(6).
     Ø2 I                                       PIC S9(8) COMP.
     Ø2 W-ERROR-MSG                             PIC X(4Ø).
     Ø2 W-RESOURCE-COUNT     VALUE +Ø           PIC S9(4) COMP.
     Ø2 W-DELETE-COUNT       VALUE +Ø           PIC S9(4) COMP.
     Ø2 W-START-DATE         VALUE SPACES       PIC X(1Ø).
     Ø2 W-LAST-DATE          VALUE SPACES       PIC X(1Ø).
     Ø2 W-RESOURCE-PARM      VALUE SPACES       PIC X(4).
        88 SW-FILE            VALUE 'FILE'.
        88 SW-PROG            VALUE 'PROG'.
        88 SW-TRAN            VALUE 'TRAN'.
        88 SW-ALL             VALUE '*   ' ' * ' '  *'.
        88 SW-VALID-RESOURCE VALUE 'TRAN' 'PROG' 'FILE' '*'.
     Ø2 W-CURRENT-RESOURCE   VALUE SPACES       PIC X(4).
        88 SW-CURRENT-FILE    VALUE 'FILE'.
        88 SW-CURRENT-PROG    VALUE 'PROG'.
        88 SW-CURRENT-TRAN    VALUE 'TRAN'.
     Ø2 W-REPORTT-DATA-PTR   VALUE +1           PIC S9(4) COMP.
*****************************************************************
 LINKAGE SECTION.
*****************************************************************
 Ø1  L-PARM-FIELDS.
     Ø2 L-PARM-LENGTH                           PIC S9(4) COMP.
     Ø2 L-PARMS                                 PIC X(2Ø).
*****************************************************************
 PROCEDURE DIVISION USING L-PARM-FIELDS.
*****************************************************************
******************
 ØØØØ-MAIN SECTION.
******************
     PERFORM P-INITIALISE.
     PERFORM P-PROCESS.
     PERFORM C-REPORT-TOTALS.
 ØØØØ-RETURN.
     PERFORM P-CLEANUP.
     MOVE W-RETURN-CODE-PIC TO RETURN-CODE.
     GOBACK.
 ØØØØ-EXIT.
```

```
     EXIT.
*****************************************************************
 P-PROCESS.
*****************************************************************
     IF NOT SW-ALL THEN
        PERFORM P-BROWSE
     ELSE
        MOVE 'FILE' TO F-NULLFILE-RESOURCE-TYPE
        SET SW-CURRENT-FILE TO TRUE
        PERFORM P-BROWSE
        MOVE LOW-VALUES TO F-NULLFILE-KEY
        MOVE 'PROG' TO F-NULLFILE-RESOURCE-TYPE
        SET SW-CURRENT-PROG TO TRUE
        PERFORM P-BROWSE
        MOVE LOW-VALUES TO F-NULLFILE-KEY
        MOVE 'TRAN' TO F-NULLFILE-RESOURCE-TYPE
        SET SW-CURRENT-TRAN TO TRUE
        PERFORM P-BROWSE
     END-IF.
*****************************************************************
 P-BROWSE.
*****************************************************************
     PERFORM P-START-BROWSE.
     MOVE SPACES TO F-REPORTT-BUFFER.
     PERFORM P-WRITE-REPORT-LINE.
     MOVE '_____________________________________________________'
        TO F-REPORTT-DATA.
     PERFORM P-WRITE-REPORT-LINE.
     STRING
        W-CURRENT-RESOURCE ' REPORT'  DELIMITED BY SIZE
        INTO F-REPORTT-DATA
     END-STRING.
     PERFORM P-WRITE-REPORT-LINE.
     STRING
        'RESOURCES NOT USED FROM ' W-START-DATE
        ' TO ' W-LAST-DATE
        DELIMITED BY SIZE INTO F-REPORTT-DATA
     END-STRING.
     MOVE '_____________________________________________________'
        TO F-REPORTT-DATA.
     PERFORM P-WRITE-REPORT-LINE.
     PERFORM UNTIL W-NULLFILE-STATUS NOT = '00'
        OR W-CURRENT-RESOURCE NOT = F-NULLFILE-RESOURCE-TYPE
        READ NULLFILE NEXT END-READ
        IF W-NULLFILE-STATUS = '00' THEN
           IF W-CURRENT-RESOURCE = F-NULLFILE-RESOURCE-TYPE
              PERFORM P-PROCESS-RECORD
           END-IF
        END-IF
     END-PERFORM.
**** AT END OF FILE OR END OF RESOURCE, WRITE OUT ANY REMAINING
```

```
**** DATA IN THE REPORT BUFFER
     IF F-REPORTT-DATA NOT = SPACES THEN
        PERFORM P-WRITE-REPORT-LINE
     END-IF.
*****************************************************************
 P-PROCESS-RECORD.
*****************************************************************
     EVALUATE TRUE
**** IF A RESOURCE LAST DATE IS > THE CONTROL RECORD LAST DATE,
******* THE CONTROL RECORD LAST DATE MUST BE CORRUPT OR INCORRECT.
        WHEN F-NULLFILE-LAST-DATE > W-LAST-DATE
           PERFORM P-DATE-LOGIC-ERROR
           GO TO 0000-RETURN
******* IF RESOURCE LAST DATE = THE CONTROL RECORD LAST DATE,
******* THE RESOURCE HAS NOT BEEN USED SINCE THE START DATE.
        WHEN F-NULLFILE-LAST-DATE = W-LAST-DATE
           PERFORM P-PROCESS-NULLUSE-RECORD
******* IF RESOURCE LAST DATE < THE CONTROL RECORD LAST DATE,
******* THE RESOURCE HAS BEEN USED SINCE THE START DATE,
******* THEREFORE WE DELETE IT FROM THE NULLUSE FILE.
        WHEN OTHER
           PERFORM P-DELETE-USED-RECORD
     END-EVALUATE.
*****************************************************************
 P-DELETE-USED-RECORD.
*****************************************************************
     DELETE NULLFILE.
     IF W-NULLFILE-STATUS NOT = '00' THEN
        MOVE 'DELETE OF NULLFILE RECORD FAILED' TO W-ERROR-MSG
        PERFORM P-VSAM-ERROR
        GO TO 0000-RETURN
     END-IF.
     DISPLAY '''USED'' RECORD DELETED'.
     DISPLAY 'RECORD = ' F-NULLFILE-BUFFER.
     DISPLAY ' '.
     ADD +1 TO W-DELETE-COUNT.
*****************************************************************
 P-PROCESS-NULLUSE-RECORD.
*****************************************************************
     IF W-CURRENT-RESOURCE = 'TRAN' THEN
        MOVE F-NULLFILE-RESOURCE-NAME TO W-WORK4
        STRING
          W-WORK4 ' ' F-NULLFILE-START-DATE '  ' DELIMITED BY SIZE
          INTO F-REPORTT-DATA WITH POINTER W-REPORTT-DATA-PTR
        END-STRING
     ELSE
        MOVE F-NULLFILE-RESOURCE-NAME TO W-WORK8
        STRING
          W-WORK8 ' ' F-NULLFILE-START-DATE '  ' DELIMITED BY SIZE
          INTO F-REPORTT-DATA WITH POINTER W-REPORTT-DATA-PTR
        END-STRING
```

```
     END-IF.
     ADD +1 TO W-RESOURCE-COUNT.
     IF W-REPORTT-DATA-PTR > C-MAX-LINE-LENGTH
        PERFORM P-WRITE-REPORT-LINE
     END-IF.
*****************************************************************
 P-DATE-LOGIC-ERROR.
*****************************************************************
     DISPLAY ' '.
     DISPLAY '***************************************'.
     DISPLAY 'NULLFILE LOGIC ERROR '.
     DISPLAY 'RECORD = ' F-NULLFILE-BUFFER.
     DISPLAY 'CONTROL RECORD LAST DATE ' W-LAST-DATE.
     DISPLAY 'IS LESS THAN RESOURCE LAST DATE'.
     DISPLAY '***************************************'.
     DISPLAY ' '.
*****************************************************************
 P-WRITE-REPORT-LINE.
*****************************************************************
     WRITE F-REPORTT-BUFFER.

     IF W-REPORTT-STATUS NOT = '00' THEN
        DISPLAY 'WRITE TO REPORTT FILE FAILED'
        DISPLAY 'STATUS CODE = ' W-REPORTT-STATUS
        GO TO 0000-RETURN
     END-IF.
     MOVE SPACES TO F-REPORTT-BUFFER.
     MOVE +1     TO W-REPORTT-DATA-PTR.
*****************************************************************
 P-START-BROWSE.
*****************************************************************
     START NULLFILE
        KEY IS GREATER THAN OR EQUAL TO F-NULLFILE-KEY
     END-START.
     IF W-NULLFILE-STATUS NOT = '00' THEN
        MOVE 'START BROWSE OF NULLFILE FAILED' TO W-ERROR-MSG
        PERFORM P-VSAM-ERROR
        GO TO 0000-RETURN
     END-IF.
*****************************************************************
 C-REPORT-TOTALS.
*****************************************************************
     PERFORM P-WRITE-REPORT-LINE.
     MOVE 'TOTALS' TO F-REPORTT-DATA.
     PERFORM P-WRITE-REPORT-LINE.
     MOVE '______' TO F-REPORTT-DATA.
     PERFORM P-WRITE-REPORT-LINE.
     MOVE W-RESOURCE-COUNT TO W-WORK-PIC.
     STRING
        'NULL USE RESOURCES = ' W-WORK-PIC DELIMITED BY SIZE
        INTO F-REPORTT-DATA
```

```
           END-STRING.
           PERFORM P-WRITE-REPORT-LINE.
           IF W-DELETE-COUNT > +0 THEN
              MOVE W-DELETE-COUNT TO W-WORK-PIC
              STRING
                 'DELETED RESOURCES  = ' W-WORK-PIC DELIMITED BY SIZE
                 INTO F-REPORTT-DATA
              END-STRING
              PERFORM P-WRITE-REPORT-LINE
              MOVE '(SEE SYSOUT FOR DETAILS OF DELETED RECORDS)' TO
                 F-REPORTT-DATA
              PERFORM P-WRITE-REPORT-LINE
           END-IF.
           PERFORM P-WRITE-REPORT-LINE.
      ****************************************************************
       P-INITIALIZE.
      ****************************************************************
           PERFORM P-OPEN-REPORTT.
           PERFORM P-OPEN-NULLFILE.
           PERFORM P-READ-CONTROL-RECORD.
           MOVE LOW-VALUES TO F-NULLFILE-BUFFER.
           DISPLAY ' '.
      **** CHECK FOR VALID RESOURCE TYPE - '*' IS DEFAULT
           IF L-PARM-LENGTH < LENGTH OF W-RESOURCE-PARM THEN
              SET SW-ALL TO TRUE
              DISPLAY 'RESOURCE OPTION = ''*'''
              DISPLAY ' '
           ELSE
              MOVE L-PARMS(1:LENGTH OF W-RESOURCE-PARM)
                 TO W-RESOURCE-PARM
              PERFORM P-CHECK-RESOURCE-TYPE
           END-IF.
      ****************************************************************
       P-CHECK-RESOURCE-TYPE.
      ****************************************************************
           EVALUATE TRUE
              WHEN W-RESOURCE-PARM = SPACES OR LOW-VALUES
                 SET SW-ALL TO TRUE
                 DISPLAY 'RESOURCE OPTION = ''*'''
              WHEN SW-FILE
                 DISPLAY 'RESOURCE OPTION = FILE'
                 MOVE 'FILE' TO F-NULLFILE-RESOURCE-TYPE
                 SET SW-CURRENT-FILE TO TRUE
              WHEN SW-PROG
                 DISPLAY 'RESOURCE OPTION = PROG'
                 MOVE 'PROG' TO F-NULLFILE-RESOURCE-TYPE
                 SET SW-CURRENT-PROG TO TRUE
              WHEN SW-TRAN
                 DISPLAY 'RESOURCE OPTION = TRAN'
                 MOVE 'TRAN' TO F-NULLFILE-RESOURCE-TYPE
                 SET SW-CURRENT-TRAN TO TRUE
```

```
         WHEN SW-ALL
            DISPLAY 'RESOURCE OPTION = ''*'''
         WHEN OTHER
            MOVE +20 TO W-RETURN-CODE-PIC
            STRING
               W-RESOURCE-PARM ' IS AN INVALID RESOURCE TYPE.'
               ' MUST BE ''FILE'', ''PROG'', ''TRAN'' OR ''*'''
               DELIMITED BY SIZE INTO W-ERROR-MSG
            END-STRING
            DISPLAY W-ERROR-MSG
            GO TO 0000-RETURN
      END-EVALUATE.
      DISPLAY ' '.
*****************************************************************
 P-READ-CONTROL-RECORD.
*****************************************************************
      MOVE LOW-VALUES TO F-NULLFILE-KEY.
      READ NULLFILE
         INVALID KEY
            DISPLAY 'CONTROL RECORD MISSING OR FILE EMPTY'
            GO TO 0000-RETURN
      END-READ.
      IF W-NULLFILE-STATUS NOT = '00' THEN
         MOVE 'READ CONTROL RECORD FAILED' TO W-ERROR-MSG
         PERFORM P-VSAM-ERROR
         GO TO 0000-RETURN
      END-IF.
      MOVE F-NULLFILE-LAST-DATE  TO W-LAST-DATE.
      MOVE F-NULLFILE-START-DATE TO W-START-DATE.
*****************************************************************
 P-OPEN-REPORTT.
*****************************************************************
      OPEN OUTPUT REPORTT.
      IF W-REPORTT-STATUS NOT = '00' THEN
         DISPLAY 'OPEN FAILED FOR REPORTT FILE'
         GO TO 0000-RETURN
      END-IF.
      SET SW-REPORTT-OPEN TO TRUE.
*****************************************************************
 P-OPEN-NULLFILE.
*****************************************************************
      OPEN I-O NULLFILE.
      IF W-NULLFILE-STATUS NOT = '00' THEN
         MOVE 'OPEN OF NULLFILE FAILED' TO W-ERROR-MSG
         PERFORM P-VSAM-ERROR
         GO TO 0000-RETURN
      END-IF.
      SET SW-NULLFILE-OPEN TO TRUE.
*****************************************************************
 P-CLEANUP.
*****************************************************************
```

```
           IF SW-NULLFILE-OPEN THEN
              CLOSE NULLFILE
              SET SW-NULLFILE-CLOSED TO TRUE
           END-IF.
           IF SW-REPORTT-OPEN THEN
              CLOSE REPORTT
              SET SW-REPORTT-CLOSED TO TRUE
           END-IF.
      ****************************************************************
       P-VSAM-ERROR.
      ****************************************************************
           MOVE W-VSAM-RETURN-CODE TO W-RETURN-CODE-PIC.
           DISPLAY W-ERROR-MSG.
           DISPLAY 'STATUS CODE      = ' W-NULLFILE-STATUS.
           DISPLAY 'VSAM RETURN CODE = ' W-VSAM-RETURN-CODE.
           DISPLAY 'VSAM REASON CODE = ' W-VSAM-REASON-CODE.
           DISPLAY 'RECORD KEY       = ' F-NULLFILE-KEY.
```

NULLFILE

```
      ************************************************************
      * COPY BOOK - RECORD DESCRIPTION FOR NULLFILE
      * RECORDS NULL-USE RESOURCES
      ************************************************************
       FD NULLFILE
           RECORD CONTAINS 40 CHARACTERS.
       01 F-NULLFILE-BUFFER.
          02 F-NULLFILE-KEY.
             03 F-NULLFILE-RESOURCE-TYPE             PIC X(4).
             03 F-NULLFILE-DUMMY-CHAR                PIC X.
             03 F-NULLFILE-RESOURCE-NAME             PIC X(8).
          02 F-NULLFILE-DATA.
             03 F-NULLFILE-NULLUSE-COUNT             PIC S9(8) COMP.
             03 FILLER                               PIC X.
             03 F-NULLFILE-START-DATE                PIC X(10).
             03 FILLER                               PIC X.
             03 F-NULLFILE-LAST-DATE                 PIC X(10).
             03 FILLER                               PIC X.
      *
```

STATJOBP

```
//STATJOBP JOB 'account info','NULL STATS',
//            MSGLEVEL=(1,1),
//            MSGCLASS=3,
//            REGION=6M,
//            CLASS=A,
//            TIME=NOLIMIT
//*
```

```
//SETENV       SET ENV=P           *TARGET ENVIRONMENT*
//*
//*****************************************************************
//* THIS JOB ANALYSES DFHSTUP OUTPUT FOR SELECTED CICS(ES) AND    *
//* LISTS NOT-USED RESOURCES.                                     *
//*****************************************************************
//EXTRACTT  EXEC CICSSTAT,ENV=&ENV,TYPE=TOR
//EXTRACTA  EXEC CICSSTAT,ENV=&ENV,TYPE=AOR
//EXTRACTF  EXEC CICSSTAT,ENV=&ENV,TYPE=FOR
//
```

CICSSTAT

```
//CICSSTAT PROC ENV=,TYPE=ALL
//*
//* PROC TO EXECUTE REXX/ISPF IN BATCH TO GET DFHSTUP NULL-USE STATS
//*
//* INITIALISE WORK FILES
//INIT     EXEC PGM=IEFBR14
//WORK1   DD  DSN=&&WORK1,DISP=(NEW,PASS,DELETE),
//         UNIT=SYSDA,SPACE=(CYL,(2Ø,1Ø)),
//         LRECL=2Ø,BLKSIZE=32ØØ,RECFM=FB,DSORG=PS
//SORTFILE DD DSN=&&SORTFILE,DISP=(NEW,PASS,DELETE),
//         UNIT=SYSDA,SPACE=(CYL,(2Ø,1Ø)),
//         LRECL=2Ø,BLKSIZE=32ØØ,RECFM=FB,DSORG=PS
//PARMFILE DD DSN=&&PARMFILE,DISP=(NEW,PASS,DELETE),
//         UNIT=SYSDA,SPACE=(TRK,(1,Ø)),
//         LRECL=28,RECFM=F,DSORG=PS
//*
//IFØ1     IF (RC=Ø) THEN
//ISPF     EXEC PGM=IKJEFTØ1,DYNAMNBR=32,REGION=5M
//NULLFILE DD DUMMY
//WORK1    DD DSN=&&WORK1,            WORK FILE
//         DISP=(MOD,PASS,DELETE),BUFNO=4Ø
//SORTFILE DD DUMMY
//PARMFILE DD DSN=&&PARMFILE,         WORK FILE
//         DISP=(OLD,PASS,DELETE),BUFNO=4Ø
//ISPLLIB  DD DSN=your.ISPLLIB,DISP=SHR
//ISPMLIB  DD DSN=your.ISPMLIB,DISP=SHR
//ISPSLIB  DD DSN=your.ISPSLIB,DISP=SHR
//ISPPLIB  DD DSN=your.ISPPLIB,DISP=SHR
//ISPTLIB  DD DSN=your.ISPTABL,DISP=SHR
//         DD DSN=your.ISPF.ISPPROF,DISP=SHR
//ISPTABL  DD DSN=your.ISPF.ISPPROF,DISP=SHR
//ISPPROF  DD DSN=your.ISPF.ISPPROF,DISP=SHR
//ISPLOG   DD SYSOUT=*,
//       RECFM=FBA,LRECL=121,BLKSIZE=3146
//ISPLIST  DD SYSOUT=*,
//       RECFM=FBA,LRECL=121,BLKSIZE=3146
```

```
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSPROC  DD DSN=your.EXEC,DISP=SHR
//SYSTSIN  DD DSN=your.PROC.DATA(ISPREFIX),DISP=SHR
//         DD DSN=your.PROC.DATA(STAT&ENV&TYPE),DISP=SHR
//*
//* SORT THE OUTPUT
//IFØ2     IF (RC=Ø) THEN
//SORT     EXEC PGM=SORT
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DSN=&&WORK1,            INPUT WORK FILE
//         DISP=(OLD,DELETE,DELETE),BUFNO=4Ø
//SORTOUT  DD DSN=&&SORTFILE,         OUTPUT WORK FILE
//         DISP=(OLD,PASS,DELETE),BUFNO=4Ø
//SYSIN    DD DSN=your.PROC.DATA(STATSORT),DISP=SHR
//*
//IFØ3     IF (RC=Ø) THEN
//ANALYSE  EXEC PGM=STATPRG2
//STEPLIB  DD DSN=your.ISPLLIB,DISP=SHR
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//NULLFILE DD DSN=CICS.&ENV..NULLUSE.RESOURCE,DISP=SHR,BUFNO=4Ø
//SORTFILE DD DSN=&&SORTFILE,         WORK FILE
//         DISP=(OLD,DELETE,DELETE),BUFNO=4Ø
//PARMFILE DD DSN=&&PARMFILE,         WORK FILE
//         DISP=(OLD,DELETE,DELETE),BUFNO=4Ø
//*
//ENDIFØ3  ENDIF
//ENDIFØ2  ENDIF
//ENDIFØ1  ENDIF
//*
```

PROCEDURE DATA IN YOUR.PROC.DATA

```
ISPREFIX
 PREFIX userid
STATPAOR
ISPSTART CMD(%STATEXTR CICS.PA*.DFHSTUP *)(Prod AORs-Tran, File & Prog)
STATPTOR
ISPSTART CMD(%STATEXTR CICS.PTØ*.DFHSTUP TRAN)(Prod TOR - transactions)
STATPFOR
ISPSTART CMD(%STATEXTR CICS.PF*.DFHSTUP FILE)     (Prod FOR - Files)
STATSORT
  SORT FIELDS=(1,2Ø,CH,A)
```

David Roth
CICS Consultant (Germany)

IBM has announced three new Redbooks, which are relevant to CICS users. The redbooks are: *A Performance Study of Web Access to CICS*; *Java Application Development for CICS*; and *CICS Transaction Server for OS/390 Version 1 Release 3: Web Support and 3270 Bridge*.

For further information contact your local IBM representative.
URL: http://www-4.ibm.com/software/ts/cics/library/index.html#redbooks.

* * *

Peerlogic has announced LiveContent PIPES Access for CICS Version 1.2, which provides connectivity between COBOL application transaction programs in the CICS address space and the LiveContent PIPES kernel for MVS. The LiveContent PIPES kernel communicates with the PAPI (and with LiveContent PIPES applications) through a cross-memory interprocess communication (IPC) mechanism.

The PAPI allows applications to access LiveContent PIPES services, such as advertising or finding a resource, establishing a session, and sending and receiving messages.

Application programs link to the CICS/COBOL interface program to request a LiveContent PIPES service. LiveContent PIPES Access for CICS includes a Master Transaction Program that applications use to handle all asynchronous events, and a Session Request Program that applications use, or customize, to process session requests from remote client programs. Several sample transaction programs are also included to help perform administrative functions, verify that LiveContent Pipes Access for CICS is properly installed, and serve as a model for developing LiveContent PIPES Access for CICS applications.

LiveContent PIPES Access for CICS Version 1.2 requires: LiveContent PIPES for MVS (Version 3.8P02 or above); MVS/ESA Version 4.1 or above; VS COBOL/II or COBOL/370; and CICS/ESA Version 3.3 or above.

For further information contact:Peerlogic, 555 De Haro Street, Suite 300, San Francisco, CA 94107-2348, USA.
Tel: (415) 626 4545.
URL: http://www.peerlogic.com/products/pipes/pipescics.html

* * *

The bad news for CICS/VSE Version 2.3 users is that IBM has withdrawn it from marketing. The program has been replaced by CICS TS for VSE/ESA Version 1.

CICS/VSE Version 2.3 will continue to be available as part of the CICS Transaction Server for VSE/ESA, delivered on the extended base tape of VSE/ESA Version 2.4 (or higher), and can be installed in the co-existence environment.

For further information contact your local IBM representative.
URL: http://www.ibmlink.ibm.com/usalets&parms=H_299-156