182 CICS update

January 2001

In this issue

3 Displaying task activity in a CICS region under stress

11 Yet another cold start next time ?

17 CICS ISC/MRO session monitoring

28 List all installed global user exits

35 Extract data from CSD to manage CICS resources from a DB2 environment

48 CICS news

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.

Displaying task activity in a CICS region under stress

A spate of CICS problems at a site where I have recently been consulting highlighted for me how difficult it is to find out exactly what is going on inside a CICS region once that region develops a serious internal problem preventing attempts to start diagnostic transactions.

There are several excellent third-party monitoring packages on the market for the CICS environment, but not every site has one of these products installed, and there are instances in my experience where even monitoring products may not be able to tell you exactly what you need to know.

The usual course of action at this point is often to cancel the CICS region with a dump, and then try to figure out what was going on and where the problem arose. Dump analysis is few people's favourite pastime, and any assistance in this process is invaluable. In addition to this, with recurring problems it is not always desirable to wait for the dumping process to run. However, one might still want to extract some information from the failing region before restarting it.

What I needed was a method to look inside a CICS region from another address space. I consulted the Xephon archives and found a number of programs that communicate with a CICS region, but this could not help me because I wanted to extract information from a CICS region that had ceased communicating. Then I found *Monitoring CICS from TSO* in *CICS Update* Issue 82, September 1992, and I knew that the access register method described was just what I needed.

Since my requirement was for a permanent record, and I wished to automate the execution of the program if at all possible, I developed a small batch program that can be run as a started task to look inside an executing CICS address space and follow some control block chains to produce a report on some aspects of the transaction activity it finds. I was also monitoring CICS Version 4.1 regions and needed to alter the logic accordingly.

The result is the program MAPTCA, which, as the name suggests, follows the TCA chains to report on tasks present in the CICS region. Here is a sample of the output from MAPTCA:

```
DATE:    20000701     TIME:    120000

CICSJOB: CICSV41A     CURRENT TCA: 0705E680

TRAN  TASKID   RESNAME   RESTYPE   TCA       STATUS      LUWID

CEDF  0000165  DFHZARQ1  ZCIOWAIT  0705F680  SUSPENDED   OURNET.CICSV41A
SNCB  0000160  DBUGUSER  EDF       00057680  SUSPENDED   OURNET.LUTP0008
CEMT  0000154  DFHZARQ1  ZCIOWAIT  00054080  SUSPENDED   OURNET.LUTP0006
CSNE  0000024                      0705F080  SUSPENDED   OURNET.CICSV41A
CSNC  0000023  MROQUEUE  CSNC      00078080  SUSPENDED   OURNET.CICSV41A
CSSY  0000022  SINGLE    KCCOMPAT  00056680  SUSPENDED   OURNET.CICSV41A
CSSY  0000021  DFHJ01A   JCJOURDS  00057080  SUSPENDED   OURNET.CICSV41A
CSSY  0000020  SUBTASK   JCTERMN   00078680  SUSPENDED   OURNET.CICSV41A
CSSY  0000008  JABSUTOK  JCJASUS   00056080  SUSPENDED   OURNET.CICSV41A
CSSY  0000007  DFHAPTIX  ICEXPIRY  00055680  SUSPENDED   OURNET.CICSV41A
CSSY  0000006  DFHAPTIM  ICMIDNTE  00055080  SUSPENDED   OURNET.CICSV41A
CSTP  0000004  DFHZDSP   TCP_NORM *0705E680* SUSPENDED   CICSV41A
      0000000  DS_NUDGE  TIEXPIRY  00000000  SUSPENDED
      0000000            SMSYSTEM  00000000  SUSPENDED
```

This CICS region is idle, as evidenced by the CSTP task 4 being the current TCA, and all the tasks suspended. The current TCA is highlighted by the '*' in the TCA column. There are two terminal sessions active, both waiting for terminal I/O. One of the sessions is running transaction SNCB under CEDF.

MAPTCA can be run against a CICS region at any time, with the caveat that a very busy region which is at a high dispatching priority relative to MAPTCA might update the control block chains while MAPTCA is still running them. If this happens then the output might be suspect, but there will never be any damage to CICS because MAPTCA is strictly a 'look but don't touch' application.

Note that MAPTCA must be linked with AC=1 into an APF library.

MAPTCA SOURCE

```
*********************************************************************
**  MAP CICS TASKS                                                 **
*********************************************************************
         LCLC     &MODULE
```

```
&MODULE  SETC     'MAPTCA'
&MODULE  CSECT
&MODULE  AMODE    31
&MODULE  RMODE    24
         YREGS
         SAVE (14,12)
         USING MAPTCA,R12
         LR    R12,R15
         LR    R14,R13
         LA    R13,SAVE
         ST    R13,8(,R14)
         ST    R14,4(,R13)
*--------------------------------------------------------------------
* READ PARAMETER - CICS REGION TO PROCESS
*--------------------------------------------------------------------
         LR    R11,R1                      PARAMETER POINTER
         L     R1Ø,Ø(R11)
         MVC   JOBNAME(8),2(R1Ø)
*--------------------------------------------------------------------
* FIND ASID OF CICS REGION
*--------------------------------------------------------------------
         L     R11,CVTPTR                  GET ADDRESS OF CVT
         L     R11,CVTASVT-CVTMAP(R11)     ASVT POINTER
         USING ASVT,R11                    ASVT ADDRESSABILITY
         LA    R1Ø,ASVTENTY                ADDRESS OF ASVT ENTRIES
         L     R9,ASVTMAXU                 MAX ADDRESS SPACES
ASVT_LOOP_ROUTINE DS ØH
         TM    Ø(R1Ø),ASVTAVAL             IS THE SLOT OCCUPIED ?
         BO    TRY_NEXT_ASCB               NO, THEN BYPASS
         L     R8,Ø(R1Ø)                   GET POINTER TO ASCB
         USING ASCB,R8                     ASCB ADDRESSABILITY
         L     R1,ASCBJBNI                 GET JOBNAME POINTER
         LTR   R1,R1                       JOBNAME ?
         BZ    TRY_STC_FOR_JOBNAME         NO, STC MAYBE
         CLC   JOBNAME(8),Ø(R1)
         BE    FOUND_CICS
TRY_STC_FOR_JOBNAME DS ØH
         L     R1,ASCBJBNS                 START/MOUNT/LOGON NAME ?
         LTR   R1,R1                       IS IT   ?
         BZ    TRY_NEXT_ASCB               NO, JUST CONTINUE
         CLC   JOBNAME(8),Ø(R1)
         BE    FOUND_CICS
TRY_NEXT_ASCB  DS ØH
         LA    R1Ø,4(R1Ø)                  POINT TO NEXT ASCB
         BCT   R9,ASVT_LOOP_ROUTINE        CONTINUE...
CICS_NOT_RUNNING DS ØH
         OPEN  (SYSPRINT,OUTPUT)
         MVC   OUTREC+2(12),=CL12'CICS REGION '
         MVC   OUTREC+14(8),JOBNAME
         MVC   OUTREC+22(1Ø),=CL1Ø' NOT FOUND'
```

```
         BAL   R9,WRITE_RECORD_TO_SYSPRINT
         CLOSE SYSPRINT
         B     @FINISH
FOUND_CICS DS  ØH
         MVC   ARASID,ASCBASID
*--------------------------------------------------------------------
* GETMAIN AREA TO STORE INFO - R7 FOR TABLE POINTER
*--------------------------------------------------------------------
         GETMAIN RU,LV=16ØØØ
         LR    R7,R1
         ST    R7,TABSTART
*--------------------------------------------------------------------
* INTO ACCESS MODE - R6 FOR CICS ADDRESS SPACE
*--------------------------------------------------------------------
         MODESET MODE=SUP,KEY=ZERO
         AXSET AX=AX1
         LH    R4,ARASID
         SSAR  R4
         SAC   512
         LAM   R6,R6,=F'1'
*--------------------------------------------------------------------
* FOLLOW CICS STORAGE CHAINS
*--------------------------------------------------------------------
         L     R6,KEKCB              DFHKEKCB
         L     R6,Ø(,R6)             DFHDSANCHOR
         LA    R6,172(,R6)
         L     R6,Ø(,R6)             DTA
PROCESS_DTA DS ØH
         LR    R5,R6                 STORE DTA
         MVC   Ø(8,R7),12(R6)        RESOURCE NAME
         MVC   8(8,R7),2Ø(R6)        RESOURCE TYPE
         MVC   16(1,R7),6Ø(R6)       TASK STATE
         LA    R6,12Ø(,R6)
         L     R6,Ø(,R6)             XMTXN
         LTR   R6,R6                 XMTXN PRESENT?
         BZ    NEXT_DTA
         MVC   17(4,R7),48(R6)       TASK NUMBER
         MVC   21(4,R7),6Ø(R6)       TRANSID
         LA    R6,12Ø(,R6)
         L     R6,Ø(,R6)             TCA
         LTR   R6,R6                 TCA PRESENT?
         BZ    NEXT_DTA
PROCESS_TCA DS ØH
         MVC   25(18,R7),617(R6)     LUWID LENGTH(1 BYTE) AND LUWID
         ST    R6,43(,R7)            TCA ADDRESS
         LA    R6,22Ø(,R6)
         L     R6,Ø(,R6)             CSA
PROCESS_CSA DS ØH
         MVC   CSACDTA,76(R6)        CURRENTLY DISPATCHED TCA
NEXT_DTA DS    ØH
```

```
         LR    R6,R5                RESTORE DTA
         LA    R7,47(,R7)
         LA    R6,36(,R6)
         L     R6,Ø(,R6)            NEXT DTA
         LTR   R6,R6                LAST DTA?
         BNZ   PROCESS_DTA
         ST    R7,TABEND
*-------------------------------------------------------------------
* OUT OF ACCESS MODE
*-------------------------------------------------------------------
         EPAR  R2
         SSAR  R2
         SAC   Ø
         AXSET AX=AXØ
         MODESET MODE=PROB,KEY=NZERO
         L     R7,TABSTART
         L     R6,TABEND
         OPEN  (SYSPRINT,OUTPUT)
WRITE_HEADER DS ØH
         TIME  DEC,TIMEDATE,LINKAGE=SYSTEM,DATETYPE=YYYYMMDD
         MVC   OUTREC+2(9),=CL9'DATE:    '
         L     R5,TIMEDATE+8
         R2D   R5,OUTREC+11(8)
         MVC   OUTREC+24(9),=CL9'TIME:    '
         L     R5,TIMEDATE
         R2D   R5,OUTREC+33(6)
         BAL   R9,WRITE_RECORD_TO_SYSPRINT
         BAL   R9,WRITE_RECORD_TO_SYSPRINT
         MVC   OUTREC+2(9),=CL9'CICSJOB: '
         MVC   OUTREC+11(8),JOBNAME
         MVC   OUTREC+24(13),=CL13'CURRENT TCA: '
         L     R5,CSACDTA
         R2D   R5,OUTREC+37(8)
         BAL   R9,WRITE_RECORD_TO_SYSPRINT
         BAL   R9,WRITE_RECORD_TO_SYSPRINT
         MVC   OUTREC+2(35),=CL35'TRAN  TASKID   RESNAME   RESTYPE   '
         MVC   OUTREC+37(28),=CL28'TCA       STATUS       LUWID'
         BAL   R9,WRITE_RECORD_TO_SYSPRINT
         BAL   R9,WRITE_RECORD_TO_SYSPRINT
WRITE_DETAIL DS ØH
         MVC   OUTREC+2(4),21(R7)        TRANS
         L     R5,17(R7)
         R2D   R5,OUTREC+8(7)              TASKID
         MVC   OUTREC+17(8),Ø(R7)        RESOURCE NAME
         MVC   OUTREC+27(8),8(R7)        RESOURCE TYPE
         L     R5,43(R7)
         R2D   R5,OUTREC+37(8)             TCA ADDRESS
         CLC   CSACDTA,43(R7)            CURRENT TCA?
         BNE   TASK_STATE_FC
         MVC   OUTREC+36(1),=CL1'*'
```

```
         MVC   OUTREC+45(1),=CL1'*'
TASK_STATE_FC DS ØH
         CLC   TASK_FC,16(R7)
         BNE   TASK_STATE_Ø1
         MVC   OUTREC+47(9),=CL9'SUSPENDED'
         B     PROCESS_LUWID
TASK_STATE_Ø1 DS ØH
         CLC   TASK_Ø1,16(R7)
         BNE   TASK_STATE_Ø3
         MVC   OUTREC+47(12),=CL12'NON EXECUTABLE'
         B     PROCESS_LUWID
TASK_STATE_Ø3 DS ØH
         CLC   TASK_Ø3,16(R7)
         BNE   TASK_STATE_Ø4
         MVC   OUTREC+47(12),=CL12'DISPATCHABLE'
         B     PROCESS_LUWID
TASK_STATE_Ø4 DS ØH
         CLC   TASK_Ø4,16(R7)
         BNE   TASK_STATE_Ø5
         MVC   OUTREC+47(12),=CL12'RUNNING'
         B     PROCESS_LUWID
TASK_STATE_Ø5 DS ØH
         CLC   TASK_Ø5,16(R7)
         BNE   TASK_STATE_FA
         MVC   OUTREC+47(12),=CL12'RESUMED'
         B     PROCESS_LUWID
TASK_STATE_FA DS ØH
         CLC   TASK_FA,16(R7)
         BNE   TASK_STATE_FD
         MVC   OUTREC+47(6),=CL6'PURGED'
         B     PROCESS_LUWID
TASK_STATE_FD DS  ØH
         CLC   TASK_FD,16(R7)
         BNE   TASK_STATE_FE
         MVC   OUTREC+47(6),=CL6'UNUSED'
         B     PROCESS_LUWID
TASK_STATE_FE DS ØH
         CLC   TASK_FE,16(R7)
         BNE   TASK_STATE_XX
         MVC   OUTREC+47(5),=CL5'RESET'
         B     PROCESS_LUWID
TASK_STATE_XX DS ØH
         MVC   OUTREC+47(8),=CL8'??    ??'
         L     R5,16(R7)
         R2D   R5,OUTREC+5Ø(2)
PROCESS_LUWID DS ØH
         MVC   OUTREC+6Ø(17),26(R7)      LUWID
         XR    R4,R4
         IC    R4,25(,R7)
         LTR   R4,R4
```

```
         BZ    NEXT_SECTION_OF_GETMAIN
         ST    R4,LUWIDL
         L     R3,F17
         SR    R3,R4
         LA    R4,OUTREC+6Ø
         A     R4,LUWIDL
SPACE_LUWID DS ØH
         MVC   Ø(1,R4),=CL1' '
         A     R4,FØ1
         BCT   R3,SPACE_LUWID
NEXT_SECTION_OF_GETMAIN DS ØH
         LA    R7,47(,R7)
         BAL   R9,WRITE_RECORD_TO_SYSPRINT
         CR    R7,R6
         BNE   WRITE_DETAIL
         CLOSE SYSPRINT
         L     R7,TABSTART
         FREEMAIN RU,LV=16ØØØ,A=(7)
@FINISH  L     R13,SAVE+4
         RETURN (14,12),RC=Ø
*-------------------------------------------------------------------*
*  WRITE TO SYSPRINT AND CLEAR OUTREC                               *
*-------------------------------------------------------------------*
WRITE_RECORD_TO_SYSPRINT DS ØH
         PUT   SYSPRINT,OUTCARD
         MVI   OUTREC,C' '
         MVC   OUTREC+1(132),OUTREC
         BR    R9                  RETURN TO CALLER
*-------------------------------------------------------------------*
*  WORKING STORAGE                                                  *
*-------------------------------------------------------------------*
         DS    ØD
SAVE     DS    18F
AXØ      DC    H'Ø'
AX1      DC    H'1'
JOBNAME  DC    CL8'CICSNAME'
TABSTART DS    F
TABEND   DS    F
PPAN     DS    F
PPXN     DS    F
FØ1      DC    F'1'
F17      DC    F'17'
CSACDTA  DS    F
LUWIDL   DS    F
TIMEDATE DS    ØCL16         TIME AND DATE RETURNED
         DC    XL16'ØØ'
ARASID   DS    H
KEKCB    DS    ØF
         DC    XL4'ØØØØ6518'
```

```
TASK_Ø1  DC    XL1'Ø1'
TASK_Ø3  DC    XL1'Ø3'
TASK_Ø4  DC    XL1'Ø4'
TASK_Ø5  DC    XL1'Ø5'
TASK_FA  DC    XL1'FA'
TASK_FC  DC    XL1'FC'
TASK_FD  DC    XL1'FD'
TASK_FE  DC    XL1'FE'
OUTCARD  DC    AL2(137),AL2(Ø)
OUTREC   DC    CL133' '
         ORG   OUTREC+133
SYSPRINT DCB   DDNAME=SYSPRINT,DSORG=PS,MACRF=PM,                      X
               LRECL=137,BLKSIZE=137Ø,RECFM=VB
         IHAASVT
         IHAASCB
         CVT   DSECT=YES
         END
```

MAPTCA uses macro R2D to convert 4-byte 'register' type fields to an 8-byte displayable value.

R2D MACRO

```
*********************************************************************
**  Convert the contents of a passed register to an 8-character      **
**  display field.                                                   **
*********************************************************************
         MACRO
&LABEL   R2D   &HEX,&DSP
&LABEL   STM   Ø,15,SAVE&SYSNDX
         ST    &HEX,WHEX&SYSNDX
         UNPK  WDSP&SYSNDX.(9),WHEX&SYSNDX.(5)
         NC    WDSP&SYSNDX.(8),MASK&SYSNDX
         TR    WDSP&SYSNDX.(8),HXTB&SYSNDX
         MVC   &DSP,WDSP&SYSNDX
         LM    Ø,15,SAVE&SYSNDX
         B     END&SYSNDX
SAVE&SYSNDX DS 16F
MASK&SYSNDX DC XL8'ØFØFØFØFØFØFØFØF'
HXTB&SYSNDX DC CL16'Ø123456789ABCDEF'
WHEX&SYSNDX DS F
            DS C
WDSP&SYSNDX DS CL8'********'
            DC CL1'.'
END&SYSNDX  DS ØH
         MEND
```

MAPTCA JCL

```
//*-------------------------------------------------------------------*
//*  Note that PARM must be an 8-byte field, pad with blanks if the  *
//*  CICS region name to map is not 8 bytes long.                     *
//*-------------------------------------------------------------------*
//MAPTCA   EXEC PGM=MAPTCA,PARM='CICSV41A'
//STEPLIB  DD DSN=YOUR.APF.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=X
//*
```

Patrick Mullen
Independent Consultant (Canada)

Yet another cold start next time

Now here is another solution for the ‘problem’ of a COLD CICS start after a ‘normal’ shutdown. Our solution bases itself on a simple COBOL program, called CHECKGCD (see below), which manages CICS start up.

As everyone know, when a CICS starts, it’s possible to pass to DFHSIP some initialization parameters through a PARM field. This can be done in three different ways:

- Directly into the PARM (PARM='SIT=TS, START=INITIAL, .END').
- Indirectly through a SYSIN (PARM='SYSIN').
- In a mixed way (PARM='START=INITIAL,SYSIN').

CHECKGCD, included in a CICS start JCL before DFHSIP, modifies this SYSIN, if needed (see below). It works in a simple way – it checks the value of the record relative to the kind of start in the CICS GLOBAL CATALOG (DFHGCD) (see below), deciding the type of CICS start. The program works with ESA 4.1 and TS 1.3 CICS versions simply passing it a PARM where you need to specify that version: ‘CICS41’ or ‘CICS53’, respectively.

An example of a SYSIN for DFHSIP is shown below:

```
APPLID=DYNASTY
INITPARM=(DFHD2INI='00,DB9')
DB2CONN=YES
DSALIM=7M
EDSALIM=90M
GRPLIST=PIPPO
GMTEXT='Have a nice day with DYNASTY CICS ! ... and ... Good luck ! ;)'
MN=ON
MNCONV=NO
MNPER=ON
PLTPI=DY
PLTSD=YD
SIT=TS<= In this compiled SIT (DFHSITTS) START field must be set to AUTO
SYSIDNT=CEDY
.END
```

As you can see, in this file are specified some parameters for the CICS you are starting. SYSIN entries update those specified in the compiled SIT. Thus it's possible to have only one compiled SIT and one SYSIN for every CICS region.

The START parameter in the compiled SIT needs to be set to AUTO (permitting an EMERGENCY START) and the SYSIN can't contain a reference to this parameter.

Below is part of CICS ESA 4.1 Global Catalog that manages the start:

```
----+----1----+----2----+--- -3----+----4----+---
    CTL     CTL              CTL            ¬
0001CED40044CED4444444444444 CED4000000400051159
0000333000003330000000000000 333000000000108F161F
------------------------------------------------
```

Below is part of CICS TS 1.3 Global Catalog that manages the start:

```
----+----1----+----2----+--- -3----+----4----+----5----+----6----+----7-
---+----
    DFHRMDM DFHRMDM_ANCHOR     NETCO.DYNASTY                        !
A
0001CCCDDCD4CCCDDCD6CDCCDD44
00DCECD4CEDCEEE0000000000000000000000000B5730B3C00000
0001468944404689444D15386900
0D55336B4851238000003040000000000000000004A52E09100000
------------------------------------------------------------------------
-------
8----+----9--

0000000000000
```

```
Ø000000000000
-------------
```

This SYSIN, read and modified by CHECKGCD, becomes the SYSIN for DFHSIP through a temporary dataset. The program modifies this SYSIN only in the case of a WARM start, forcing it to COLD, otherwise it passes this SYSIN without making any modification (EMERGENCY start). How the project works is shown below:

```
IF is_possible_a_WARM_start THEN
      ADD 'START=COLD' TO PARAIN      /* Forcing CICS to start COLD */
      MOVE PARAIN TO PARAOU.
ELSE
      MOVE PARAIN TO PARAOU.
                        /* Accepting START=AUTO => EMERGENCY start */
PASS PARAOU TO DFHSIP AS SYSIN.
EXECUTE DFHSIP WITH PARM=SYSIN.
```

An example of a program call is shown below:

```
//DYNASTY  PROC
//*************************************************************
//* AUTOMATIC RESTART MANAGEMENT
//*************************************************************
//CHECKGCD  EXEC PGM=CHECKGCD,PARM='CICS53'
//STEPLIB   DD DISP=SHR,DSN=linklib
//SYSPRINT  DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//DFHGCD    DD DISP=OLD,DSN=Cics_Name.DFHGCD
//PARAIN    DD DISP=SHR,DSN=Cics_Table(Cics_Name)        <- input SYSIN
//PARAOU    DD DSN=&&STARTO,DISP=(,PASS),  <- Modified SYSIN for DFHSIP
//          UNIT=3390,SPACE=(80,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=80,RECFM=FB)
...
//*************************************************************
//******************  EXECUTE CICS  ***************************
//*************************************************************
//CICS53    EXEC PGM=DFHSIP,REGION=64M,TIME=1440,
//          PARM='SYSIN'
//SYSIN     DD DSN=&&STARTO,DISP=OLD                  <- Udpated SYSIN
...
```

CHECKGCD SOURCE

```
IDENTIFICATION DIVISION.
       PROGRAM-ID.                        CHECKGCD.
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
```

```
 FILE-CONTROL.
     SELECT  GCD-FILE        ASSIGN TO DFHGCD
                             ORGANIZATION IS INDEXED
                             ACCESS MODE IS RANDOM
                             RECORD KEY IS GCD-KEY.
     SELECT  PARM-INPUT      ASSIGN TO PARAIN
                             ORGANIZATION IS SEQUENTIAL.
     SELECT  PARM-OUTPUT     ASSIGN TO PARAOU
                             ORGANIZATION IS SEQUENTIAL.
 DATA DIVISION.
 FILE SECTION.
 FD  GCD-FILE
     LABEL RECORDS STANDARD
     DATA RECORD IS GCD-RECORD.
 Ø1  GCD-RECORD.
     Ø3  GCD-KEY PIC X(28).
     Ø3  FILLER PIC X(4Ø61).
 FD  PARM-INPUT
     RECORDING MODE IS F
     RECORD CONTAINS 8Ø CHARACTERS
     DATA RECORD DATA-INPUT.
 Ø1  DATA-INPUT PIC X(8Ø).
 FD  PARM-OUTPUT
     RECORDING MODE IS F
     RECORD CONTAINS 8Ø CHARACTERS
     DATA RECORD DATA-OUTPUT.
 Ø1  DATA-OUTPUT PIC X(8Ø).
 WORKING-STORAGE SECTION.
 Ø1 CTL-KEY-CX41 PIC X(28) VALUE '   _CTL     CTL             '.
* WHAT YOU SEE    '    CTL     CTL             ' *
* WHAT YOU WRITE X'ØØØ1CED4ØØ44CED4444444444444  *
* WITH HEX ON      ØØØØ333ØØØØØ333ØØØØØØØØØØØØ' *
 Ø1 RMDM-KEY-TS13 PIC X(28) VALUE '   _DFHRMDM DFHRMDM_ANCHOR  '.
* WHAT YOU SEE    '    DFHRMDM DFHRMDM_ANCHOR  ' *
* WHAT YOU WRITE X'ØØØ1CCCDDCD4CCCDDCD6CDCCDD44  *
* WITH HEX ON      ØØØ146894440468944 4D15386ØØ' *
 Ø1  GCD-RECORD-IN-CX41.
     Ø3  GCD-KEY-IN-CX41 PIC X(28).
        88 DFHCTL-KEY-CX41 VALUE '   _CTL     CTL             '.
* SEE ABOVE *
     Ø3  FILLER PIC X(1Ø).
     Ø3  GCD-KEYPOINT-CX41 PIC X(1).
        88 GCD-WARMKEYP-CX41 VALUE '-'.
* WHAT YOU SEE    '-' *
* WHAT YOU WRITE X'6  *
* WITH HEX ON      Ø' *
        88 GCD-EMERKEYP-CX41 VALUE ' '.
* WHAT YOU SEE    ' ' *
* WHAT YOU WRITE X'4  *
* WITH HEX ON      Ø' *
```

```
     Ø3  FILLER PIC X(9).
 Ø1  GCD-RECORD-IN-TS13.
     Ø3  GCD-KEY-IN-TS13 PIC X(28).
        88 DFHRMDM-KEY-TS13 VALUE '   _DFHRMDM DFHRMDM_ANCHOR  '.
* SEE ABOVE *
     Ø3  FILLER PIC X(16).
     Ø3  GCD-KEYPOINT-TS13 PIC X(4).
        88 GCD-WARMKEYP-TS13 VALUE '   _'.
* WHAT YOU SEE    '    ' *
* WHAT YOU WRITE X'ØØØØ  *
* WITH HEX ON      ØØØ3' *
        88 GCD-EMERKEYP-TS13 VALUE '   Ü'.
* WHAT YOU SEE    '    ' *
* WHAT YOU WRITE X'ØØØØ  *
* WITH HEX ON      ØØØ4' *
     Ø3  FILLER              PIC X(4Ø61).
 LINKAGE SECTION.
 Ø1  PARM-DATA.
     Ø5 FILLER               PIC X(2).
     Ø5 CICS-VERSION         PIC X(6).
 PROCEDURE DIVISION USING PARM-DATA.
 BEGIN.
     DISPLAY 'CHECKGCD: CICS RESTART MANAGER' UPON CONSOLE.
     OPEN INPUT PARM-INPUT.
     OPEN OUTPUT PARM-OUTPUT.
     READ PARM-INPUT
       AT END MOVE HIGH-VALUES TO DATA-INPUT
     END-READ
     OPEN INPUT GCD-FILE.
     IF CICS-VERSION = 'CICS41' GO TO CICS41Ø-RESTART-ROUTINE.
     IF CICS-VERSION = 'CICS53' GO TO CICS53Ø-RESTART-ROUTINE.
 CICS41Ø-RESTART-ROUTINE.
     MOVE CTL-KEY-CX41 TO GCD-KEY
     READ GCD-FILE INTO GCD-RECORD-IN-CX41
     IF GCD-KEY-IN-CX41 > CTL-KEY-CX41 GO TO END-ROUTINE.
     IF GCD-EMERKEYP-CX41
      DISPLAY 'CHECKGCD: EMERGENCY START' UPON CONSOLE
      MOVE '16' TO RETURN-CODE.
     IF GCD-WARMKEYP-CX41
      MOVE 'START=COLD' TO DATA-OUTPUT
      WRITE DATA-OUTPUT
      DISPLAY 'CHECKGCD: FORCING TO START=COLD' UPON CONSOLE
      MOVE ZERO TO RETURN-CODE.
     GO TO END-ROUTINE.
 CICS53Ø-RESTART-ROUTINE.
     MOVE RMDM-KEY-TS13 TO GCD-KEY
     READ GCD-FILE INTO GCD-RECORD-IN-TS13
     IF GCD-KEY-IN-TS13 > RMDM-KEY-TS13 GO TO END-ROUTINE.
     IF GCD-EMERKEYP-TS13
      DISPLAY 'CHECKGCD: EMERGENCY START' UPON CONSOLE
```

```
        MOVE '16' TO RETURN-CODE.
       IF GCD-WARMKEYP-TS13
        MOVE 'START=COLD' TO DATA-OUTPUT
        WRITE DATA-OUTPUT
        DISPLAY 'CHECKGCD: FORCING TO START=COLD' UPON CONSOLE
        MOVE ZERO TO RETURN-CODE.
       GO TO END-ROUTINE.
END-ROUTINE.
       PERFORM UNTIL DATA-INPUT = HIGH-VALUES
         MOVE DATA-INPUT TO DATA-OUTPUT
         WRITE DATA-OUTPUT
         READ PARM-INPUT
           AT END MOVE HIGH-VALUES TO DATA-INPUT
         END-READ
       END-PERFORM
       CLOSE GCD-FILE.
       CLOSE PARM-INPUT.
       CLOSE PARM-OUTPUT.
       GOBACK.
```

Gianluca Bonzano (Italy)

CICS ISC/MRO session monitoring

Recently the CICS maintenance level at my current site was upgraded from CICS/ESA Version 4.1 to CICS/TS Version 1.3. One of the problems we encountered was that CICS APPC connection sessions are getting released with the following messages:

```
DFHZC3437 I 09/25/00 10:41:07 CICSXXXX -XXN CSNE Node nnnnnnnn action
taken:
CLSDST ABTASK ABSEND ABRECV ((1) Module name: DFHZNAC)

DFHSN1500 09/25/00 10:41:07 CICSXXXX Session signoff for session -XXN is
complete. 14 transactions entered with 0   errors.

DFHZC3462 I 09/25/00 10:41:07 CICSXXXX -XXN CSNE Node nnnnnnnn session
terminated.  ((2) Module name: DFHZCLS)
```

In CICS/ESA Version 4.1 the session -XXN remained acquired but in CICS/TS V1.3 the -XXN session was released. CICS processing continued until only the session manager session remained acquired, at which time processing from the partner LU is rejected – while the connection itself remained INService and ACQuired.

While the CICS technical supports staff researched this problem we had to find a way round it. The first thing we did was to code a program to monitor the connection status, and CONNMON was coded and implemented.

CONNMON

This program monitors the use of the CICS systems programming interface to run the connection/terminal chains and displays the following information (up to 21 MRO/ISC connection entries):

```
CONNCTION NAME
NETNAME
SERVICE STATUS      INS/OUT
CONNCTIONS STATUS   ACQ/REL
ACQUIRED SESSION COUNT  (FOR NON-XM)
RELEASED SESSION COUNT  (FOR NON-XM)
```

The program refreshes itself every three minutes, and allowed us to

monitor the status of the MRO/ISC connections for the CICS region, especially the connection which is losing the sessions.

```
//JOBCARD  JOB (CCS),'CSH CONNMON  ',
//         CLASS=A,NOTIFY=&SYSUID,
//         MSGCLASS=X,MSGLEVEL=(1,1),REGION=8M
//*
//*
//CONNMON  EXEC DFHEITAL,PARM.TRN='SP' COND.LKED=(Ø,LE)
//TRN.SYSIN  DD *
*        PRINT NOGEN
         TITLE 'CSH DISPLAY MRO/ISC CONNECTION STATUS'
**********************************************************************
**********************************************************************
***                                                                ***
***  DISPLAY UP TO 21 MRO/ISC CONNECTIONS WITHE FOLLOWING DATA:    ***
***                                                                ***
***    CONNCTION NAME                                              ***
***    NETNAME                                                     ***
***    SERVICE STATUS      INS/OUT                                 ***
***    CONNCTIONS STATUS   ACQ/REL                                 ***
***    ACQUIRED SESSION COUNT  (FOR NON-XM)                        ***
***    RELEASED SESSION COUNT  (FOR NON-XM)                        ***
***                                                                ***
***  ** REFRESHES DISPLAY EVERY 3 MINUTES                          ***
***  ** CLEAR SCREEN/FREEKB UPON CLEAR/PF3/PFIØ                    ***
***                                                                ***
***                                                                ***
***  REGISTER USAGE TABLE                                          ***
***                                                                ***
***  R3  - BASE REG  (DEFAULT)                                     ***
***  R4  - BASE REG  FOR DETAIL LINE                               ***
***  R5  - BASE REG  FOR CONNECTION TABLE                          ***
***  RIØ - LOOP COUNTER                                            ***
***  R11 - EIB REG   (DEFAULT)                                     ***
***  R13 - DATA REG  (DEFAULT)                                     ***
***                                                                ***
**********************************************************************
**********************************************************************
         COPY  DFHAID
         COPY  DFHBMSCA
STFIELD  EQU   X'1D'
NEWLINE  EQU   X'15'
         CVT   DSECT=YES
         IEESMCA
         IHAPSA
DFHEISTG DSECT
REQID    DS    ØCL8
REQTERM  DS    CL4
```

```
REQTRAN  DS    CL4
LINEØ1   DS    CL(LINEØ1L)
         ORG   LINEØ1
HEADERA  DS    CL2
LTRANSID DS    CL4
HEADERTR DS    CL28
SYSID    DS    CL4
         DS    CL1
APPLID   DS    CL8
         DS    CL4
SMFID    DS    CL4
         DS    CL5
HEADERT  DS    CL8         HH:MM:SS
         DS    CL2
HEADERD  DS    CL8         MM/DD/YY
         ORG
LINEØ2   DS    CL(3)
LINEØ3   DS    CL(LINEL)
LINEØ4   DS    CL(LINEL)
LINEØ5   DS    CL(LINEL)
LINEØ6   DS    CL(LINEL)
LINEØ7   DS    CL(LINEL)
LINEØ8   DS    CL(LINEL)
LINEØ9   DS    CL(LINEL)
LINE1Ø   DS    CL(LINEL)
LINE11   DS    CL(LINEL)
LINE12   DS    CL(LINEL)
LINE13   DS    CL(LINEL)
LINE14   DS    CL(LINEL)
LINE15   DS    CL(LINEL)
LINE16   DS    CL(LINEL)
LINE17   DS    CL(LINEL)
LINE18   DS    CL(LINEL)
LINE19   DS    CL(LINEL)
LINE2Ø   DS    CL(LINEL)
LINE21   DS    CL(LINEL)
LINE22   DS    CL(LINEL)
LINE23   DS    CL(LINEL)
LINE24   DS    CL(LINEL)
SCREENL  EQU   *-LINEØ1
CONNØ1   DS    CL24
CONNØ2   DS    CL24
CONNØ3   DS    CL24
CONNØ4   DS    CL24
CONNØ5   DS    CL24
CONNØ6   DS    CL24
CONNØ7   DS    CL24
CONNØ8   DS    CL24
CONNØ9   DS    CL24
```

```
CONN1Ø   DS    CL24
CONN11   DS    CL24
CONN12   DS    CL24
CONN13   DS    CL24
CONN14   DS    CL24
CONN15   DS    CL24
CONN16   DS    CL24
CONN17   DS    CL24
CONN18   DS    CL24
CONN19   DS    CL24
CONN2Ø   DS    CL24
CONN21   DS    CL24
CONN22   DS    CL24
RETURNC  DS    F
ABSTIME  DS    CL8
TACQST   DS    F
TERMID   DS    CL4
TNETNAME DS    CL8
CONNMON  DFHEIENT
         MVC   REQTERM,EIBTRMID     MOVE TERMID FOR REQUEST ID
         MVC   REQTRAN,EIBTRNID     MOVE TRANID FOR REQUEST ID
         EXEC  CICS CANCEL TRANSID(EIBTRNID) REQID(REQID) RESP(RETURNC)
         CLI   EIBAID,DFHCLEAR      IS THIS CLEAR?
         BE    RETURN               YES, RETURN AND END
         CLI   EIBAID,DFHPF3        IS THIS PF3?
         BE    RETURN               YES, RETURN AND END
         CLI   EIBAID,DFHPF1Ø       IS THIS PF1Ø?
         BE    RETURN               YES, RETURN AND END
         CLI   EIBAID,DFHPF15       IS THIS PF15?
         BE    RETURN               YES, RETURN AND END
         CLI   EIBAID,DFHPF22       IS THIS PF22?
         BE    RETURN               YES, RETURN AND END
         MVC   LINEØ1,LINEØ1I
         MVC   LINEØ2,LINEØ2I
         MVC   LINEØ3,LINEØ2I
         MVC   LINEØ4,LINEØ2I
         MVC   LINEØ5,LINEØ2I
         MVC   LINEØ6,LINEØ2I
         MVC   LINEØ7,LINEØ2I
         MVC   LINEØ8,LINEØ2I
         MVC   LINEØ9,LINEØ2I
         MVC   LINE1Ø,LINEØ2I
         MVC   LINE11,LINEØ2I
         MVC   LINE12,LINEØ2I
         MVC   LINE13,LINEØ2I
         MVC   LINE14,LINEØ2I
         MVC   LINE15,LINEØ2I
         MVC   LINE16,LINEØ2I
         MVC   LINE17,LINEØ2I
         MVC   LINE18,LINEØ2I
```

```
         MVC   LINE19,LINEØ2I
         MVC   LINE2Ø,LINEØ2I
         MVC   LINE21,LINEØ2I
         MVC   LINE22,LINEØ2I
         MVC   LINE23,LINEØ2I
         MVC   LINE24,LINEØ2I
         MVC   CONNØ1,CONNØ1I
         MVC   CONNØ2,CONNØ2I
         MVC   CONNØ3,CONNØ3I
         MVC   CONNØ4,CONNØ4I
         MVC   CONNØ5,CONNØ5I
         MVC   CONNØ6,CONNØ6I
         MVC   CONNØ7,CONNØ7I
         MVC   CONNØ8,CONNØ8I
         MVC   CONNØ9,CONNØ9I
         MVC   CONN1Ø,CONN1ØI
         MVC   CONN11,CONN11I
         MVC   CONN12,CONN12I
         MVC   CONN13,CONN13I
         MVC   CONN14,CONN14I
         MVC   CONN15,CONN15I
         MVC   CONN16,CONN16I
         MVC   CONN17,CONN17I
         MVC   CONN18,CONN18I
         MVC   CONN19,CONN19I
         MVC   CONN2Ø,CONN2ØI
         MVC   CONN21,CONN21I
         MVC   CONN22,CONN22I
*
*        MVC   EIBTRNID,LTRANSID
         USING PSA,Ø
         L     1,FLCCVT
         USING CVT,1
         L     1,CVTSMCA
         USING SMCABASE,1
         MVC   SMFID,SMCASID   MOVE IN SMF ID
         DROP  1
         EXEC  CICS ASSIGN SYSID(SYSID) APPLID(APPLID)
         EXEC  CICS ASKTIME ABSTIME(ABSTIME)
         EXEC  CICS FORMATTIME ABSTIME(ABSTIME)                        X
               TIME(HEADERT) TIMESEP MMDDYY(HEADERD) DATESEP
         EXEC  CICS INQUIRE CONNECTION START RESP(RETURNC)
         LA    5,CONNØ1
         USING CONNDSCT,5
         LA    1Ø,22           INITIALIZE LOOP COUNTER
*
CONNLOOP DS    ØH
         EXEC  CICS INQUIRE CONNECTION(CONNNAME) NETNAME(NETNAME)      X
               CONNSTATUS(CONNSTA) SERVSTATUS(SERVSTA) NEXT            X
               RESP(RETURNC)
```

```
         CLC   RETURNC,DFHRESP(NORMAL)
         BNE   CONNDONE
         LA    5,CONNL(5)      NEXT ENTRY
         BCT   1Ø,CONNLOOP
*
CONNDONE DS    ØH
         EXEC  CICS INQUIRE CONNECTION END RESP(RETURNC)
*
         EXEC  CICS INQUIRE TERMINAL START
TERMINQL DS    ØH
         EXEC  CICS INQUIRE TERMINAL(TERMID) NEXT RESP(RETURNC)        X
               ACQSTATUS(TACQST) NETNAME(TNETNAME)
         CLC   RETURNC,DFHRESP(NORMAL) SEE IF NORMAL RETURN
         BNE   TERMINQE
*
         LA    1Ø,22           INTIALIZE LOOP COUNTER
         LA    5,CONNØ1        GO TO FIRST CONNECTION
TESTCONN DS    ØH
         CLC   NETNAME,TNETNAME
         BNE   NEXTCONN
         CLC   TACQST,DFHVALUE(ACQUIRED)
         BE    TSETACQ
         CLC   TACQST,DFHVALUE(RELEASED)
         BNE   TERMINQL
         AP    SESSRELC,=P'1'  ADD ONE TO RELEASED SESSION COUNTER
         B     TERMINQL
TSETACQ  DS    ØH
         AP    SESSACQC,=P'1'  ADD ONE TO ACQUIRED SESSION COUNTER
         B     TERMINQL
NEXTCONN DS    ØH
         LA    5,CONNL(5)      TEST NEXT CONN ENTRY
         BCT   1Ø,TESTCONN
         B     TERMINQL        NEXT TERMINAL
*
TERMINQE DS    ØH
         EXEC  CICS INQUIRE TERMINAL END
*
SENDTEXT DS    ØH
         LA    4,LINEØ3
         LA    5,CONNØ1
         USING LINEDSCT,4
         LA    1Ø,22           INITIALIZE LOOP COUNTER
LINELOOP DS   ØH
         CLI   CONNNAME,C' '   BLANKS?
         BE    ACQDONE         NO, SET UP LINE
DOCONN   DS    ØH
         MVC   LINEOUT(LINEL),LINEØ3I MOVE CONSTANTS
         MVC   CONNO,CONNNAME  MOVE CONNECTION NAME
         MVC   NETNO,NETNAME   MOVE NETNAME
         ED    SESSACNT,SESSACQC UNPACK ACQUIRED COUNT FOR CONNECTION
```

```
         ED    SESSRCNT,SESSRELC UNPACK RELEASED COUNT FOR CONNECTION
         MVC   INSSTAO,=CL3'OUT'
         CLC   SERVSTA,DFHVALUE(INSERVICE)
         BNE   SERVDONE
         MVC   INSSTAO,=CL3'INS'
SERVDONE DS    ØH
         MVC   ACQSTAO,=CL3'REL'
         CLC   CONNSTA,DFHVALUE(ACQUIRED)
         BNE   ACQDONE
         MVC   ACQSTAO,=CL3'ACQ'
ACQDONE  DS    ØH
         LA    4,LINEL(4)      DO NEXT LINE
         LA    5,CONNL(5)      DO NEXT CONNECTION
         BCT   1Ø,LINELOOP
*
         EXEC  CICS SEND TEXT FROM(LINEØ1) LENGTH(SENDL) ERASE FREEKB
*
RETURNE  DS    ØH
         EXEC  CICS START TRANSID(EIBTRNID) INTERVAL(ØØØ3ØØ)           X
               TERMID(EIBTRMID) REQID(REQID) RESP(RETURNC)
         EXEC  CICS RETURN TRANSID(EIBTRNID)
*
RETURN   DS    ØH
         EXEC  CICS SEND TEXT FROM(ENDLINE) LENGTH(ENDLINEL)           X
               ERASE FREEKB
         EXEC  CICS RETURN
*
* CONSTANTS
*
LINEØ1I  DS    CL(LINEØ1L)
         ORG   LINEØ1I
LINEØ1II DC    AL1(STFIELD,DFHBMASK)
         DC    CL4' '
         DC    CL28' ISC CONNECTION MONITOR'
         DC    CL4' '
         DC    CL1'/'
         DC    CL8' '
         DC    CL4' ON '
         DC    CL4' '
         DC    CL5' '
         DC    CL8' '     HH:MM:SS
         DC    CL2' '
         DC    CL8' '     MM/DD/YY
LINEØ1L  EQU   *-LINEØ1II
         ORG
LINEØ2I  DC    AL1(NEWLINE,STFIELD,DFHBMASK)
         DC    CL77' '
LINEØ3I  DS    CL(LINEL)
         ORG   LINEØ3I
         DC    AL1(NEWLINE,STFIELD,DFHBMASK)
```

```
         DC    CL4' ' CONN
         DC    CL1' '
         DC    CL8' ' NETNAME
         DC    CL2' '
         DC    CL3' ' INS STA
         DC    CL1' '
         DC    CL3' ' ACQ STA
         DC    CL18'    SESSIONS  ACQ: '
         DC    X'40Ø2Ø212Ø' ACQ CNT
         DC    CL5' REL:'
         DC    X'40Ø2Ø212Ø' REL CNT
         ORG
         DS    ØH
ENDLINEL DC    AL2(L'ENDLINE)
ENDLINE  DC    CL8Ø'     ENTER NEXT TRANSACTION'
SENDL    DC    AL2(SCREENL)
         DS    ØF
CONNØ1I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONNØ2I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONNØ3I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONNØ4I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONNØ5I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONNØ6I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONNØ7I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONNØ8I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONNØ9I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN1ØI  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN11I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN12I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN13I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN14I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN15I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN16I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN17I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN18I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN19I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN2ØI  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN21I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
CONN22I  DC    CL4' ',CL8' ',F'Ø',F'Ø',PL2'Ø',PL2'Ø'
*
LINEDSCT DSECT
LINEOUT  DS    CL3
CONNO    DS    CL4
         DS    CL1
NETNO    DS    CL8
         DS    CL2
INSSTAO  DS    CL3
         DS    CL1
ACQSTAO  DS    CL3
         DS    CL18
```

```
SESSACNT DS    CL4
         DS    CL5
SESSRCNT DS    CL4
LINEL    EQU   *-LINEOUT
*
         DS    ØF
CONNDSCT DSECT
CONNNAME DS    CL4
NETNAME  DS    CL8
CONNSTA  DS    F
SERVSTA  DS    F
SESSACQC DS    PL2
SESSRELC DS    PL2
CONNL    EQU   *-CONNNAME
         END
//*
//ASM.SYSLIB   DD
//             DD
//             DD
//             DD
//             DD DISP=SHR,DSN=SYS1.MODGEN
//LKED.SYSLMOD DD DISP=SHR,DSN=CICS.PROGRAM.LOADLIB(COMMON)
```

CONNACQR

While the problem of sessions being released was being researched, we were having to release and re-acquire the sessions (because the sessions can't be acquired at session level, only at the connection level). We were having to monitor the connection and release/re-acquire manually even during the off-shift hours, and it was decided we had to automate the process. Program CONNACQR was written to perform this function.

Program CONNACQR is added to DFHPLTPI to be executed at CICS intialization, and starts the transaction named in label TRANSID (currently with content of CL4'ACQR') in three minutes. When the transaction starts, it checks for the connection/netname named in labels CONN/CONNNETN and tests the count of acquired sessions for that connection using the CICS SPI interface commands. If the count of acquired connections is less than two, then the connection is released and re-acquired (with a 5-second wait in between) also using CICS SPI interface commands.

```
//JOBCARD  JOB (CCS),'CSH CONNACQR  ',
//         CLASS=A,NOTIFY=&SYSUID,
```

```
//         MSGCLASS=X,MSGLEVEL=(1,1),REGION=8M
//*
//CONNACQR EXEC DFHEITAL,PARM.TRN='SP' COND.LKED=(Ø,LE)
//TRN.SYSIN  DD *
*        PRINT NOGEN
         TITLE 'CSH CONNECTION TESTING/REAQUIRE PROGRAM'
*********************************************************************
*********************************************************************
***                                                               ***
***  COUNTS THE ACQUIRED SESSIONS FOR A PARTICULAR ISC CONNECTION ***
***  DENOTED BY THE LABELS:                                       ***
***    CONN     DC    CL4'XXXX'                                   ***
***    CONNNETN DC    CL8'NNNNNNNN'                               ***
***  IN THE PROGRAM.  IF THE NUBMER OF SESSIONS FALLS BELOW TWO,  ***
***  THE CONNECTION IS RELEASED AND THEN ACQUIRED.                ***
***                                                               ***
***  THE PROGRAM IS STARTED VIA DFHPLTPI, STARTED AS TRANSID      ***
***  IN THE PROGRAM LABEL (CURRENTLY 'ACQR')                      ***
***                                                               ***
***    TRANSID  DC    CL4'ACQR'                                   ***
***                                                               ***
***  THE TRANSACTION IS STARTED AND ACTIVATES EVERY FIVE MINUTES, ***
***  TO DO THE TEST/RELEASE/ACQUIRE PROCESSING. THE PROGRAM WILL  ***
***  ALSO CHECK FOR CICS SHUTDOWN STATUS AND TERMINATE UPON CICS  ***
***  SHUTDOWN                                                     ***
***                                                               ***
***                                                               ***
***  REGISTER USAGE TABLE                                         ***
***                                                               ***
***  R3  - BASE REG  (DEFAULT)                                    ***
***  R11 - EIB REG   (DEFAULT)                                    ***
***  R13 - DATA REG  (DEFAULT)                                    ***
***                                                               ***
*********************************************************************
*********************************************************************
DFHEISTG DSECT
RETURNC  DS    F
STATUS   DS    F
TERMID   DS    CL4
NETNAME  DS    CL8
COUNT    DS    PL2
CONNACQR DFHEIENT
*
         CLC   EIBTRNID,TRANSID               ARE WE RUNNING ALREADY?
         BE    RUNXACT
         EXEC  CICS START TRANSID(TRANSID) INTERVAL(3ØØ) RESP(RETURNC)
         B     RETURNX
*
RUNXACT  DS    ØH
```

```
         EXEC  CICS INQUIRE SYSTEM SHUTSTATUS(STATUS) RESP(RETURNC)
         CLC   STATUS,DFHVALUE(NOTAPPLIC)
         BNE   RETURN
*
         ZAP   COUNT,=P'Ø'
         EXEC  CICS INQUIRE TERMINAL START RESP(RETURNC)
TERMINQL DS    ØH
         EXEC  CICS INQUIRE TERMINAL(TERMID) NEXT RESP(RETURNC)        X
               ACQSTATUS(STATUS) NETNAME(NETNAME)
         CLC   RETURNC,DFHRESP(NORMAL) SEE IF NORMAL RETURN
         BNE   TERMINQE
         CLC   NETNAME,CONNNETN
         BNE   TERMINQL
         CLC   STATUS,DFHVALUE(ACQUIRED)
         BNE   TERMINQL
         AP    COUNT,=P'1'     ADD ONE TO ACQUIRED SESSION COUNTER
         B     TERMINQL
*
TERMINQE DS    ØH
         EXEC  CICS INQUIRE TERMINAL END RESP(RETURNC)
         CP    COUNT,=P'1'
         BH    DELAY
*
         EXEC  CICS SET CONNECTION(CONN) RELEASED RESP(RETURNC)
         EXEC  CICS DELAY INTERVAL(ØØØØØ5)
         EXEC  CICS SET CONNECTION(CONN) INSERVICE ACQUIRED            X
               RESP(RETURNC)
         EXEC  CICS WRITE OPERATOR                                     X
               TEXT('*** ISC CONNECTION XXXX RELEASE/ACQUIRED')        X
               TEXTLENGTH(4Ø) RESP(RETURNC)
DELAY    DS    ØH
         EXEC  CICS DELAY INTERVAL(ØØØ455)
         B     RUNXACT
*
RETURN   DS    ØH
         EXEC  CICS WRITE OPERATOR                                     X
               TEXT('*** ISC CHECK/ACQUIRE TRANSACTION ENDED ')        X
               TEXTLENGTH(4Ø) RESP(RETURNC)
RETURNX  DS    ØH
         EXEC  CICS RETURN
*
TRANSID  DC    CL4'ACQR'
CONN     DC    CL4'XXXX'
CONNNETN DC    CL8'NNNNNNNN'
         END
//*
//LKED.SYSLMOD DD DISP=SHR,DSN=CICS.PROGRAM.LOADLIB(CONNACQR)
```

The transaction is a long-running one with the program waiting for

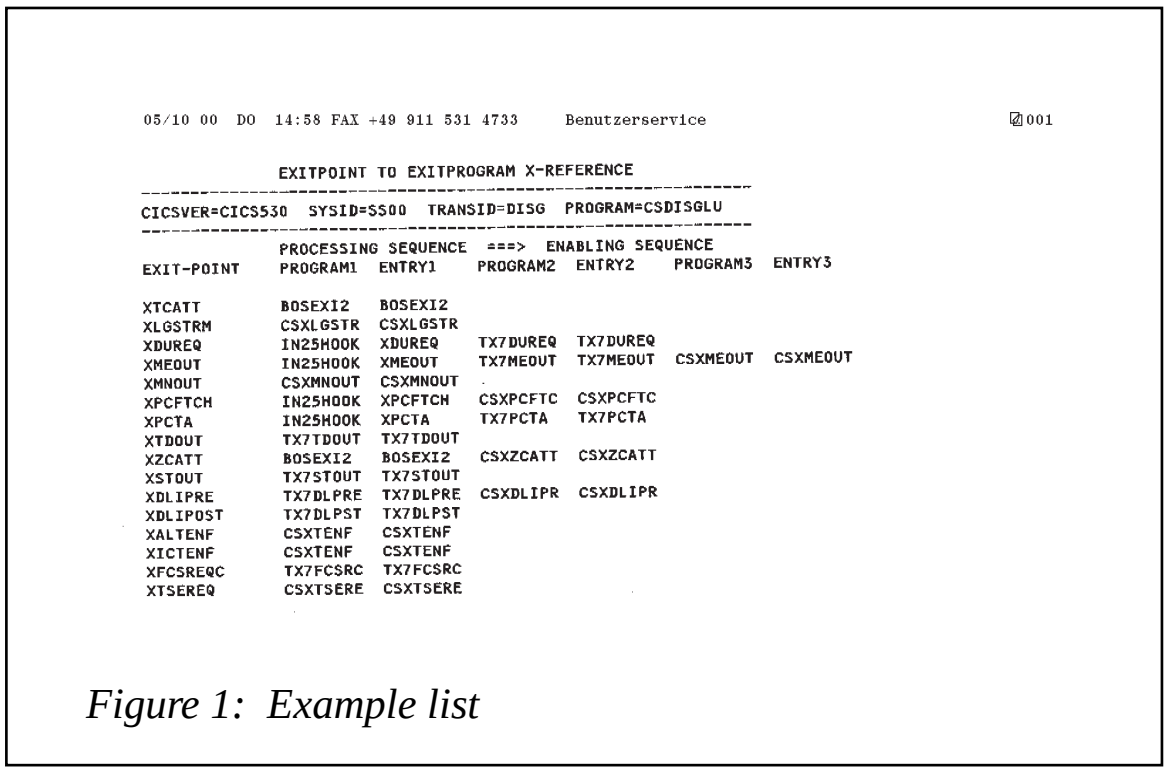
five minutes between checking the connection status and REL/ACQ actions. The program also checks for CICS shutdown status and terminates itself upon CICS termination.

Chorng S (Jack) Hwang
Principal
HSA Systems (USA)

List all installed global user exits

DESCRIPTION

With the program CSDISGLU you are able to list all installed GLUEs (global user exits) in a CICS region and you don't need an OEM product to do it. The transaction DISG calls program CSDISGLU, and you get the list shown in Figure 1, which changes depending on the installation on which the transaction is running:



```
05/10 00  DO  14:58 FAX +49 911 531 4733     Benutzerservice                    001

             EXITPOINT TO EXITPROGRAM X-REFERENCE
-------------------------------------------------------------------
CICSVER=CICS530  SYSID=SS00  TRANSID=DISG  PROGRAM=CSDISGLU
-------------------------------------------------------------------
             PROCESSING SEQUENCE  ===>  ENABLING SEQUENCE
EXIT-POINT   PROGRAM1  ENTRY1    PROGRAM2  ENTRY2    PROGRAM3  ENTRY3

XTCATT       BOSEXI2   BOSEXI2
XLGSTRM      CSXLGSTR  CSXLGSTR
XDUREQ       IN25HOOK  XDUREQ    TX7DUREQ  TX7DUREQ
XMEOUT       IN25HOOK  XMEOUT    TX7MEOUT  TX7MEOUT  CSXMEOUT  CSXMEOUT
XMNOUT       CSXMNOUT  CSXMNOUT
XPCFTCH      IN25HOOK  XPCFTCH   CSXPCFTC  CSXPCFTC
XPCTA        IN25HOOK  XPCTA     TX7PCTA   TX7PCTA
XTDOUT       TX7TDOUT  TX7TDOUT
XZCATT       BOSEXI2   BOSEXI2   CSXZCATT  CSXZCATT
XSTOUT       TX7STOUT  TX7STOUT
XDLIPRE      TX7DLPRE  TX7DLPRE  CSXDLIPR  CSXDLIPR
XDLIPOST     TX7DLPST  TX7DLPST
XALTENF      CSXTENF   CSXTENF
XICTENF      CSXTENF   CSXTENF
XFCSREQC     TX7FCSRC  TX7FCSRC
XTSEREQ      CSXTSERE  CSXTSERE
```

Figure 1: Example list

You need to define the transaction DISG in your PCT and the program CSDISGLU in your PPT. When you've done this, compile the program as a command-level program and run DISG

CSDISGLU

```
*ASM XOPTS(CICS,SP)
*************************************************************************
*  PROGRAMMENAME:  CSDISGLU                                             *
*  FUNCTION:       THIS IS A SIMPLE PROGRAM TO DISPLAY THE EXITPGMS     *
*                  BY NAME FOR A SPECIFIC EXIT-POINT IN THE ORDER       *
*                  OF ENABLING = ORDER OF PROCESSING SEQUENCE.          *
*                                                                       *
*  SUPPORTED CICS VERSION:                                              *
*                                                                       *
*  CICS 5.3.Ø                                                           *
*                                                                       *
*     THE TERMINAL OUTPUT LOOKS LIKE :                                  *
*                                                                       *
*                    EXITPOINT TO EXITPROGRAM X-REFERENCE               *
*     ----------------------------------------------------------------  *
*     CICSVER=CICS53Ø  SYSID=YYYY  TRANSID=TTTT  PROGRAM=PPPPPPP        *
*     ----------------------------------------------------------------  *
*                 PROCESSING SEQUENCE  ===>  ENABLING SEQUENCE          *
*     EXIT-POINT PROGRAM1 ENTRY1   PROGRAM2 ENTRY2   PROGRAM3 ENTRY3    *
*       NAME     PROGRAM  ENTRY                                         *
*       ....     .......  .......                                       *
*                                                                       *
*                                                                       *
*     NAME...       = EXIT POINT NAME                                   *
*     PROGRAM1-5    = PROGRAM NAME AT EXIT POINT                        *
*                     THE SEQUENCE GIVES THE ENABLE SEQUENCE, WHICH     *
*                     IS THE PROCESSING SEQUENCE TOO.                   *
*     ENTRY1-3      = ENTRYPOINT NAME IF ENABLED WITH ENTRYNAME          *
*                     IF NOT ENABLED WITH ENTRYNAME IT DEFAULTS TO      *
*                     THE PROGRAM NAME.                                 *
*************************************************************************
*   EXPAND THE DFHEISTG FOR THE REQUIRED USER FIELDS
*
DFHEISTG DSECT
*************************************************************************
*    THE TERMINAL PRESENTATION OUTPUT AREA BEGINS HERE
*************************************************************************
OUT      DS    ØCL18ØØ
*********************  HEADER LINE
HDRMSG   DS    XL79
NL       DS    XL1
FILLØ    DS    XL62
```

```
NLØ      DS    XL1
************************  ENVIRONMENT LINE
CICSVER  DS    CL8
CICS     DS    CL7
SYSIDENT DS    CL8
SYSID    DS    CL4
TRANID   DS    CL1Ø
TRANSID  DS    CL4
PROGNA   DS    CL1Ø
PROGRAM  DS    CL8
NL1      DS    XL1
************************  STARTER STARS
FILL1    DS    XL62
NL2      DS    XL1
************************  1ST LINE OF DATA OUTPUT
FILLER1  DS    CL14
EXPLAIN  DS    CL5Ø
NEWLINE  DS    XL1
************************  2ND LINE OF DATA OUTPUT
GLUE     DS    CL1Ø
FILL2    DS    CL4
PROGNAM1 DS    CL8
FILL3    DS    CL2
ENTRY1   DS    CL8
FILL4    DS    CL2
PROGNAM2 DS    CL8
FILL5    DS    CL2
ENTRY2   DS    CL8
FILL6    DS    CL2
PROGNAM3 DS    CL8
FILL7    DS    CL2
ENTRY3   DS    CL8
NL3      DS    XL1
LIST     DS    CL15ØØ
*************************************************************************
*    THE OUTPUT AREA ENDS HERE
*************************************************************************
         EJECT
*************************************************************************
*     HERE BEGINS THE CSECT ITSELF                                      *
*************************************************************************
CSDISGLU CSECT
CSDISGLU AMODE 31
CSDISGLU RMODE ANY
         B     START
PROGNAME DC    CL8'CSDISGLU'       SET
         DC    CL8'&SYSDATE'       EYECATCHER
         DC    CL8'&SYSTIME'       INFORMATION
*
BEGIN    DS    ØH
```

```
*********************************************************************
*   CICS BASIC CODING                                               *
*   BASEREGISTER IS R3                                              *
*********************************************************************
         L     12,X'21C'          ADDRESS CURRENT TCB
         L     12,X'DØ'(,12)      ADDRESS TCB EXTN
         L     12,X'14'(,12)      ADDRESS AFCX
         L     12,X'Ø8'(,12)      ADDRESS THE CSA
         L     5,X'C8'(,12)       ADDRESS THE CSAOPFLA
         L     5,X'1C8'(,5)       ADDRESS THE USER EXIT TABLE
*********************************************************************
CICS53Ø  DS    ØH
         CLI   X'9F'(12),X'53'    CICS 5.3.Ø ?
         BNE   ERRMSG1            NO OTHER SUPPORTED
         MVC   CICS,=C'CICS53Ø'   MOVE VER/REL INTO OUTPUT FIELD
         L     4,X'84'(,5)        LOAD POINTER TO LAST UETE
         LA    4,X'28'(,4)        POINT AFTER LAST UETE
*********************************************************************
*        GO OVER THE UET HEADER TO THE 1ST UETE                     *
*        THERE IS 1 UETE FOR EACH EXIT POINT                        *
*********************************************************************
         LA    1Ø,X'BØ'(,5)       POINT TO 1ST UETE
         LA    5,EXITTAB          POINT TO EXITTABLE
         LA    2,LIST             POINT TO OUTPUT AREA
*********************************************************************
*  THE UEPL IS IN UETE + X'1Ø' IN LENGTH OF X'18'                   *
*  THE FIRST EPB IS POINTED TO BY X'1Ø' IN UEPL                     *
*********************************************************************
SCANLOOP DS    ØH
         LA    7,X'1Ø'(,1Ø)       POINT TO UEPL WITHIN UETE
         L     6,X'1Ø'(,7)        FIRST EPB FROM EPL
         LTR   6,6                IS ANY PROG ENABLED AT THIS EXIT?
         BZ    NEXTUETE           NO, CHECK NEXT UETE
         MVI   Ø(2),X'15'         SET NL CHARACTER
         LA    2,X'1'(,2)         POINT TO NEXT BUFFER LOCATION
         MVC   Ø(8,2),Ø(5)        MOVE EXIT POINT NAME TO AREA
         LA    2,X'E'(,2)         POINT TO THE 1ST PROGNAME OUTFIELD
         MVC   Ø(8,2),X'3C'(6)    MOVE 1ST EXITPGM NAME TO OUTPUT AREA
         LA    2,X'A'(,2)         POINT TO THE 1ST ENTRYNAME OUTFIELD
         MVC   Ø(8,2),8(6)        MOVE 1ST ENTRY NAME TO OUTPUT AREA
         LA    2,X'A'(,2)         POINT TO THE NEXT PROGNAME OUTFIELD
*********************************************************************
* COUNT FOR 2 ADDITIONAL ENTRIES PER LINE, NO MORE POSSIBLE        *
*********************************************************************
         LA    9,X'2'
*********************************************************************
*  THERE IS 1 UEPL FOR EACH SUBSEQUENT ENABLED PROGRAM              *
*  THE NEXT UEPL IS POINTED TO BY X'4' IN UEPL                      *
*********************************************************************
NEXTUEPL DS    ØH
```

```
         L     7,X'4'(,7)         NEXT UEPL FROM CURRENT UEPL
         LTR   7,7                IS THERE A NEXT PROGRAM ENABLED ?
         BZ    NEXTUETE           NO, CHECK NEXT UETE
         L     8,X'10'(,7)        LOAD POINTER TO NEXT EPB FROM EPL
         MVC   0(8,2),X'3C'(8)    MOVE EXITPGM NAME TO OUTPUT AREA
         LA    2,X'A'(,2)         POINT TO NEXT ENTRYNAME OUTFIELD
         MVC   0(8,2),8(8)        MOVE ENTRY NAME TO OUTPUT AREA
         LA    2,X'A'(,2)         POINT TO NEXT PROGNAME OUTFIELD
         BCT   9,NEXTUEPL         CHECK NEXT UEPL
*                                 IF MORE THAN 3 ENTRIES PER LINE
*                                 GOTO NEXT UETE
*********************************************************************
NEXTUETE DS    0H
         LA    5,X'8'(,5)         POINT TO THE NEXT EXITTAB FIELD
         LA    10,X'28'(,10)      POINT TO NEXT UETE
         CR    10,4               END OF UETE REACHED ?
         BE    SENDLIST           YES, SEND OUTPUT LIST
         B     SCANLOOP           CHECK NEXT UETE
*********************************************************************
ERRMSG1  DS    0H
         EXEC CICS SEND TEXT FROM(MSGDAT1) ERASE FREEKB
         B     RETURN
*********************************************************************
SENDLIST DS    0H
         MVI   17(2),X'15'       MOVE FINAL NL TO BUFFER
         EXEC CICS ASSIGN SYSID(SYSID) PROGRAM(PROGRAM)
         MVC   TRANSID,EIBTRNID
         EXEC CICS SEND TEXT FROM(OUT) ERASE FREEKB PAGING
*********************************************************************
RETURN   DS    0H
         EXEC CICS RETURN
*********************************************************************
START    DS   0H
*********************************************************************
*     LOAD DFHEISTG USER FIELDS                                     *
*********************************************************************
         MVC   CICSVER,=C'CICSVER='
         MVC   SYSIDENT,=C'  SYSID= '
         MVC   TRANID,=C'  TRANSID='
         MVC   PROGNA,=C'  PROGRAM='
         MVC   GLUE,=C'EXIT-POINT'
         MVC   PROGNAM1,=C'PROGRAM1'
         MVC   ENTRY1,=C'ENTRY1  '
         MVC   PROGNAM2,=C'PROGRAM2'
         MVC   ENTRY2,=C'ENTRY2  '
         MVC   PROGNAM3,=C'PROGRAM3'
         MVC   ENTRY3,=C'ENTRY3  '
         MVC   EXPLAIN,EXTEXT
         MVI   NL,X'15'
         MVI   NL1,X'15'
```

```
         MVI   NL2,X'15'
         MVI   NL3,X'15'
         MVI   NEWLINE,X'15'
         MVI   FILLØ,C'-'
         MVC   FILLØ+1,FILLØ
         MVI   NLØ,X'15'
         MVC   FILL1,FILLØ
         MVC   HDRMSG,MSGDAT
         B     BEGIN
*************************************************************************
*        CONSTANTS                                                      *
*************************************************************************
MSGDAT   DC    CL79'              EXITPOINT TO EXITPROGRAM X-REFERENCE'
MSGDAT1  DC    CL8Ø'CICS VERSION/RELEASE NOT SUPPORTED !'
EXTEXT   DC    CL5Ø'PROCESSING SEQUENCE  ===>  ENABLING SEQUENCE'
*************************************************************************
*  THE EXIT SEQUENCE BELOW IS EQUIVALENT TO THE UETE SEQUENCE         *
*************************************************************************
EXITTAB  DS  ØF
EXIT1    DC   CL8'XTCIN   '
EXIT2    DC   CL8'XTCOUT  '
EXIT3    DC   CL8'XTCATT  '
EXIT4    DC   CL8'XTCTIN  '
EXIT5    DC   CL8'XTCTOUT '
EXIT6    DC   CL8'XDSBWT  '
EXIT7    DC   CL8'XDSAWT  '
EXIT8    DC   CL8'XLGSTRM '
EXIT9    DC   CL8'XDUREQ  '
EXIT1Ø   DC   CL8'XDUCLSE '
EXIT11   DC   CL8'XDUOUT  '
EXIT12   DC   CL8'XMEOUT  '
EXIT13   DC   CL8'XFCREQ  '
EXIT14   DC   CL8'XFCREQC '
EXIT15   DC   CL8'XTSPTOUT'
EXIT16   DC   CL8'XGMTEXT '
EXIT17   DC   CL8'XMNOUT  '
EXIT18   DC   CL8'XRCINIT '
EXIT19   DC   CL8'XRCINPT '
EXIT2Ø   DC   CL8'XICREQ  '
EXIT21   DC   CL8'XICEXP  '
EXIT22   DC   CL8'XISLCLQ '
EXIT23   DC   CL8'XPCFTCH '
EXIT24   DC   CL8'XPCHAIR '
EXIT25   DC   CL8'XPCTA   '
EXIT26   DC   CL8'XPCABND '
EXIT27   DC   CL8'XPCREQ  '
EXIT28   DC   CL8'XPCREQC '
EXIT29   DC   CL8'XTDREQ  '
EXIT3Ø   DC   CL8'XTDIN   '
EXIT31   DC   CL8'XTDOUT  '
```

```
EXIT32   DC   CL8'XTSQRIN '
EXIT33   DC   CL8'XTSQROU '
EXIT34   DC   CL8'XTSPTIN '
EXIT35   DC   CL8'XZCIN   '
EXIT36   DC   CL8'XZCOUT  '
EXIT37   DC   CL8'XZCATT  '
EXIT38   DC   CL8'XZCOUT1 '
EXIT39   DC   CL8'XXRSTAT '
EXIT4Ø   DC   CL8'XXDFA   '
EXIT41   DC   CL8'XXDFB   '
EXIT42   DC   CL8'XXDTO   '
EXIT43   DC   CL8'XSTOUT  '
EXIT44   DC   CL8'XDLIPRE '
EXIT45   DC   CL8'XDLIPOST'
EXIT46   DC   CL8'XFCSREQ '
EXIT47   DC   CL8'XEIIN   '
EXIT48   DC   CL8'XEIOUT  '
EXIT49   DC   CL8'XALTENF '
EXIT5Ø   DC   CL8'XICTENF '
EXIT51   DC   CL8'XDTAD   '
EXIT52   DC   CL8'XDTRD   '
EXIT53   DC   CL8'XDTLC   '
EXIT54   DC   CL8'XSTERM  '
EXIT55   DC   CL8'XSRAB   '
EXIT56   DC   CL8'XFCSREQC'
EXIT57   DC   CL8'XSZBRQ  '
EXIT58   DC   CL8'XSZARQ  '
EXIT59   DC   CL8'XISCONA '
EXIT6Ø   DC   CL8'XRSINDI '
EXIT61   DC   CL8'XXMATT  '
EXIT62   DC   CL8'XZIQUE  '
EXIT63   DC   CL8'XTSEREQ '
EXIT64   DC   CL8'XTSEREQC'
EXIT65   DC   CL8'XTDEREQ '
EXIT66   DC   CL8'XTDEREQC'
EXIT67   DC   CL8'XICEREQ '
EXIT68   DC   CL8'XICEREQC'
EXIT69   DC   CL8'XALCAID '
EXIT7Ø   DC   CL8'XSNON   '
EXIT71   DC   CL8'XSNOFF  '
EXIT72   DC   CL8'XRMIIN  '
EXIT73   DC   CL8'XRMIOUT '
EXIT74   DC   CL8'XAKUSER '
EXIT75   DC   CL8'XFCNREC '
EXIT76   DC   CL8'XFCBFAIL'
EXIT77   DC   CL8'XFCLDEL '
EXIT78   DC   CL8'XFCBOVER'
EXIT79   DC   CL8'XFCBOUT '
EXIT8Ø   DC   CL8'XFCVSDS '
EXIT81   DC   CL8'XFCQUIS '
```

```
EXIT82   DC   CL8'XDUREQC '
EXIT83   DC   CL8'XFCAREQ '
EXIT84   DC   CL8'XFCAREQC'
EXIT85   DC   CL8'XEISPIN '
EXIT86   DC   CL8'XEISPOUT'
EXIT87   DC   CL8'XNQEREQ '
EXIT88   DC   CL8'XNQEREQC'
EXIT89   DC   CL8'XFAINTU '
EXIT9Ø   DC   CL8'XBMIN'
EXIT91   DC   CL8'XBMOUT'
EXIT92   DC   CL8'XBADEACT'
EXIT93   DC   CL8'XLDLOAD'
EXIT94   DC   CL8'XLDELETE'
EXIT95   DC   CL8'XINDT1'
EXIT96   DC   CL8'XINDT2'
EXIT97   DC   CL8'XLGWBC'
         EJECT
         LTORG
         SPACE
         END CSDISGLU
```

Claus Reis
CICS Systems Programmer
Nuernberger Lebensversicherung AG (Germany)

Extract data from CSD to manage CICS resources from a DB2 environment

The CICS system definition utility program DFHCSDUP is a component of Resource Definition Online, and it's an off-line utility program that allows you to read from and/or write to a CICS System Definition (CSD) file.

This utility program supplies various functions in order to manage CICS resources. Using the different functions, it is possible to extract requested data from a CSD file and pass it to a named user program for processing.

The extracted data can then be processed in various ways. You can obtain a file to load into DB2 tables with the objective of:

- Maintaining the history of CICS resources.

- Having one simple interface in order to carry out a query about CICS resources with several selection criteria (QMF interface).
- Creating reports of various types:
  - For later editing and commenting to document CSD resources.
  - As a sample back-up copy of the CSD VSAM file.

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

The command EXTRACT of the DFHCSDUP utility program causes the CSD data you select to be passed unformatted to a user program.

There is a CSD/DB2 formatting utility program to organize the CSD data passed to it from DFHCSDUP into a format suitable for the DB2 table load utility.

The data is organized into columns that correspond to the columns defined in the load utility's input.

Each selected resource causes a record to be written to this program's output file, with the first four characters identifying the resource type (eg MAPSet, PROGram, PROFile, TRANsaction, CONNection, SESSion, PARTitionset, TERMinal, TYPEterm, FILE, LSRPool, PARTner, TRCLass).

The program is DFH0FORC and it is in VS COBOL II.

It produces a sequential file with all the attributes of the definition resource of a CSD file.

The program DFH0FORC must be run against an EXTRACT command with the following format:

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

or:

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

The output file of the CICS CSD Extract function can then be processed in order to carry out the import to DB2 tables.

Only then is it possible to use the DB2 interface for management.

COMPILE AND LINK-EDITING EXTRACT USER PROGRAM

You must compile and link-edit the DFH0FORC user programs as batch programs, not as CICS applications.

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

- An ENTRY statement that defines the entry name as DFHEXTRA. Specify the entry name as DFHEXTRA, which is the entry name in the CICS supplied stub, DFHEXCI.
- An INCLUDE statement for a CICS supplied stub that must be included in your user program. Include DFHEXCI in any COBOL language user program that you write for use with the DFHCSDUP EXTRACT command. DFHEXCI is the interface stub between DFHCSDUP and the COBOL user program.
- Specify the COBOL routines in the INCLUDE statements.
- A CHANGE statement to change the dummy CSECT name in the CICS supplied stub from EXITEP to the name of your user program. The CICS supplied stub, DFHEXCI, is generated with a link to the user program using a dummy CSECT name (EXITEP). Use the link-edit CHANGE statement to change the CSECT name from EXITEP to the name specified on the PROGRAM-ID statement in the user program. The CSD/DB2 user program DFH0FORC uses the program-id FORMCSD.

You must use AMODE(24) and RMODE(24).

SAMPLE JOB TO COMPILE AND LINK-EDIT THE USER PROGRAM

```
//COMPILEJ JOB (LTYZ1100),
//         CLASS=S,
//         MSGCLASS=X,
//         MSGLEVEL=(1,1),
//         NOTIFY=&SYSUID
//*
//*-------------------------------------------------------------------*/
//*--------   Step to compile COBOL II  user program     ---------*/
//*-------------------------------------------------------------------*/
//*
//COB2    EXEC PGM=IGYCRCTL,
//   PARM='APOST,LIB,OFFSET,MAP,NOSEQ,BUFSIZE(30K),NORENT,NORES'
```

```
//*
//STEPLIB  DD DISP=SHR,DSN=COBII.COB2COMP
//SYSIN    DD DISP=SHR,DSN=USER.LIB.SOURCE(DFHØFORC)
//SYSLIB   DD DISP=SHR,DSN=USER.LIB.SOURCE
//SYSLIN   DD DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=VIO,
//         SPACE=(800,(1000,1000))
//SYSUDUMP DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSUT1   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//SYSUT2   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//SYSUT3   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//SYSUT4   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//SYSUT5   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//SYSUT6   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//SYSUT7   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//SYSUT8   DD SPACE=(800,(1000,1000),,,ROUND),UNIT=VIO
//*
//MODOBJ   EXEC PGM=IEBGENER
//SYSUT1    DD DSN=&&LOADSET,DISP=(OLD,PASS)
//SYSUT2    DD DSN=CICS.LIBRARY.USER.OBJ(DFHØFORC),DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//*
//*-------------------------------------------------------------------*/
//*--------   Step to link-edit COBOL II  user program    --------*/
//*-------------------------------------------------------------------*/
//*
//LKED    EXEC PGM=IEWL,
//   PARM='LIST,XREF,LET,AMODE(24),RMODE(24)'
//*
//SYSLIB   DD DISP=SHR,DSN=COBII.COB2LIB
//CICSLIB  DD DISP=SHR,DSN=CICS.SDFHLOAD
//OBJLIB   DD DISP=SHR,DSN=CICS.LIBRARY.USER.OBJ
//SYSLMOD  DD DISP=SHR,DSN=CICS.LIBRARY.USER.LOAD
//SYSUT1   DD SPACE=(1024,(50,50)),UNIT=VIO
//SYSPRINT DD SYSOUT=*
//COBLIB   DD DISP=SHR,DSN=COBII.COB2LIB
//SYSUDUMP DD SYSOUT=*
//SYSLIN   DD *
 ENTRY     DFHEXTRA
 CHANGE    EXITEP(FORMCSD)
 INCLUDE   CICSLIB(DFHEXCI)
 INCLUDE   SYSLIB(ILBOSRV)
 INCLUDE   SYSLIB(ILBOCMM)
 INCLUDE   SYSLIB(ILBOBEG)
 INCLUDE   OBJLIB(DFHØFORC)
 NAME      DFHØFORC(R)
/*
/*
//
```

PROGRAM DFH0FORC

```
*****************************************************************
* MODULE NAME = DFHØFORC                                        *
*****************************************************************
*   DB2 Formatting Sample Program                               *
*---------------------------------------------------------------*
*                                                               *
*  Descriptions.                                                *
*                                                               *
*  The function of this program is to format data passed to it  *
*  via the user exit in the off-line utility program DFHCSDUP,  *
*  into a format suitable for the DB2 table load utility. The   *
*  data is formatted into columns which correspond to those     *
*  defined in the load utility input. Each resource on the CSD  *
*  in the selected RDO groups results in a record in this       *
*  program's output file with the first 4 characters of the     *
*  record identifying the resource type (this is necessary so   *
*  that the load utility can determine which DB2 table to load  *
*  a records contents into).                                    *
*                                                               *
*  This program is designed to work with any EXTRACT command of *
*  the form:                                                    *
*             EXTRACT GROUP(<group name> OBJECTS                *
*      or                                                       *
*             EXTRACT LIST(<list name> OBJECTS                  *
*      where                                                    *
*           <group name> may be generic                         *
*           <list name> must not be generic                     *
*                                                               *
* NB  This program must be link edited with the stub DFHEXCI    *
*     and the VS COBOL II library subroutines ILBOSRV, ILBOBEG, *
*     and ILBOCMM.                                              *
*****************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. FORMCSD.
 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT DATAOUT
            ASSIGN TO SYSØØ2-DA-338Ø-S-FOROUT
            ORGANIZATION IS SEQUENTIAL
            FILE STATUS IS OUTPUT-FILE-STATUS
            ACCESS IS SEQUENTIAL.
 EJECT.
 DATA DIVISION.
 FILE SECTION.
 FD  DATAOUT
     RECORD CONTAINS 46Ø CHARACTERS
     BLOCK CONTAINS Ø RECORDS
```

```
     RECORDING MODE IS F
     LABEL RECORDS ARE OMITTED.
* ****************************** *
* Define the output record types *
* ****************************** *
 Ø1  PROGRAM-F        PIC X(46Ø).
 Ø1  TRANSACTION-F    PIC X(46Ø).
 Ø1  CONNECTION-F     PIC X(46Ø).
 Ø1  MAPSET-F         PIC X(46Ø).
 Ø1  PARTITIONSET-F   PIC X(46Ø).
 Ø1  PROFILE-F        PIC X(46Ø).
 Ø1  SESSIONS-F       PIC X(46Ø).
 Ø1  RDO-TERMINAL-F   PIC X(46Ø).
 Ø1  RDO-TYPETERM-F   PIC X(46Ø).
 Ø1  RDO-FILE-F       PIC X(46Ø).
 Ø1  RDO-LSRPOOL-F    PIC X(46Ø).
 Ø1  RDO-PARTNER-F    PIC X(46Ø).
 Ø1  RDO-TRANCLASS-F  PIC X(46Ø).
 EJECT.
* ************************************************************ *
*          D E F I N E  W O R K I N G  S T O R A G E           *
* ************************************************************ *
 WORKING-STORAGE SECTION.
 Ø1  EXIT-KEYWORD-VALUE-W.
     Ø2 KEYWORD-VALUE-CHAR        PIC X OCCURS 182 TIMES.
 Ø1  RESOURCE-DEFINITION-DATA.
     Ø2 RESOURCE-TYPE             PIC X(12).
     Ø2 RESOURCE-KEYWORD-VALUES   PIC X(53Ø).
 77  RESOURCE-DEFINITION-PTR      PIC 999.
 77  STRING-DELIMETER-1           PIC X VALUE HIGH-VALUES.
 77  STRING-DELIMETER-2           PIC X VALUE LOW-VALUES.
 77  MAX-KEYWORD-VALUE-LENGTH     PIC 999 COMP VALUE IS 182.
     88  KEYWORD-VALUE-LENGTH-OK  VALUES ARE 1 THRU 182.
 77  KEYWORD-COUNT                PIC 99.
 77  CMD-PART-1                   PIC X(1ØØ).
 77  OBJECTS-OPTION               PIC X(7).
     88  OBJECTS-NOT-SPECIFIED    VALUE IS SPACES.
* ************************************************************ *
* Define the error status flag and all return code values set by *
* this program.                                                  *
* ************************************************************ *
 77  OUTPUT-FILE-STATUS           PIC XX.
     88  ERROR-ON-OUTPUT-FILE     VALUES ARE 'Ø1' THRU '99'.
 77  OBJECTS-OPTION-MISSING       PIC S9999 COMP VALUE IS 1.
 77  OUTPUT-FILE-OPEN-ERROR       PIC S9999 COMP VALUE IS 2.
 77  OUTPUT-FILE-CLOSE-ERROR      PIC S9999 COMP VALUE IS 3.
 77  OUTPUT-FILE-WRITE-ERROR      PIC S9999 COMP VALUE IS 5.
 77  KEYWORD-VALUE-LENGTH-ERROR   PIC S9999 COMP VALUE IS 5.
 77  KEYWORD-TABLE-ERROR          PIC S9999 COMP VALUE IS 6.
 77  UNDEFINED-RESOURCE-TYPE      PIC S9999 COMP VALUE IS 7.
```

```
 77  OUTPUT-FILE-FLAG             PIC X.
     88  OUTPUT-FILE-OPEN         VALUE IS '1'.
 77  FILE-OPEN                    PIC X VALUE '1'.
 EJECT.
* ************************************************************ *
* Define the maximum number of keywords in each type of resource *
* DEFINITION (PLUS 1 FOR THE DEF-PADDING FIELD).               *
* ************************************************************ *
 77   PROGRAM-KEYWORD-COUNT       PIC 99 VALUE IS 15.
 77   TRANSACTION-KEYWORD-COUNT   PIC 99 VALUE IS 35.
 77   CONNECTION-KEYWORD-COUNT    PIC 99 VALUE IS 22.
 77   MAPSET-KEYWORD-COUNT        PIC 99 VALUE IS 6.
 77   PARTITIONSET-KEYWORD-COUNT  PIC 99 VALUE IS 6.
 77   PROFILE-KEYWORD-COUNT       PIC 99 VALUE IS 18.
 77   SESSIONS-KEYWORD-COUNT      PIC 99 VALUE IS 24.
 77   TERMINAL-KEYWORD-COUNT      PIC 99 VALUE IS 27.
 77   TYPETERM-KEYWORD-COUNT      PIC 99 VALUE IS 68.
 77   FILE-KEYWORD-COUNT          PIC 99 VALUE IS 34.
 77   LSRPOOL-KEYWORD-COUNT       PIC 99 VALUE IS 44.
 77   PARTNER-KEYWORD-COUNT       PIC 99 VALUE IS 7.
 77   TRANCLASS-KEYWORD-COUNT     PIC 99 VALUE IS 4.
* ************************************************************ *
* Define the area  where all the resource definitions are      *
* formatted into the appropriate columns before output. The    *
* first three fields of each record are:                       *
*             1) Resource type identifier eg TRAN, PROG etc    *
*             2) Resource name                                 *
*             3) RDO Group Name                                *
* The fields that follow are dependent on the resource type    *
* and the overlays of FORMATTED-AREA which follow define       *
* the formats of the various different resources. It is essential*
* that any change in the CEDA language is reflected by changes *
* to both the overlay definition and to the associated processing*
* procedure.                                                   *
* ************************************************************ *
 Ø1  FORMATTED-AREA.
     Ø5 RESOURCE-TYPE-F  PIC X(4).
     Ø5 RESOURCE-NAME-F  PIC X(8).
     Ø5 RDO-GROUP-NAME-F PIC X(8).
     Ø5 FORMATTED-DATA   PIC X(44Ø) USAGE IS DISPLAY.

*
*          P R O G R A M  D E F I N I T I O N
*
* Please Note - the length of PROGRAM-W must be equal to that
*               of FORMATTED-DATA.
*
 Ø5  PROGRAM-W REDEFINES FORMATTED-DATA.
     1Ø  PROGRAM-DEF.
         15 DESCRIPTION  PIC X(58)   USAGE IS DISPLAY.
```

```
          15 LANGUAGE     PIC X(9)    USAGE IS DISPLAY.
          15 RDO-RELOAD   PIC X(3)    USAGE IS DISPLAY.
          15 RESIDENT     PIC X(3)    USAGE IS DISPLAY.
          15 RDO-USAGE    PIC X(9)    USAGE IS DISPLAY.
          15 USELPACOPY   PIC X(3)    USAGE IS DISPLAY.
          15 STATUZ       PIC X(8)    USAGE IS DISPLAY.
          15 CEDF         PIC X(3)    USAGE IS DISPLAY.
          15 DATALOCATION PIC X(5)    USAGE IS DISPLAY.
          15 EXECKEY      PIC X(4)    USAGE IS DISPLAY.
          15 REMOTESYSTEM PIC X(4)    USAGE IS DISPLAY.
          15 REMOTENAME   PIC X(8)    USAGE IS DISPLAY.
          15 TRANSID      PIC X(4)    USAGE IS DISPLAY.
          15 EXECUTIONSET PIC X(9)    USAGE IS DISPLAY.
          15 DEF-PADDING  PIC X(310)  USAGE IS DISPLAY.
 EJECT.
*
*          T R A N S A C T I O N  D E F I N I T I O N
*
* Please Note - the length of TRANSACTION-W must be equal
*               to that of FORMATTED-DATA.
*
 05  TRANSACTION-W REDEFINES FORMATTED-DATA.
     10  TRANSACTION-DEF.
          15 DESCRIPTION  PIC X(58)   USAGE IS DISPLAY.
          15 PROGRM       PIC X(8)    USAGE IS DISPLAY.
          15 TWASIZE      PIC X(5)    USAGE IS DISPLAY.
          15 PROFILE      PIC X(8)    USAGE IS DISPLAY.
          15 PARTITIONSET PIC X(8)    USAGE IS DISPLAY.
          15 STATUZ       PIC X(8)    USAGE IS DISPLAY.
          15 TASKDATALOC  PIC X(5)    USAGE IS DISPLAY.
          15 TASKDATAKEY  PIC X(4)    USAGE IS DISPLAY.
          15 STORAGECLEAR PIC X(3)    USAGE IS DISPLAY.
          15 RUNAWAY      PIC X(7)    USAGE IS DISPLAY.
          15 SHUTDOWN     PIC X(8)    USAGE IS DISPLAY.
          15 ISOLATE      PIC X(3)    USAGE IS DISPLAY.
          15 RDO-DYNAMIC  PIC X(3)    USAGE IS DISPLAY.
          15 REMOTESYSTEM PIC X(4)    USAGE IS DISPLAY.
          15 REMOTENAME   PIC X(8)    USAGE IS DISPLAY.
          15 TRPROF       PIC X(8)    USAGE IS DISPLAY.
          15 LOCALQ       PIC X(3)    USAGE IS DISPLAY.
          15 PRIORITY     PIC X(3)    USAGE IS DISPLAY.
          15 TRANCLASS    PIC X(8)    USAGE IS DISPLAY.
          15 ALIAS        PIC X(4)    USAGE IS DISPLAY.
          15 TASKREQ      PIC X(4)    USAGE IS DISPLAY.
          15 XTRANID      PIC X(8)    USAGE IS DISPLAY.
          15 TPNAME       PIC X(64)   USAGE IS DISPLAY.
          15 XTPNAME      PIC X(128)  USAGE IS DISPLAY.
          15 DTIMOUT      PIC X(4)    USAGE IS DISPLAY.
          15 INDOUBT      PIC X(7)    USAGE IS DISPLAY.
          15 RESTART      PIC X(3)    USAGE IS DISPLAY.
```

```
         15 SPURGE       PIC X(3)    USAGE IS DISPLAY.
         15 TPURGE       PIC X(3)    USAGE IS DISPLAY.
         15 DUMP         PIC X(3)    USAGE IS DISPLAY.
         15 RDO-TRACE    PIC X(3)    USAGE IS DISPLAY.
         15 CONFDATA     PIC X(3)    USAGE IS DISPLAY.
         15 RESSEC       PIC X(3)    USAGE IS DISPLAY.
         15 CMDSEC       PIC X(3)    USAGE IS DISPLAY.
         15 DEF-PADDING  PIC X(35)   USAGE IS DISPLAY.
 EJECT.
*
*          C O N N E C T I O N  D E F I N I T I O N
*
* Please Note - the length of CONNECTION-W must be equal to
*               that of FORMATTED-DATA.
*
 Ø5  CONNECTION-W REDEFINES FORMATTED-DATA.
     1Ø  CONNECTION-DEF.
         15 DESCRIPTION  PIC X(58)  USAGE IS DISPLAY.
         15 NETNAME      PIC X(8)   USAGE IS DISPLAY.
         15 INDSYS       PIC X(4)   USAGE IS DISPLAY.
         15 REMOTESYSTEM PIC X(4)   USAGE IS DISPLAY.
         15 REMOTENAME   PIC X(4)   USAGE IS DISPLAY.
         15 REMOTESYSNET PIC X(8)   USAGE IS DISPLAY.
         15 ACCESSMETHOD PIC X(8)   USAGE IS DISPLAY.
         15 PROTOCOL     PIC X(4)   USAGE IS DISPLAY.
         15 CONNTYPE     PIC X(8)   USAGE IS DISPLAY.
         15 SINGLESESS   PIC X(3)   USAGE IS DISPLAY.
         15 DATASTREAM   PIC X(8)   USAGE IS DISPLAY.
         15 RECORDFORMAT PIC X(2)   USAGE IS DISPLAY.
         15 QUEUELIMIT   PIC X(4)   USAGE IS DISPLAY.
         15 MAXQTIME     PIC X(4)   USAGE IS DISPLAY.
         15 AUTOCONNECT  PIC X(3)   USAGE IS DISPLAY.
         15 INSERVICE    PIC X(3)   USAGE IS DISPLAY.
         15 SECURITYNAME PIC X(8)   USAGE IS DISPLAY.
         15 ATTACHSEC    PIC X(1Ø)  USAGE IS DISPLAY.
         15 BINDSECURITY PIC X(3)   USAGE IS DISPLAY.
         15 USEDFLTUSER  PIC X(3)   USAGE IS DISPLAY.
         15 PSRECOVERY   PIC X(1Ø)  USAGE IS DISPLAY.
         15 DEF-PADDING  PIC X(273) USAGE IS DISPLAY.
 EJECT.
*
*          M A P S E T  D E F I N I T I O N
*
* Please Note - the length of MAPSET-W must be equal to that
*               of FORMATTED-DATA.
*
 Ø5  MAPSET-W REDEFINES FORMATTED-DATA.
     1Ø  MAPSET-DEF.
         15 DESCRIPTION  PIC X(58)  USAGE IS DISPLAY.
         15 RESIDENT     PIC X(3)   USAGE IS DISPLAY.
```

```
         15 RDO-USAGE    PIC X(9)   USAGE IS DISPLAY.
         15 USELPACOPY   PIC X(3)   USAGE IS DISPLAY.
         15 STATUZ       PIC X(8)   USAGE IS DISPLAY.
         15 DEF-PADDING  PIC X(359) USAGE IS DISPLAY.
 EJECT.
*
*          P A R T I T I O N S E T  D E F I N I T I O N
*
* Please Note - the length of PARTITIONSET-W must be equal to
*               that of FORMATTED-DATA.
*
 Ø5  PARTITIONSET-W REDEFINES FORMATTED-DATA.
     1Ø  PARTITIONSET-DEF.
         15 DESCRIPTION  PIC X(58)  USAGE IS DISPLAY.
         15 RESIDENT     PIC X(3)   USAGE IS DISPLAY.
         15 RDO-USAGE    PIC X(9)   USAGE IS DISPLAY.
         15 USELPACOPY   PIC X(3)   USAGE IS DISPLAY.
         15 STATUZ       PIC X(8)   USAGE IS DISPLAY.
         15 DEF-PADDING  PIC X(359) USAGE IS DISPLAY.
 EJECT.
*
*          P R O F I L E  D E F I N I T I O N
*
* Please Note - the length of PROFILE-W must be equal to
*               that of FORMATTED-DATA.
*
 Ø5  PROFILE-W REDEFINES FORMATTED-DATA.
     1Ø  PROFILE-DEF.
         15 DESCRIPTION  PIC X(58)  USAGE IS DISPLAY.
         15 SCRNSIZE     PIC X(9)   USAGE IS DISPLAY.
         15 UCTRAN       PIC X(3)   USAGE IS DISPLAY.
         15 MODENAME     PIC X(8)   USAGE IS DISPLAY.
         15 PRINTERCOMP  PIC X(3)   USAGE IS DISPLAY.
         15 JOURNAL      PIC X(2)   USAGE IS DISPLAY.
         15 MSGJRNL      PIC X(6)   USAGE IS DISPLAY.
         15 MSGINTEG     PIC X(3)   USAGE IS DISPLAY.
         15 ONEWTE       PIC X(3)   USAGE IS DISPLAY.
         15 PROTECT      PIC X(3)   USAGE IS DISPLAY.
         15 CHAINCONTROL PIC X(3)   USAGE IS DISPLAY.
         15 DVSUPRT      PIC X(7)   USAGE IS DISPLAY.
         15 INBFMH       PIC X(4)   USAGE IS DISPLAY.
         15 RAQ          PIC X(3)   USAGE IS DISPLAY.
         15 LOGREC       PIC X(3)   USAGE IS DISPLAY.
         15 NEPCLASS     PIC X(3)   USAGE IS DISPLAY.
         15 RTIMOUT      PIC X(4)   USAGE IS DISPLAY.
         15 DEF-PADDING  PIC X(315) USAGE IS DISPLAY.
 EJECT.
*
*          S E S S I O N  D E F I N I T I O N
*
```

```
* Please Note - the length of SESSIONS-W must be equal to that
*               of FORMATTED-DATA.
*
 Ø5  SESSIONS-W REDEFINES FORMATTED-DATA.
     1Ø  SESSIONS-DEF.
         15 DESCRIPTION  PIC X(58)  USAGE IS DISPLAY.
         15 CONNECTION   PIC X(4)   USAGE IS DISPLAY.
         15 SESSNAME     PIC X(4)   USAGE IS DISPLAY.
         15 NETNAMEQ     PIC X(8)   USAGE IS DISPLAY.
         15 MODENAME     PIC X(8)   USAGE IS DISPLAY.
         15 PROTOCOL     PIC X(4)   USAGE IS DISPLAY.
         15 MAXIMUM      PIC X(7)   USAGE IS DISPLAY.
         15 RECEIVEPFX   PIC X(2)   USAGE IS DISPLAY.
         15 RECEIVECOUNT PIC X(3)   USAGE IS DISPLAY.
         15 SENDPFX      PIC X(2)   USAGE IS DISPLAY.
         15 SENDCOUNT    PIC X(3)   USAGE IS DISPLAY.
         15 SENDSIZE     PIC X(5)   USAGE IS DISPLAY.
         15 RECEIVESIZE  PIC X(5)   USAGE IS DISPLAY.
         15 SESSPRIORITY PIC X(3)   USAGE IS DISPLAY.
         15 USERID       PIC X(8)   USAGE IS DISPLAY.
         15 AUTOCONNECT  PIC X(3)   USAGE IS DISPLAY.
         15 BUILDCHAIN   PIC X(3)   USAGE IS DISPLAY.
         15 USERAREALEN  PIC X(3)   USAGE IS DISPLAY.
         15 IOAREALEN    PIC X(11)  USAGE IS DISPLAY.
         15 RELREQ       PIC X(3)   USAGE IS DISPLAY.
         15 DISCREQ      PIC X(3)   USAGE IS DISPLAY.
         15 NEPCLASS     PIC X(3)   USAGE IS DISPLAY.
         15 RECOVOPTION  PIC X(11)  USAGE IS DISPLAY.
         15 DEF-PADDING  PIC X(276) USAGE IS DISPLAY.
 EJECT.
*
*         T E R M I N A L  D E F I N I T I O N
*
* Please Note - the length of RDO-TERMINAL-W must be equal
*               to that of FORMATTED-DATA.
*
 Ø5  RDO-TERMINAL-W REDEFINES FORMATTED-DATA.
     1Ø  RDO-TERMINAL-DEF.
         15 DESCRIPTION  PIC X(58)  USAGE IS DISPLAY.
         15 AUTINSTMODEL PIC X(4)   USAGE IS DISPLAY.
         15 AUTINSTNAME  PIC X(8)   USAGE IS DISPLAY.
         15 TYPETERM     PIC X(8)   USAGE IS DISPLAY.
         15 NETNAME      PIC X(8)   USAGE IS DISPLAY.
         15 RDO-CONSOLE  PIC X(3)   USAGE IS DISPLAY.
         15 RDO-CONSNAME PIC X(8)   USAGE IS DISPLAY.
         15 REMOTESYSTEM PIC X(4)   USAGE IS DISPLAY.
         15 REMOTENAME   PIC X(4)   USAGE IS DISPLAY.
         15 REMOTESYSNET PIC X(8)   USAGE IS DISPLAY.
         15 MODENAME     PIC X(8)   USAGE IS DISPLAY.
         15 PRINTER      PIC X(4)   USAGE IS DISPLAY.
```

```
          15 PRINTERCOPY  PIC X(3)   USAGE IS DISPLAY.
          15 ALTPRINTER   PIC X(4)   USAGE IS DISPLAY.
          15 ALTPRINTCOPY PIC X(3)   USAGE IS DISPLAY.
          15 POOL         PIC X(8)   USAGE IS DISPLAY.
          15 TASKLIMIT    PIC X(5)   USAGE IS DISPLAY.
          15 USERID       PIC X(8)   USAGE IS DISPLAY.
          15 NATLANG      PIC X(1)   USAGE IS DISPLAY.
          15 TRANSACTION  PIC X(4)   USAGE IS DISPLAY.
          15 TERMPRIORITY PIC X(3)   USAGE IS DISPLAY.
          15 INSERVICE    PIC X(3)   USAGE IS DISPLAY.
          15 SECURITYNAME PIC X(8)   USAGE IS DISPLAY.
          15 ATTACHSEC    PIC X(10)  USAGE IS DISPLAY.
          15 BINDSECURITY PIC X(3)   USAGE IS DISPLAY.
          15 USEDFLTUSER  PIC X(3)   USAGE IS DISPLAY.
          15 DEF-PADDING  PIC X(249) USAGE IS DISPLAY.
 EJECT.
*
*          T Y P E T E R M  D E F I N I T I O N
*
* Please Note - the length of TYPETERM-W must be equal to
*               that of FORMATTED-DATA.
*
* LOGMODECOM added by APAR PA15147
 05  TYPETERM-W REDEFINES FORMATTED-DATA.
     10  TYPETERM-DEF.
          15 DESCRIPTION  PIC X(58)  USAGE IS DISPLAY.
          15 DEVICE       PIC X(8)   USAGE IS DISPLAY.
          15 TERMMODEL    PIC X(1)   USAGE IS DISPLAY.
          15 SESSIONTYPE  PIC X(8)   USAGE IS DISPLAY.
          15 LDCLIST      PIC X(8)   USAGE IS DISPLAY.
          15 SHIPPABLE    PIC X(3)   USAGE IS DISPLAY.
          15 PAGESIZE     PIC X(7)   USAGE IS DISPLAY.
          15 ALTPAGE      PIC X(7)   USAGE IS DISPLAY.
          15 ALTSUFFIX    PIC X(1)   USAGE IS DISPLAY.
          15 FMHPARM      PIC X(3)   USAGE IS DISPLAY.
          15 OBOPERID     PIC X(3)   USAGE IS DISPLAY.
          15 AUTOPAGE     PIC X(3)   USAGE IS DISPLAY.
          15 DEFSCREEN    PIC X(7)   USAGE IS DISPLAY.
          15 ALTSCREEN    PIC X(7)   USAGE IS DISPLAY.
          15 APLKYBD      PIC X(3)   USAGE IS DISPLAY.
          15 APLTEXT      PIC X(3)   USAGE IS DISPLAY.
          15 AUDIBLEALARM PIC X(3)   USAGE IS DISPLAY.
          15 COLOR        PIC X(3)   USAGE IS DISPLAY.
          15 RDO-COPY     PIC X(3)   USAGE IS DISPLAY.
          15 DUALCASEKYBD PIC X(3)   USAGE IS DISPLAY.
          15 EXTENDEDDS   PIC X(3)   USAGE IS DISPLAY.
          15 HILIGHT      PIC X(3)   USAGE IS DISPLAY.
          15 KATAKANA     PIC X(3)   USAGE IS DISPLAY.
          15 LIGHTPEN     PIC X(3)   USAGE IS DISPLAY.
          15 MSRCONTROL   PIC X(3)   USAGE IS DISPLAY.
```

```
           15 OBFORMAT     PIC X(3)   USAGE IS DISPLAY.
           15 PARTITIONS   PIC X(3)   USAGE IS DISPLAY.
           15 PRINTADAPTER PIC X(3)   USAGE IS DISPLAY.
           15 PROGSYMBOLS  PIC X(3)   USAGE IS DISPLAY.
           15 VALIDATION   PIC X(3)   USAGE IS DISPLAY.
           15 FORMFEED     PIC X(3)   USAGE IS DISPLAY.
           15 HORIZFORM    PIC X(3)   USAGE IS DISPLAY.
           15 VERTICALFORM PIC X(3)   USAGE IS DISPLAY.
           15 TEXTKYBD     PIC X(3)   USAGE IS DISPLAY.
           15 TEXTPRINT    PIC X(3)   USAGE IS DISPLAY.
           15 QUERY        PIC X(4)   USAGE IS DISPLAY.
           15 OUTLINE      PIC X(3)   USAGE IS DISPLAY.
           15 SOSI         PIC X(3)   USAGE IS DISPLAY.
           15 BACKTRANS    PIC X(3)   USAGE IS DISPLAY.
           15 CGCSGID      PIC X(11)  USAGE IS DISPLAY.
           15 ASCII        PIC X(2)   USAGE IS DISPLAY.
           15 SENDSIZE     PIC X(5)   USAGE IS DISPLAY.
           15 RECEIVESIZE  PIC X(5)   USAGE IS DISPLAY.
           15 BRACKET      PIC X(3)   USAGE IS DISPLAY.
           15 LOGMODE      PIC X(8)   USAGE IS DISPLAY.
           15 LOGMODECOM   PIC X(3)   USAGE IS DISPLAY.
           15 ERRLASTLINE  PIC X(3)   USAGE IS DISPLAY.
           15 ERRINTENSIFY PIC X(3)   USAGE IS DISPLAY.
           15 ERRCOLOUR    PIC X(9)   USAGE IS DISPLAY.
           15 ERRHILIGHT   PIC X(9)   USAGE IS DISPLAY.
           15 AUTOCONNECT  PIC X(3)   USAGE IS DISPLAY.
           15 ATI          PIC X(3)   USAGE IS DISPLAY.
           15 TTI          PIC X(3)   USAGE IS DISPLAY.
           15 CREATESESS   PIC X(3)   USAGE IS DISPLAY.
           15 RELREQ       PIC X(3)   USAGE IS DISPLAY.
           15 DISCREQ      PIC X(3)   USAGE IS DISPLAY.
           15 NEPCLASS     PIC X(3)   USAGE IS DISPLAY.
           15 SIGNOFF      PIC X(6)   USAGE IS DISPLAY.
           15 XRFSIGNOFF   PIC X(7)   USAGE IS DISPLAY.
           15 ROUTEMSGS    PIC X(8)   USAGE IS DISPLAY.
           15 LOGONMSG     PIC X(3)   USAGE IS DISPLAY.
           15 BUILDCHAIN   PIC X(3)   USAGE IS DISPLAY.
           15 USERAREALEN  PIC X(3)   USAGE IS DISPLAY.
           15 IOAREALEN    PIC X(11)  USAGE IS DISPLAY.
           15 UCTRAN       PIC X(6)   USAGE IS DISPLAY.
           15 RECOVOPTION  PIC X(11)  USAGE IS DISPLAY.
           15 RECOVNOTIFY  PIC X(11)  USAGE IS DISPLAY.
           15 DEF-PADDING  PIC X(89)  USAGE IS DISPLAY.
       EJECT.
```

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

Espedito Morvillo
Systems Programmer (Italy)

IBM has announced Version 5.0 of Tivoli Decision Support for OS/390, which provides a centralized repository for access to enterprise-wide IT data for use in performance management, capacity management, service level management, and cost accounting.

New features include a Java GUI Console (Viewer), which provides automatic graph saving in batch reporting, report setting at the viewer level, graph type mix view, automated installation, and JDK migration. Usability and security have also been improved.

Additional platform support means there's now support for CICS TS Version 1.3, as well as DB2 Version 6.1, IMS Version 7.1, OS/400 Version 4.4 and Version 4.5, RMF for OS/390 Version 2.8, MQSeries up to Version 2.1, and also TCP/IP for OS/390.

As well as the base Tivoli Decision Support product, the new version also includes optional features including CICS, which collects and stores needed data, allows data collection from the CICS monitoring facility, provides statistical analysis and transaction information, and provides multiple reporting views.

For further information contact your local IBM representative.
URL: http://www.tivoli.com/products/index/decision_support/.

* * *

IBM has released Version 2.5 of VSE/ESA, replacing all previous versions and releases of VSE. There are security and interoperability upgrades to MQSeries Version 2.1, and there's new CICS Web Support and the CICS 3270 Bridge.

TCP/IP for VSE/ESA Version 1.4 adds new facilities required for e-business connectors, CICS Web Support, DB2 Server for VSE Version 7 Release 1, and VSE/POWER PNET.

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

* * *

B & M has announced products aimed at service management solutions for e-business and enterprise systems.

TDSLink comprises a series of intelligent agents, which monitor and administer the performance of OS/390 components. The suite includes operations and performance management solutions for CICS, SNA, TCP/IP, and MVS networks.

For further information contact:
B & M, Heath End House, West Street, Tadley, Berks RG26 3ST, UK.
Tel: (0118) 9811880.
URL: http://www.bmeurope.com.

* * *