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

© Xephon plc 2001
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.

© Xephon plc 2001. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs $36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

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:

<table>
<thead>
<tr>
<th>TRAN</th>
<th>TASKID</th>
<th>RESNAME</th>
<th>RESTYPE</th>
<th>TCA</th>
<th>STATUS</th>
<th>LUWID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEDF</td>
<td>0000165</td>
<td>DFHZARQ1</td>
<td>ZCIWAIT</td>
<td>0705F680</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>SNCB</td>
<td>0000160</td>
<td>DBUGUSER</td>
<td>EDF</td>
<td>00057680</td>
<td>SUSPENDED</td>
<td>OURNET.LUTP0008</td>
</tr>
<tr>
<td>CEMT</td>
<td>0000154</td>
<td>DFHZARQ1</td>
<td>ZCIWAIT</td>
<td>00054080</td>
<td>SUSPENDED</td>
<td>OURNET.LUTP0006</td>
</tr>
<tr>
<td>CSNE</td>
<td>0000024</td>
<td></td>
<td></td>
<td>0705F080</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>CSNC</td>
<td>0000023</td>
<td>MROQUEUE</td>
<td>CSNC</td>
<td>00078000</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>CSSY</td>
<td>0000022</td>
<td>SINGLE</td>
<td>KCCOMPAT</td>
<td>00056600</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>CSSY</td>
<td>0000021</td>
<td>DFHJ01A</td>
<td>JCTERMS</td>
<td>00057000</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>CSSY</td>
<td>0000020</td>
<td>SUBTASK</td>
<td>JCTERMS</td>
<td>00078600</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>CSSY</td>
<td>0000008</td>
<td></td>
<td>JCTERMN</td>
<td>00056000</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>CSSY</td>
<td>0000007</td>
<td>DFHAPTIX</td>
<td>ICEXPIRY</td>
<td>00055600</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>CSSY</td>
<td>0000006</td>
<td>DFHAPTIM</td>
<td>ICMI000</td>
<td>00055000</td>
<td>SUSPENDED</td>
<td>OURNET.CICSV41A</td>
</tr>
<tr>
<td>CSTP</td>
<td>0000004</td>
<td>DFHZDSP</td>
<td>TCP_NORM</td>
<td>0705E680</td>
<td>SUSPENDED</td>
<td>CICSV41A</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>DS_NUDGE</td>
<td>TIEXPIRY</td>
<td>0000000</td>
<td>SUSPENDED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>SMSYSTEM</td>
<td></td>
<td>SUSPENDED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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(10),=CL10' NOT FOUND'

BAL  R9,WRITE_RECORD_TO_SYSPRINT
CLOSE SYSPRINT
B  @FINISH

FOUND_CICS DS 0H
MVC  ARASID,ASCBASID

* GETMAIN AREA TO STORE INFO - R7 FOR TABLE POINTER *

GETMAIN RU, LV=16000
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, 0(.R6)  DFHDSANCHOR
LA  R6, 172(.R6)
L  R6, 0(.R6)  DTA

PROCESS DTA DS 0H
LR  R5, R6  STORE DTA
MVC 0(8, R7), 12(R6)  RESOURCE NAME
MVC 8(8, R7), 20(R6)  RESOURCE TYPE
MVC 16(1, R7), 60(R6)  TASK STATE
LA  R6, 120(.R6)
L  R6, 0(.R6)  XMTXN
LTR R6, R6  XMTXN PRESENT?
BZ  NEXT DTA
MVC 17(4, R7), 48(R6)  TASK NUMBER
MVC 21(4, R7), 60(R6)  TRANSID
LA  R6, 120(.R6)
L  R6, 0(.R6)  TCA
LTR R6, R6  TCA PRESENT?
BZ  NEXT DTA

PROCESS_TCA DS 0H
MVC 25(18, R7), 617(R6)  LUWID LENGTH(1 BYTE) AND LUWID
ST  R6, 43(.R7)  TCA ADDRESS
LA  R6, 220(.R6)
L  R6, 0(.R6)  CSA

PROCESS_CSA DS 0H
MVC  CSACDTA, 76(R6)  CURRENTLY DISPATCHED TCA

NEXT DTA DS 0H
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'CIUSJOB:   '
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 0H
CLC TASK_FC,16(R7)
BNE TASK_STATE_01
MVC OUTREC+47(9),=CL9'SUSPENDED'
B PROCESS_LUWID

TASK_STATE_01 DS 0H
CLC TASK_01,16(R7)
BNE TASK_STATE_03
MVC OUTREC+47(12),=CL12'NON EXECUTABLE'
B PROCESS_LUWID

TASK_STATE_03 DS 0H
CLC TASK_03,16(R7)
BNE TASK_STATE_04
MVC OUTREC+47(12),=CL12'DISPATCHABLE'
B PROCESS_LUWID

TASK_STATE_04 DS 0H
CLC TASK_04,16(R7)
BNE TASK_STATE_05
MVC OUTREC+47(12),=CL12'RUNNING'
B PROCESS_LUWID

TASK_STATE_05 DS 0H
CLC TASK_05,16(R7)
BNE TASK_STATE_FA
MVC OUTREC+47(12),=CL12'RESUMED'
B PROCESS_LUWID

TASK_STATE_FA DS 0H
CLC TASK_FA,16(R7)
BNE TASK_STATE_FD
MVC OUTREC+47(6),=CL6'PURGED'
B PROCESS_LUWID

TASK_STATE_FD DS 0H
CLC TASK_FD,16(R7)
BNE TASK_STATE_FE
MVC OUTREC+47(6),=CL6'UNUSED'
B PROCESS_LUWID

TASK_STATE_FE DS 0H
CLC TASK_FE,16(R7)
BNE TASK_STATE_XX
MVC OUTREC+47(5),=CL5'RESET'
B PROCESS_LUWID

TASK_STATE_XX DS 0H
MVC OUTREC+47(8),=CL8'?? ??'
L R5,16(R7)
R2D R5,OUTREC+50(2)

PROCESS_LUWID DS 0H
MVC OUTREC+60Ø(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+60
A     R4,LUWIDL

SPACE_LUWID DS ØH
  MVC    Ø(1,R4),=CL1', 'A    R4,F01
  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=16000,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    XL16'00'
ARASID  DS    H
KEKCB   DS    ØF
  DC    XL4'00006518'
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 XL8'0F0F0F0F0F0F0F'
HXTB&SYSNDX DC CL16'0123456789ABCDEF'
WHEX&SYSNDX DS F
DS C
WDSP&SYSNDX DS CL8'*******'
   DC CL1'.'
END&SYSNDX DS ØH
MEND
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='ØØ,DB9')
DB2CONN=YES
DSALIM=7M
EDSALIM=9ØM
GRPLIST=PIPP0
GMTEXT='Have a nice day with DYNASTY CICS ! ... and ... Good luck ! ;)'
MN=ON
MNCONV=NO
MNPER=ON
PLTPI=DY
PLTSD=YL
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   ¬
0001CED40044CED444444444444 CED4ØØØØØØØØØØ0051159
00033300000333000000000000 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
0001CCCD4CD4CCCD4CD4CCCD4CD44
00DC4CD4CDEE00000000000000000000B5730B3C0000
001468944404689444D15386900
005336848512380000000000000004A52E0910000
```

-------

8----+----9--

000000000000
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:

```plaintext
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:

```plaintext
//DYNASTY  PROC
//***********************************************************
//* AUTOMATIC RESTART MANAGEMENT
//***********************************************************
//CHECKGCD  EXEC PGM=CHECKGCD,PARM='CICS53'
//STEPLIB   DD DISP=SHR,DSN=linklib
//SYSPRINT  DD SYSOUT=*  
//SYSGOUT  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  <- Updated 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(4061).
FD PARM-INPUT
   RECORDING MODE IS F
   RECORD CONTAINS 80 CHARACTERS
   DATA RECORD DATA-INPUT.
Ø1 DATA-INPUT PIC X(80).
FD PARM-OUTPUT
   RECORDING MODE IS F
   RECORD CONTAINS 80 CHARACTERS
   DATA RECORD DATA-OUTPUT.
Ø1 DATA-OUTPUT PIC X(80).

WORKING-STORAGE SECTION.
Ø1 CTL-KEY-CX41 PIC X(28) VALUE '   _CTL     CTL             '.
   * WHAT YOU SEE    '    CTL     CTL             ' *
   * WHAT YOU WRITE X'0001CED40044CED4444444444444  *
   * WITH HEX ON      00033000033000000000000  *
Ø1 RMDM-KEY-TS13 PIC X(28) VALUE '   _DFHRMDM DFHRMDM_ANCHOR  '.
   * WHAT YOU SEE    '    DFHRMDM DFHRMDM_ANCHOR  ' *
   * WHAT YOU WRITE X'0001CCCDDCD4CCDDCD6DCCCDD44  *
   * WITH HEX ON      00146894440468944401538690  *
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(10).
   Ø3 GCD-KEYPOINT-CX41 PIC X(1).
      88 GCD-WARMKEYP-CX41 VALUE ' - '.
      * WHAT YOU SEE    ' - ' *
      * WHAT YOU WRITE X'6' *
      * WITH HEX ON      0  *
      88 GCD-EMERKEYP-CX41 VALUE ' . '.
      * WHAT YOU SEE    ' . ' *
      * WHAT YOU WRITE X'4' *
      * WITH HEX ON      0  *
 Scientist: I need to create a new variable named 'AGE' and assign it a value of 25. I will then use the `IF` statement to check if the variable 'AGE' is greater than 18. If it is, I will display 'You are eligible for voting.' If 'AGE' is not greater than 18, I will display 'You are not eligible for voting.'

```sql
SET @AGE = 25;

IF @AGE > 18 THEN
  SELECT 'You are eligible for voting.';
ELSE
  SELECT 'You are not eligible for voting.';
END IF;
```

Dr. Smith: Let's define a new variable named 'BMI' based on the user's weight and height. We will calculate the BMI using the formula `BMI = weight / (height * height)`. Next, we will use the `IF` statement to check if the BMI is less than 18.5, which indicates underweight. If the BMI is greater than or equal to 18.5, we will display 'You are not underweight.'

```sql
DECLARE @weight DECIMAL(10,2);
DECLARE @height DECIMAL(10,2);

SET @weight = 78.5;
SET @height = 1.7;

SET @BMI = @weight / (@height * @height);

IF @BMI < 18.5 THEN
  SELECT 'You are underweight.';
ELSE
  SELECT 'You are not underweight.';
END IF;
```
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) © Xephon 2001

Need help with a CICS problem or project?

Maybe we can help:

- If it’s on a topic of interest to other subscribers, we’ll commission an article on the subject, which we’ll publish in *CICS Update*, and which we’ll pay for – it won’t cost you anything.

- If it’s a more specialized, or more complex, problem, you can advertise your requirements (including one-off projects, freelance contracts, permanent jobs, etc) to the thousands of CICS professionals who visit *CICS Update*’s home page every month. This service is also free of charge.

Visit the *CICS Update* Web site, http://www.xephon.com/cicsupdate.html, and follow the link to **Suggest a topic** or **Opportunities for CICS specialists**.
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):

<table>
<thead>
<tr>
<th>CONNCTION NAME</th>
<th>NETNAME</th>
<th>SERVICE STATUS</th>
<th>INS/OUT</th>
<th>CONNECTIONS STATUS</th>
<th>ACQ/REL</th>
<th>ACQUIRED SESSION COUNT (FOR NON-XM)</th>
<th>RELEASED SESSION COUNT (FOR NON-XM)</th>
</tr>
</thead>
</table>

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 WITH THE FOLLOWING DATA:
***
***  CONNCTION NAME
***  NETNAME
***  SERVICE STATUS      INS/OUT
***  CONNECTION 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/PF1Ø
***
***
***  REGISTER USAGE TABLE
***
***
***  R3  - BASE REG (DEFAULT)
***  R4  - BASE REG FOR DETAIL LINE
***  R5  - BASE REG FOR CONNECTION TABLE
***  R10 - 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)
LINE10    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)
LINE20    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

CONN10 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
CONN20 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,DFHPF10 IS THIS PF10?
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 LINE01,LINE01I
MVC LINE02,LINE02I
MVC LINE03,LINE03I
MVC LINE04,LINE04I
MVC LINE05,LINE05I
MVC LINE06,LINE06I
MVC LINE07,LINE07I
MVC LINE08,LINE08I
MVC LINE09,LINE09I
MVC LINE10,LINE10I
MVC LINE11,LINE11I
MVC LINE12,LINE12I
MVC LINE13,LINE13I
MVC LINE14,LINE14I
MVC LINE15,LINE15I
MVC LINE16,LINE16I
MVC LINE17,LINE17I
MVC LINE18,LINE18I
MVC LINE19,LINE02I
MVC LINE20,LINE02I
MVC LINE21,LINE02I
MVC LINE22,LINE02I
MVC LINE23,LINE02I
MVC LINE24,LINE02I
MVC CONN01,CONN01I
MVC CONN02,CONN02I
MVC CONN03,CONN03I
MVC CONN04,CONN04I
MVC CONN05,CONN05I
MVC CONN06,CONN06I
MVC CONN07,CONN07I
MVC CONN08,CONN08I
MVC CONN09,CONN09I
MVC CONN10,CONN10I
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 CONN20,CONN20I
MVC CONN21,CONN21I
MVC CONN22,CONN22I

* MVC EIBTRNID,LTRANSID
USING PSA,0
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) TIME(HEADERT) TIMESEP MMDDYY(HEADERD) DATESEP
EXEC CICS INQUIRE CONNECTION START RESP(RETURNC)
LA 5,CONN01
USING CONNDSCT,5
LA 10,22 INITIALIZE LOOP COUNTER

* CONNLOOP DS 0H
EXEC CICS INQUIRE CONNECTION(CONNNAME) NETNAME(NETNAME) CONNSTATUS(CONNSTA) SERVSTATUS(SERVSTA) NEXT RESP(RETURNC)
CLC  RETURNC,DFHRESP(NORMAL)
BNE  CONNDONE
LA  5,CONN(5)       NEXT ENTRY
BCT  10,CONNLOOP

CONNDONE DS  0H
EXEC  CICS INQUIRE CONNECTION END RESP(RETURNC)

* EXEC  CICS INQUIRE TERMINAL START
TERMINQL DS  0H
EXEC  CICS INQUIRE TERMINAL(TERMID) NEXT RESP(RETURNC)        X
    ACQSTATUS(TACOST) NETNAME(TNETNAME)
CLC  RETURNC,DFHRESP(NORMAL) SEE IF NORMAL RETURN
BNE  TERMINQ

* LA  10,22       INITIALIZE LOOP COUNTER
LA  5,CONN01      GO TO FIRST CONNECTION
TESTCONN DS  0H
CLC  NETNAME,TNETNAME
BNE  NEXTCONN
CLC  TACOST,DFHVALUE(ACQUIRED)
BE  TSETACQ
CLC  TACOST,DFHVALUE(RELEASED)
BNE  TERMINQ
AP  SESSRELC,=P'1'  ADD ONE TO RELEASED SESSION COUNTER
B  TERMINQ

TSETACQ DS  0H
AP  SESSACQC,=P'1'  ADD ONE TO ACQUIRED SESSION COUNTER
B  TERMINQ

NEXTCONN DS  0H
LA  5,CONN(5)       TEST NEXT CONN ENTRY
BCT  10,TESTCONN
B  TERMINQ

* TERMINQ DS  0H
EXEC  CICS INQUIRE TERMINAL END

* SENDTEXT DS  0H
LA  4,LIN03
LA  5,CONN01
USING LINEDSCT,4
LA  10,22       INITIALIZE LOOP COUNTER

LINELOOP DS  0H
CLI  CONNNAME,C' '  BLANKS?
BE  ACQDONE      NO, SET UP LINE

DOCONN DS  0H
MVC  LINEOUT(LINEL),LIN03I MOVE CONSTANTS
MVC  CONN0,CONNNAME  MOVE CONNECTION NAME
MVC  NET0,NETNAME  MOVE NETNAME
ED  SESSACNT,SESSACQC UNPACK ACQUIRED COUNT FOR CONNECTION
ED SESSRNCT, SESSRELС UNPACK RELEASED COUNT FOR CONNECTION
MVC INSSTA0, = CL3'OUT'
CLC SERVSTA, DFHVALUE(INSERVICE)
BNE SERVDONE
MVC INSSTA0, = CL3'INS'
SERVDONE DS 0H
MVC ACOSTA0, = CL3'REL'
CLC CONNSTA, DFHVALUE(ACQUIRED)
BNE ACQDONE
MVC ACOSTA0, = CL3'ACQ'
ACQDONE DS 0H
LA 4, LINEL(4) DO NEXT LINE
LA 5, CONNL(5) DO NEXT CONNECTION
BCT 10, LINELoop

* EXEC CICS SEND TEXT FROM(LINE01) LENGTH(SENDL) ERASE FREEKB

* RETURN DS 0H
EXEC CICS START TRANSID(EIBTRNID) INTERVAL(000300) X
TERMID(EIBTRMID) REQID(REQID) RESP(RETURNNC)
EXEC CICS RETURN TRANSID(EIBTRNID)

* RETURN DS 0H
EXEC CICS SEND TEXT FROM(ENDLINE) LENGTH(ENDLINEL) X
ERASE FREEKB
EXEC CICS RETURN

* CONSTANTS

* LINE01I DS CL(LINE01L)
ORG LINE01I
LINE01II 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
LINE01L EQU *-LINE01II
ORG
LINE02I DC AL1(NEWLINE, STFIELD, DFHBMASK)
   DC CL77' '
LINE03I DS CL(LINEL)
ORG LINE03I
DC AL1(NEWLINE, STFIELD, DFHBMASK)
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 initialization, 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.

//JOB CARD 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 NUMBER 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
RETURNNC 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(300) RESP(RETURNNC)
     B RETURNX
     * 
RUNXACT DS ØH
The transaction is a long-running one with the program waiting for
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:

```text
05/10 00 DO 14:58 FAX +49 911 531 4733 Bonateerservice Q601
EXITPOINT TO EXITPROGRAM X-REFERENCE
CICSVER=CICS550 SYSPLEX=5500 TRANSID=DISG PROGRAM=CSDISGLU
EXIT-POINT PROCESSING SEQUENCE  ==>  ENABLING SEQUENCE
PROGRAM ENTRY  PROGRAM ENTRY  PROGRAM ENTRY  PROGRAM ENTRY
XTCATT  BASEXIZ  BASEXIZ
XLSTTRM  C3XL DSTR  C3XL DSTR
XBUYER  IN5H500K  XBUYER  TX7DURQ  TX7DURQ
XMDOUT  IN5H500K  XMDOUT  TX7MDOUT  TX7MDOUT  CSMMDOUT  CSMMDOUT
XMDOUT  CSMMDOUT  CSMMDOUT
XPCFTCH  IN5H500K  XPCFTCH  CSMPCFCT  CSMPCFCT
XPCFTCA  IN5H500K  XPCFTA  TX7PCFT  TX7PCFT
XTBOUT  TX7TBDOUT  TX7TBDOUT
XZCATT  BASEXIZ  BASEXIZ  CSMZCAT  CSMZCAT
XSTOUT  TX7TSTOUT  TX7TSTOUT
XSLIPRE  TX7BLPRE  TX7BLPRE  CSMXSLIPR  CSMXSLIPR
XRLPST  TX7BLPST  TX7BLPST
XNÝENF  CSMNÝENF  CSMNÝENF
XCTENF  CSMCTENF  CSMCTENF
XFCSENC  TX7PCSR  TX7PCSR
XTENF  CSMXTENF  CSMXTENF
```

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)

***********************************************************************
* PROGRAMNAME: 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 5.3.0                                                        *
*                                                                     *
* THE TERMINAL OUTPUT LOOKS LIKE:
*                                                                     *
*                   EXITPOINT TO EXITPROGRAM X-REFERENCE             *
*                                                                     *
* CICSVER=CICS530 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 = ENTRYPONT NAME IF ENABLED WITH ENTRYPNAME              *
* IF NOT ENABLED WITH ENTRYPNAME 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 CL10
TRANSID DS CL4
PROGNA DS CL10
PROGRAM DS CL8
NL1 DS XL1
************************ STARTER STARS
FILL1 DS XL62
NL2 DS XL1
************************ 1ST LINE OF DATA OUTPUT
FILLER1 DS CL14
EXPLAIN DS CL50
NEWLINE DS XL1
************************ 2ND LINE OF DATA OUTPUT
GLUE DS CL10
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 CL1500
******************************************************************************
* 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'D0'(,12)  ADDRESS TCB EXTN
L 12,X'14'(,12)  ADDRESS AFCX
L 12,X'08'(,12)  ADDRESS THE CSA
L 5,X'CB'(,12)  ADDRESS THE CSAOPFLA
L 5,X'1CB'(,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 10,X'B0'(,5)  POINT TO 1ST UETE
LA 5,EXITTAB  POINT TO EXITTABLE
LA 2,LIST  POINT TO OUTPUT AREA
********************************************************************
*  THE UEPL IS IN UETE + X'10' IN LENGTH OF X'18'               *
*  THE FIRST EPB IS POINTED TO BY X'10' IN UEPL                 *
********************************************************************
SCANLOOP DS ØH
LA 7,X'10'(,10)  POINT TO UEPL WITHIN UETE
L 6,X'10'(,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 EXTPGM 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
NEXTUETE DS ØH
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 ØH
EXEC CICS SEND TEXT FROM(MSGDAT1) ERASE FREEKB
B RETURN
******************************************************************************
SENDLIST DS ØH
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 ØH
EXEC CICS RETURN
******************************************************************************
START DS ØH
******************************************************************************
* 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 CL80'CICS VERSION/RELEASE NOT SUPPORTED !'
EXTEXT DC CL50'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'XTCTT  '
EXIT4 DC CL8'XTCTIN '
EXIT5 DC CL8'XTCTOUT '
EXIT6 DC CL8'XSDBWT '
EXIT7 DC CL8'XSLSAWT '
EXIT8 DC CL8'XLSGSTRM '
EXIT9 DC CL8'XDUEREQ '
EXIT10 DC CL8'XDUCLSE '
EXIT11 DC CL8'XDUOUT '
EXIT12 DC CL8'XMEOUT '
EXIT13 DC CL8'XFCREQ '
EXIT14 DC CL8'XFCREQC '
EXIT15 DC CL8'XTPSPTOUT'
EXIT16 DC CL8'XGMSHTEX '
EXIT17 DC CL8'XMNOUT '
EXIT18 DC CL8'XRGINIT '
EXIT19 DC CL8'XRGINPT '
EXIT20 DC CL8'XIGREQ '
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 '
EXIT30 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'
EXIT40 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'XFCSEREQ'
EXIT47 DC CL8'XEIIN'
EXIT48 DC CL8'XEIOUT'
EXIT49 DC CL8'XALTENF'
EXIT50 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'XFCSEQC'
EXIT57 DC CL8'XSZBRQ'
EXIT58 DC CL8'XSZARQ'
EXIT59 DC CL8'XISCONA'
EXIT60 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'
EXIT70 DC CL8'XSNON'
EXIT71 DC CL8'XSNOFF'
EXIT72 DC CL8'XRFMIIN'
EXIT73 DC CL8'XRMIOUT'
EXIT74 DC CL8'XAKUSER'
EXIT75 DC CL8'XFCNREC'
EXIT76 DC CL8'XFCBFAIL'
EXIT77 DC CL8'XFCVDEL'
EXIT78 DC CL8'XFCBOVER'
EXIT79 DC CL8'XFCBOUT'
EXIT80 DC CL8'XFCVSDS'
EXIT81 DC CL8'XFCQUIS'
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 DFHØFORC 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 DFHØFORC 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.
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

```plaintext
//COMPILEJ JOB (LTYZ1100),
//         CLASS=S,
//         MSGCLASS=X,
//         MSGLEVEL=(1,1),
//         NOTIFY=&SYSUID
.isNull
/*/-----------------------------------------------*/
//---------- 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.SDFHLDS
//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(ILBOCMX)
INCLUDE  SYSLIB(ILBOBEG)
INCLUDE  OBJLIB(DFHØFORC)
NAME     DFHØFORC(R)
*/
/*
*/
PROGRAM DFH0FORC

*****************************************************************
* MODULE NAME = DFH0FORC                                          *
*****************************************************************
* 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 record's 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-COMMISION.
   SELECT DATAOUT
   ASSIGN TO SYS02-DA-3380-S-FOROUT
   ORGANIZATION IS SEQUENTIAL
   FILE STATUS IS OUTPUT-FILE-STATUS
   ACCESS IS SEQUENTIAL.
EJECT.
DATA DIVISION.
FILE SECTION.
FD DATAOUT
   RECORD CONTAINS 460 CHARACTERS
   BLOCK CONTAINS 0 RECORDS
RECORDING MODE IS F
LABEL RECORDS ARE OMITTED.

* ************************************************************* *
* Define the output record types *
* ************************************************************* *
Ø1 PROGRAM-F PIC X(460).
Ø1 TRANSACTION-F PIC X(460).
Ø1 CONNECTION-F PIC X(460).
Ø1 MAPSET-F PIC X(460).
Ø1 PARTITIONSET-F PIC X(460).
Ø1 PROFILE-F PIC X(460).
Ø1 SESSIONS-F PIC X(460).
Ø1 RDO-TERMINAL-F PIC X(460).
Ø1 RDO-TYPETERM-F PIC X(460).
Ø1 RDO-FILE-F PIC X(460).
Ø1 RDO-LSRPOOL-F PIC X(460).
Ø1 RDO-PARTNER-F PIC X(460).
Ø1 RDO-TRANCLASS-F PIC X(460).
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(530).
77 RESOURCE-DEFINITION-PTR PIC 999.
77 STRING-DELIMITER-1 PIC X VALUE HIGH-VALUES.
77 STRING-DELIMITER-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(100).
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 '01' 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.
* 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(440) USAGE IS DISPLAY.
 *
* PROGRAM DEFINITION *
* 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.
*
* TRANSACTION DEFINITION
*
* Please Note - the length of TRANSACTION-W must be equal
to that of FORMATTED-DATA.
*
Ø5 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 TPNNAME 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 CONFDATA     PIC X(3)    USAGE IS DISPLAY.
15 DEF-PADDING  PIC X(35)   USAGE IS DISPLAY.
EJECT.
*
*
CONNECTION DEFINITION
*
*
* 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(10)  USAGE IS DISPLAY.
    15 BINDSECURITY PIC X(3) USAGE IS DISPLAY.
    15 USEDFLTUSER PIC X(3)  USAGE IS DISPLAY.
    15 PSRECOVERY PIC X(10)  USAGE IS DISPLAY.
    15 DEF-PADDING PIC X(273) USAGE IS DISPLAY.
EJECT.
*
*
MAPSET DEFINITION
*
*
* 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.
*
*  PARTITIONSET DEFINITION  *
*  Please Note - the length of PARTITIONSET-W must be equal to  *
*         that of FORMATTED-DATA.  *
*  *
Ø5 PARTITIONSET-W REDEFINES FORMATTED-DATA.
10 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.
*
*  PROFILE DEFINITION  *
*  Please Note - the length of PROFILE-W must be equal to  *
*         that of FORMATTED-DATA.  *
*  *
Ø5 PROFILE-W REDEFINES FORMATTED-DATA.
10 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 DVSPRTE PIC X(7) USAGE IS DISPLAY.
  15 INBFMH PIC X(4) USAGE IS DISPLAY.
  15 RAO 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.
*
*  SESSION DEFINITION  *
*
* Please Note - the length of SESSIONS-W must be equal to that of FORMATTED-DATA.

Ø5 SESSIONS-W REDEFINES FORMATTED-DATA.
  10 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.

* TERMINAL DEFINITION

* Please Note - the length of RDO-TERMINAL-W must be equal to that of FORMATTED-DATA.

Ø5 RDO-TERMINAL-W REDEFINES FORMATTED-DATA.
  10 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
Ø5 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 ALTSCREEN PIC X(7) USAGE IS DISPLAY.
   15 FHPARM 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 APLOYBD 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 HIGHLIGHT 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 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 ERRINTENSIFY PIC X(3) USAGE IS DISPLAY.
15 ERRLASTLINE PIC X(3) USAGE IS DISPLAY.
15 ERREXCEPTION PIC X(3) USAGE IS DISPLAY.
15 ERRHILIGHT PIC X(9) USAGE IS DISPLAY.
15 ATI 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 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)  © Xephon 2001
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.


For further information contact your local IBM representative.

** **

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.

** **