May 2001

In this issue

3 Cleaning up ‘orphaned’ temporary storage queues
9 Sending messages at CICS log-on
23 Displaying task storage use in a CICS region
33 Displaying a list of terminals
48 CICS news

© Xephon plc 2001
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, and complete issues in Acrobat PDF format, can be downloaded from our Web site at http://www.xephon.com/cicsupdate.html; you will need to supply a word from the printed issue.

Editor
Trevor Eddolls

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

Contributions
When Xephon is given copyright, articles published in CICS Update are paid for at the rate of £170 ($260) per 1000 words and £100 ($160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 ($80) per 100 lines. In addition, there is a flat fee of £30 ($50) per article. To find out more about contributing an article, without any obligation, please 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.
Cleaning up ‘orphaned’ temporary storage queues

BACKGROUND
CICS application developers are taught early on to delete temporary storage queues as soon as they are no longer needed. Those of us who support CICS environments may observe, to our chagrin, that this recommendation often goes unheeded. When a site favours MAIN temporary storage over AUX, the result can be much wasted ECDSA. This article describes a method to log the creation of temporary storage queues by terminalID and, assuming it is safe to do so, delete those queues when the user signs off from CICS.

LOGGING TS QUEUE CREATION
The temporary storage EXEC interface exit XTSEREQ provides a convenient mechanism to record the creation of temporary storage queues. Prior to the RDO TSMODEL capabilities provided in Transaction Server 1.3, XTSEREQ global user exits were commonly written to direct AUX temporary storage puts to MAIN or vice versa. For our purposes here, we will intercept TS WRITE operations and log TS queue creation in a user-maintained data table (UMT). The UMT is a logical choice because we anticipate relatively high volumes of data to which we need speedy direct access, and because the data is not useful beyond the life of the CICS region.

SOURCE FOR XTSEREQ

```
*ASM XOPTS(CICS,SP,NOEDF)
TITLE 'XTSEREQ GLOBAL USER EXIT'
PRINT ON,NOGEN
DFHUEXIT TYPE=EP,ID=XTSEREQ   gen standard user exit plist
DFHUEXIT TYPE=XPIENV          gen XPI interface
COPY DFHTSUED                 gen command level plist for TS
DFHEISTG DSECT
CMRDSP DS F
EXITRC DS F
CMDLEN DS H
CMRID DS CL20
CMDUSR DS CL8
```
COPY TSQLOG                      TSQLOG UMT record
GLUETSQ DFHEIENT
GLUETSQ AMODE 31
GLUETSQ RMODE ANY
LR R2,R1                       address GLUE plist
USING DFHUEPAR,R2
LA R15,UECRNORM
ST R15,EXITRNC
L R7,UEPEXN                    get exit id addr
CLI Ø(R7),XTSEREQ               is this our exit point?
BE CHKRECUR                    y - check recursion
LA R6,MSGBADXP                  n - point to msg
BAL R5,NOTIFY                   tell bad news
B GLUEEXIT                     return
CHKRECUR DS ØH
L R7,UEPPRECUR                 get recursion count addr
LH R7,Ø(R7)                    pick up counter
LTR R7,R7                      recursive call to exit?
BZ CHKTSREQ                    n - evaluate ts request
LA R6,MSGRECUR                 y - point to msg
BAL R5,NOTIFY                   tell bad news
B GLUEEXIT                     return
CHKTSREQ DS ØH
L R4,UEPCLPS                   address CLPS
USING TS_ADDR_LIST,R4           ... via reg 4
L R7,TS_ADDRØ                  address EID
USING TS_EID,R7                ... via reg 7
CLI TS_GROUP,TS_TEMPSTOR_GROUP  is this a ts request?
BE CHKTSPUT                    y - check tsq name
LA R6,MSGTSNOT                 n - point to msg
BAL R5,NOTIFY                   tell bad news
B GLUEEXIT                     return
CHKTSPUT DS ØH
CLI TS_FUNCT,TS_WRITEQ          are we talkin ts writeq?
BNE GLUEEXIT                   n - nothing to do
L R4,TS_ADDR1                  get tsqname addr

*-----------------------------------------------------------------------*
* Register 4 points to the tsqueue (8 bytes) or tsqname (16 bytes).
* This is the place for customized exclude logic. You don't want
* to log TSQ creation for any queues not eligible for deletion, so
* the code is along these lines:
*    CLC Ø(8,R4),=C'??????????'       can't delete this queue
*    BE GLUEEXIT                  so skip logging
*-----------------------------------------------------------------------*
LOG1 DS ØH
EXEC CICS ADDRESS EIB(R11)
OC EIBTRMID,EIBTRMID          associated termid?
BZ GLUEEXIT                 n - bypass logging
EXEC CICS ASSIGN            +
USERID(CMDUSR)           +
RESP(CMDRSP)              +
MVC  TSQLOG_TRMID,EIBTRMID
MVI  TSQLOG_TSQID,X'40'            blank out ts queue/qname
MVC  TSQLOG_TSQID+1(L'TSQLOG_TSQID-1),TSQLOG_TSQID
LA   R8,8                         presume ts queue
LA   R9,TSQLOG_TSQID
CLI  TS_EIDOPT5,TS_QNAME_X        test if ts qname
BNE  LOG2                        n - length is 8 bytes
SLA  R8,1(Ø)                     y - bump length to 16
LOG2  BCTR R8,RØ
EX   R8,LOGMVC                   log queue/qname
B    LOG3
LOGMVC MVC Ø(1,R9),Ø(R4)          <- Executed instruction
LOG3 MVC TSQLOG_DATE,EIBDATE      log date
MVC  TSQLOG_TIME,EIBTIME          ... time
MVC  TSQLOG_USER,CMDUSR           ... user id
MVC  CMDRID,TSQLOG_KEY
LA   R8,TSQLOG_REC_LEN
STH  R8,CMDLEN
EXEC  CICS WRITE                +
    FILE('TSQLOG')              +
    FROM(TSQLOG_RECORD)        +
    LENGTH(CMDLEN)            +
    RIDFLD(CMDRID)            +
    RESP(CMDRSP)              +
B     GLUEXIT
NOTIFY DS ØH
WTO   MF=(E,(R6))
BR    R5
*
GLUEXIT DS ØH                    API GLUE exit return
L    R15,EXITRC
DFHEIRET RCREG=15
*
Messages:
MSGBADXP WTO  'GLUETSQ - Invalid exit point',ROUTCDE=(14),MF=L
MSGRECUR WTO  'GLUETSQ - Called recursively',ROUTCDE=(14),MF=L
MSGTSNOT WTO  'GLUETSQ - Called for non-TS request',ROUTCDE=(14),MF=L
LTORG
END   GLUETSQ

SOURCE FOR TSQLOG DSECT

Note: include this file layout as member TSQLOG in a SYSLIB PDS.

TSQLOG_RECORD DS CL36
    ORG TSQLOG_RECORD
TSQLOG_KEY  DS CL20
    ORG TSQLOG_KEY
TSQLOG_TRMID DS CL4
TSQLOG_TSQID DS CL16
TSQLOG_DATE  DS PLØ4

RDO FILE DEFINITION FOR TSQLOG

Note: this definition can be repeated in as many regions as you employ the exit. Please adjust the Maxnumrecs parameter to meet your requirements.

CEDA View File( TSQLOG )
File : TSQLOG
Group : SUPPORT
Description : Base file for TSQLOG UMT
VSAM PARAMETERS
DSName : CICSTS.GLOBAL.BASE.TSQLOG
Password : PASSWORD NOT SPECIFIED
RLsaccess : No | Yes | No
LSrpoolid : 1-8 | None
READInteg : Uncommitted | Consistent | Repeatable
DNSSharing : Allreqs | Modifyreqs
STRings : 1-255
Nsrngroup :
REMOTE ATTRIBUTES
REMOTESystem :
REMOTEName :
REMOTE AND CFDATATABLE PARAMETERS
RECORDSize : 1-32767
Keylength : 1-255 (1-16 For CF Datatable)
INITIAL STATUS
STATUS : Enabled | Disabled | Unenabled
Opentime : Firstref | Startup
Disposition : Share | Old
BUFFERS
Databuffers : 2-32767
Indexbuffers : 1-32767
DATATABLE PARAMETERS
TABLE : User | No | CIcs | User | CF
Maxnumrecs : 1-99999999
CFDATATABLE PARAMETERS
Cfdtpool :
TABLEName :
UPDATEModel : Locking | Contention | Locking
Load : No | Yes
DATA FORMAT
RECORDFormat ==> V | F
OPERATIONS
Add ==> Yes | No | Yes
BRowse ==> Yes | Yes
DElete ==> Yes | Yes
READ ==> Yes | No
UPDATE ==> Yes | Yes
AUTO JOURNALLING
JOURNAL ==> No | 1-99
JNLRead ==> None | Updateonly | Readonly | All
JNLSYNCRead ==> No | Yes
JNLUpdate ==> No | Yes
JNLAdd ==> None | Before | After | ALL
JNLSYNCWrite ==> Yes | No
RECOVERY PARAMETERS
RECOVERY ==> None | Backoutonly | All
Fwdrecovlog ==> No | 1-99
BACkuptype ==> Static | Static | Dynamic
SECURITY
RESsecnum : ØØ | Ø-24 | Public

IDCAMS UTILITY STATEMENT TO DEFINE TSQLOG

DEFINE CLUSTER(NAME(CICSTS.GLOBAL.BASE.TSQLOG)-
  INDEXED -
  TRK(1 1)-
  SHR(2 3)-
  VOLUME(??????) REUSE) -
  DATA(NAME(CICSTS.GLOBAL.BASE.TSQLOG.DATA)-
    CISZ(4Ø96)-
    KEYS(Ø0 Ø)) -
  INDEX(NAME(CICSTS.GLOBAL.BASE.TSQLOG.INDEX))

DELETING TS QUEUE AT TERMINAL DELETION
With the above logging of TS queue creation in place, we can now associate queues with the user and terminal responsible for their creation. Since we have been careful to log only TS queues whose scope is limited to the individual terminal’s CICS session, we should be able to delete any queues associated with the terminal when the user signs off. We will also need to delete all records in the TSQLOG UMT for this terminal at the same time. The logical place to do all of this is in the terminal autoinstall exit.

SOURCE FOR DFHZATDX DELETE PROCESSING
Note: only relevant portions of the autoinstall URM are shown here.

DFHEISTG DSECT
RECLEN   DS   H
TSQLOG_START_KEY   DS   ØCL2Ø
TSQLOG_START_TRMID   DS   CL4
TSQLOG_START_TSQID   DS   CL16
COPY   TSQLOG   layout for TSQLOG UMT record
COPY   DFHTCUDS   commarea dsect
*

DFHZATDX   CSECT
DFHZATDX   RMODE ANY
DFHREGS
OC   EIBCALEN,EIBCALEN
BZ   RETURN
L   R2,DFHEICAP
USING INSTALL_EXIT_COMMAREA,R2
CLI   INSTALL_EXIT_FUNCTION,DELETE_CODE
BE   DELETE_TERMINAL
....   .............
DELETE_TERMINAL   DS   ØH
USING DELETE_EXIT_COMMAREA,R2
....   .............
EXEC   CICS HANDLE CONDITION ERROR(RETURN)
EXEC   CICS HANDLE CONDITION ENDFILE(RETURN)
EXEC   CICS IGNORE CONDITION QIDERR
MVC   TSQLOG_START_TRMID,DELETE_TERM_ID
XC   TSQLOG_START_TSQID,TSQLOG_START_TSQID
LA   R8,TSQLOG_REC_LEN
STH   R8,RECLEN
EXEC   CICS STARTBR   +
FILE('TSQLOG')   +
RIDFLD(TSQLOG_START_KEY)   +
GTEQ
MVC   TSQLOG_KEY(L'TSQLOG_TRMID+L'TSQLOG_TSQID),TSQLOG_START_KEY
DELTSQ   DS   ØH
EXEC   CICS READNEXT   +
FILE('TSQLOG')   +
RIDFLD(TSQLOG_KEY)   +
INTO(TSQLOG_RECORD)   +
LENGTH(RECLEN)
CLC   TSQLOG_TRMID,DELETE_TERM_ID
BNE   RETURN
EXEC   CICS DELETEQ TS   +
QNAME(TSQLOG_TSQID)
EXEC   CICS DELETE   +
FILE('TSQLOG')   +
RIDFLD(TSQLOG_KEY)
B   DELTSQ
....   .............
RETURN   DS   ØH
EXEC   CICS RETURN
MRO CONSIDERATIONS

For the sake of simplicity, this discussion has assumed a single-region CICS environment. Obviously, things get a bit more complicated when TS queues are created in regions that are interconnected to the terminal-owning region. Fortunately, the terminal autoinstall program is invoked for installation and deletion of shipped terminals as well. So, the processing prescribed here to occur at terminal deletion could be applied to shipped terminal deletion in MRO environments. On the other hand, since storage constraint is typically less of a problem in MRO configurations, one may want to carefully weigh the housekeeping over-head before proceeding.

Russell Hunt
Systems Programmer (USA)

Sending messages at CICS log-on

There are occasions when we need to broadcast a messages to CICS users. One way to do this is with the standard CMSG transaction. Another way is to send the message at CICS logon and/or logoff. To do this, one possible way is to replace the CICS good-morning transaction (CSGM) by our own. CSGM is declared in the SIT under GMTRAN, and is executed when a terminal is auto-installed in CICS. It is responsible for displaying the ‘Welcome to CICS/ESA’ screen or equivalent.

If you replace CSGM with your own transaction and associated program, you can send the user whatever initial screen or message you want. The only condition is that your transaction and program must be public to RACF, since the user has not yet signed-on at this stage.

The complete solution is implemented as follows:

• Define to CICS a VSAM KSDS with a key of 3 bytes and a record length of 1000. This file will hold the message to display. The default key, in my example, is ‘000’.

• Define also the following programs and transactions:
- Transaction MSGM and program MSGCP02. This transaction will replace CSGM in the SIT GMTRAN parameter. The program reads the VSAM file with key ‘000’.

- If that record is not found, it means that there are no messages to display, and it launches the regular CICS transaction CSGM and exits. From the user’s point of view, there is no difference to everyday log-ons.

- If the record is found, the program verifies whether today is within the range of valid days. If it is not, then the message has expired, and the program deletes the record and proceeds as above for CSGM. For this reason, the VSAM file needs full public access under CICS and RACF.

- If the day is still valid, then the message is sent to the screen and the transaction exits to CICS. After reading the message, the user can clear the screen and call the SIGN-ON transaction. This program has no BMS map associated. The message is sent as a clean 3270 datastream. For simple tasks like this, where no multiple sendings and receivings are involved, I prefer this approach.

- Transaction MSGP, program MSGCP01, and mapset MSGCS01. They are responsible for the creation and maintenance of the messages, and their use should be appropriately restricted.

When you call MSGP, the associated program tries to read the file. If the record is there, it loads it to the mapset and displays it. If it isn’t, then you have 12 empty lines, underscored for better visibility, where you can type your message. Don’t worry about leaving the underscores in the lines, they will be removed from the message (and for that reason, an underscore cannot be part of the message!). The screen is shown below:

```
+----------------------------------------------------------------------+
|                    Message for CICS Sign-on                            |
|   Message text:                                                     |
|                                                                     |
| ____________________________________________________________________|
| ____________________________________________________________________|
| _____________________Good morning, everyone_________________________|
| ___________________________________________________________________|
| _______________This is just a test, please ignore!_________________|
```
On the lower right corner, put the number of days that the message should be active. For example, if you only want the message to be displayed today, put one day. If you want it today and tomorrow, put two days. You don’t have to take any action to delete an expired message. Program MSGCP02 takes care of that, as explained above. If you wish to remove a message that you have previously set before it expires, just clear all the lines in the screen. This way, the record is
deleted from the file and no new record is written.

MSGCP01 SOURCE CODE

IDENTIFICATION DIVISION.
PROGRAM-ID. MSGCP01.
*=======================================================================*
* This program is associated with transaction MSGP.                  *
* This program prepares 'good-morning' messages for program MSGCP02.  *
* MSGCP02. Messages are written to a VSAM KSDS file that must be      *
* accessible by CICS.                                                *
*=======================================================================*
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*
77 W-NDAYS PIC 9(3).
77 TRANSACTION PIC X(4) VALUE 'MSGP'.
77 FICNAME PIC X(8) VALUE 'FILEMSG'.
77 FICLEN PIC S9(4) COMP VALUE +918.
77 COMMAREALEN PIC S9(4) COMP VALUE +1.
77 MSGFL PIC S9(4) COMP VALUE +30.
77 MSGF PIC X(30) VALUE SPACES.
77 MSGF1 PIC X(30) VALUE 'THERE ARE NO ACTIVE MESSAGES'.
77 MSGF2 PIC X(30) VALUE '*** MESSAGE PREPARED ***'.
77 MSGF3 PIC X(30) VALUE 'ERROR - FILE IS CLOSED'.
*
Ø1 COMMAREA PIC X VALUE LOW-VALUE.
*
Ø1 EIB-DAYS.
  Ø2 EIBDAT PIC 9(7) COMP-3.
  Ø2 EIBDATA PIC 9(7).
  Ø2 EIBDATA-R REDEFINES EIBDATA.
  Ø3 FILLER PIC X(4).
  Ø3 EIBDAYS PIC 9(3).
*
Ø1 FIC-RECORD PIC X(918) VALUE LOW-VALUES.
Ø1 FIC-RECORD-R REDEFINES FIC-RECORD.
  Ø2 FIC-KEY PIC 999.
  Ø2 FIC-NDAYS PIC 999.
  Ø2 FIC-LINES.
    Ø3 FIC-LINE01 PIC X(76).
    Ø3 FIC-LINE02 PIC X(76).
    Ø3 FIC-LINE03 PIC X(76).
    Ø3 FIC-LINE04 PIC X(76).
    Ø3 FIC-LINE05 PIC X(76).
    Ø3 FIC-LINE06 PIC X(76).
    Ø3 FIC-LINE07 PIC X(76).
Ø3 FIC-LINE08 PIC X(76).
Ø3 FIC-LINE09 PIC X(76).
Ø3 FIC-LINE10 PIC X(76).
Ø3 FIC-LINE11 PIC X(76).
Ø3 FIC-LINE12 PIC X(76).

*  
Ø1 MSGCS01-COPY.
Ø2 MSGCS01I PIC X(964) VALUE LOW-VALUES.
Ø2 MSGCS01I-R REDEFINES MSGCS01I.
   Ø5 FILLER PIC X(12).
   Ø5 LINE01L COMP PIC S9(4).
   Ø5 LINE01F PIC X(Ø1).
   Ø5 LINE01I PIC X(76).
   Ø5 LINE02L COMP PIC S9(4).
   Ø5 LINE02F PIC X(Ø1).
   Ø5 LINE02I PIC X(76).
   Ø5 LINE03L COMP PIC S9(4).
   Ø5 LINE03F PIC X(Ø1).
   Ø5 LINE03I PIC X(76).
   Ø5 LINE04L COMP PIC S9(4).
   Ø5 LINE04F PIC X(Ø1).
   Ø5 LINE04I PIC X(76).
   Ø5 LINE05L COMP PIC S9(4).
   Ø5 LINE05F PIC X(Ø1).
   Ø5 LINE05I PIC X(76).
   Ø5 LINE06L COMP PIC S9(4).
   Ø5 LINE06F PIC X(Ø1).
   Ø5 LINE06I PIC X(76).
   Ø5 LINE07L COMP PIC S9(4).
   Ø5 LINE07F PIC X(Ø1).
   Ø5 LINE07I PIC X(76).
   Ø5 LINE08L COMP PIC S9(4).
   Ø5 LINE08F PIC X(Ø1).
   Ø5 LINE08I PIC X(76).
   Ø5 LINE09L COMP PIC S9(4).
   Ø5 LINE09F PIC X(Ø1).
   Ø5 LINE09I PIC X(76).
   Ø5 LINE10L COMP PIC S9(4).
   Ø5 LINE10F PIC X(Ø1).
   Ø5 LINE10I PIC X(76).
   Ø5 LINE11L COMP PIC S9(4).
   Ø5 LINE11F PIC X(Ø1).
   Ø5 LINE11I PIC X(76).
   Ø5 LINE12L COMP PIC S9(4).
   Ø5 LINE12F PIC X(Ø1).
   Ø5 LINE12I PIC X(76).
   Ø5 NDAYSL COMP PIC S9(4).
   Ø5 NDAYSF PIC X(Ø1).
   Ø5 NDAYSI PIC 9(Ø1).
Ø2 MSGCS010 REDEFINES MSGCS01I.
   Ø5 FILLER PIC X(964).
* LINKAGE SECTION.
*----------------------------*
Ø1  DFHCOMMAREA.
   Ø2   FILLER PIC X.
*----------------------------*
PROCEDURE DIVISION.
*----------------------------*
*
MOVE EIBDATE TO EIBDAT
MOVE EIBDAT TO EIBDATA
EXEC CICS HANDLE CONDITION
   MAPFAIL (SEND-MESSAGE-RETURN)
   NOTOPEN (FILE-CLOSED)
END-EXEC
IF EIBCALEN = Ø
   GO TO FIRST-TIME
ELSE
   GO TO SECOND-TIME
END-IF.
*
FIRST-TIME.
*-------------*
EXEC CICS HANDLE CONDITION
   NOTFND (SEND-INITIAL)
END-EXEC
MOVE 'ØØØ' TO FIC-KEY
EXEC CICS READ DATASET (FICNAME)
   INTO (FIC-RECORD)
   RIDFLD (FIC-KEY)
   LENGTH (FICLEN)
END-EXEC
MOVE FIC-LINEØ1 TO LINEØ1I
MOVE FIC-LINEØ2 TO LINEØ2I
MOVE FIC-LINEØ3 TO LINEØ3I
MOVE FIC-LINEØ4 TO LINEØ4I
MOVE FIC-LINEØ5 TO LINEØ5I
MOVE FIC-LINEØ6 TO LINEØ6I
MOVE FIC-LINEØ7 TO LINEØ7I
MOVE FIC-LINEØ8 TO LINEØ8I
MOVE FIC-LINEØ9 TO LINEØ9I
MOVE FIC-LINE1Ø TO LINE1ØI
MOVE FIC-LINE11 TO LINE11I
MOVE FIC-LINE12 TO LINE12I
SUBTRACT EIBDAYS FROM FIC-NDAYS GIVING NDAYSI.
*
SEND-INITIAL.
*-------------*
IF NDAYSI NOT NUMERIC
MOVE 1 TO NDAYSI
END-IF
EXEC CICS SEND MAP ('MSGCS01')
    ERASE
END-EXEC
EXEC CICS RETURN TRANSID (TRANSACTION)
    COMMAREA (COMMAREA)
    LENGTH   (COMMAREALEN)
END-EXEC.
*
*======================================================================*
SECOND-TIME.
*------------*
EXEC CICS HANDLE AID PF3 (SEND-MESSAGE-RETURN)
    PF15 (SEND-MESSAGE-RETURN)
END-EXEC
EXEC CICS RECEIVE MAP('MSGCS01')
END-EXEC
IF NDAYSI NOT NUMERIC
    MOVE 1 TO NDAYSI
END-IF
INSPECT LINE01I REPLACING ALL '_' BY SPACES
INSPECT LINE02I REPLACING ALL '_' BY SPACES
INSPECT LINE03I REPLACING ALL '_' BY SPACES
INSPECT LINE04I REPLACING ALL '_' BY SPACES
INSPECT LINE05I REPLACING ALL '_' BY SPACES
INSPECT LINE06I REPLACING ALL '_' BY SPACES
INSPECT LINE07I REPLACING ALL '_' BY SPACES
INSPECT LINE08I REPLACING ALL '_' BY SPACES
INSPECT LINE09I REPLACING ALL '_' BY SPACES
INSPECT LINE10I REPLACING ALL '_' BY SPACES
INSPECT LINE11I REPLACING ALL '_' BY SPACES
INSPECT LINE12I REPLACING ALL '_' BY SPACES
INSPECT LINE01I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE02I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE03I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE04I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE05I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE06I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE07I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE08I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE09I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE10I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE11I REPLACING ALL LOW-VALUES BY SPACES
INSPECT LINE12I REPLACING ALL LOW-VALUES BY SPACES
MOVE 'ØØØ' TO FIC-KEY
EXEC CICS IGNORE CONDITION
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

* FILE-CLOSED.
*---------------------------------------------------*
EXEC CICS SEND (MSGF)
LENGTH (MSGFL)
ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

* FILE-CLOSED.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

SEND-MESSAGE-RETURN.
*---------------------------------------------------*
EXEC CICS SEND FROM (MSGF) LENGTH (MSGFL) ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.
MOVE MSGF3 TO MSGF
GO TO SEND-MESSAGE-RETURN.

MSGCP02 SOURCE CODE

PROGRAM-ID. MSGCP02.

*MSGCP02 - Program to show CICS 'good-morning' messages that were prepared by transaction MSGP.
*This program is associated with transaction MSGM.
*MSGM must be declared as GMTRAN at CICS SIT, instead of CSGM
*This transaction must be declared public in RACF, since it is invoked before any sign-on.

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.

77 FICNAME PIC X(8) VALUE 'FILEMSG'.
77 FICLEN PIC S9(4) COMP VALUE +918.
77 MESSAGELEN PIC S9(4) COMP VALUE +1176.

Ø1 EIBDATTT.
Ø2 EIBDAT PIC 9(7) COMP-3.
Ø2 EIBDATA PIC 9(7).
Ø2 FILLER REDEFINES EIBDATA.
Ø3 FILLER PIC X(4).
Ø3 EIBDAYS PIC 9(3).

Ø1 FIC-RECORD.

Ø2 FIC-KEY PIC 999.
Ø2 FIC-NDAYS PIC 999.
Ø2 FIC-LINES.
Ø3 FIC-LINE01 PIC X(76).
Ø3 FIC-LINE02 PIC X(76).
Ø3 FIC-LINE03 PIC X(76).
Ø3 FIC-LINE04 PIC X(76).
Ø3 FIC-LINE05 PIC X(76).
Ø3 FIC-LINE06 PIC X(76).
Ø3 FIC-LINE07 PIC X(76).
Ø3 FIC-LINE08 PIC X(76).
Ø3 FIC-LINE09 PIC X(76).
Ø3 FIC-LINE10 PIC X(76).
Ø3 FIC-LINE11 PIC X(76).
Ø3 FIC-LINE12 PIC X(76).

Ø1 MESSAGE-SCREEN.

Ø3 FILLER PIC X VALUE X'11'.
'<<<<>>> Please read the following Message <<<<<<'.

© 2001. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.
EXEC CICS ASSIGN
    APPLID (APPLID)
END-EXEC
EXEC CICS HANDLE ABEND
    LABEL (RETURN-CSGM)
END-EXEC
EXEC CICS HANDLE CONDITION
    NOTFND (RETURN-CSGM)
    NOTOPEN (RETURN-CSGM)
    LENGERR (RETURN-CSGM)
END-EXEC.
MOVE '000' TO FIC-KEY
EXEC CICS READ DATASET (FICNAME)
    INTO (FIC-RECORD)
    RIDFLD (FIC-KEY)
    LENGTH (FICLEN)
END-EXEC.
*  
MOVE EIBDATE TO EIBDAT
MOVE EIBDAT TO EIBDATA
IF FIC-NDAYS NOT GREATER EIBDAYS
    MOVE '000' TO FIC-KEY
    EXEC CICS DELETE DATASET (FICNAME)
        RIDFLD (FIC-KEY)
END-EXEC
    GO TO RETURN-CSGM
END-IF.
*  
SEND-MESSAGE.
*-----------------------------*
MOVE FIC-LINE01 TO LINE01
MOVE FIC-LINE02 TO LINE02
MOVE FIC-LINE03 TO LINE03
MOVE FIC-LINE04 TO LINE04
MOVE FIC-LINE05 TO LINE05
MOVE FIC-LINE06 TO LINE06
MOVE FIC-LINE07 TO LINE07
MOVE FIC-LINE08 TO LINE08
MOVE FIC-LINE09 TO LINE09
MOVE FIC-LINE10 TO LINE10
MOVE FIC-LINE11 TO LINE11
MOVE FIC-LINE12 TO LINE12

EXEC CICS SEND CONTROL
   ALARM
   FREEKB
END-EXEC
EXEC CICS SEND FROM (MESSAGE-SCREEN)
   LENGTH (MESSAGELEN)
   ERASE
END-EXEC.
EXEC CICS RETURN
END-EXEC.

* RETURN-CSGM.
*-----------------------------*
EXEC CICS START TRANSID ('CSGM')
   TERMID (EIBTRMID)
END-EXEC
EXEC CICS RETURN
END-EXEC.
GOBACK.

**MSGCS01 SOURCE CODE**

MAPSET DFHMSD TYPE=&SYSPARM,MODE=INOUT,CTRL=(FREEKB),
   LANG=COBOL,TIOAPFX=YES,EXTATT=MAPONLY
* MSGCS01 DFHMDI SIZE=(24,80)
*                  DFHMDF POS=(04,25),LENGTH=24,ATTRB=(ASKIP,PROT),
*                  COLOR=RED,
*                  INITIAL='Message for CICS Sign-on'
*                  DFHMDF POS=(05,01),LENGTH=13,ATTRB=(ASKIP,PROT),
*                  COLOR=DEFAULT,
*                  INITIAL='Message text:'
*
LINE01  DFHMDF  POS=(07,01), LENGTH=76, ATTRB=(UNPROT, FSET, IC), *
        COLOR=DEFAULT, *
        INITIAL='_______________________________________________*
                       _______________________________

        *  
        DFHMDF  POS=(07,78), LENGTH=01, ATTRB=(ASKIP, PROT)
LINE02  DFHMDF  POS=(08,01), LENGTH=76, ATTRB=(UNPROT, FSET), *
        COLOR=DEFAULT, *
        INITIAL='_______________________________________________*
                       _______________________________

        *  
        DFHMDF  POS=(08,78), LENGTH=01, ATTRB=(ASKIP, PROT)
LINE03  DFHMDF  POS=(09,01), LENGTH=76, ATTRB=(UNPROT, FSET), *
        COLOR=DEFAULT, *
        INITIAL='_______________________________________________*
                       _______________________________

        *  
        DFHMDF  POS=(09,78), LENGTH=01, ATTRB=(ASKIP, PROT)
LINE04  DFHMDF  POS=(10,01), LENGTH=76, ATTRB=(UNPROT, FSET), *
        COLOR=DEFAULT, *
        INITIAL='_______________________________________________*
                       _______________________________

        *  
        DFHMDF  POS=(10,78), LENGTH=01, ATTRB=(ASKIP, PROT)
LINE05  DFHMDF  POS=(11,01), LENGTH=76, ATTRB=(UNPROT, FSET), *
        COLOR=DEFAULT, *
        INITIAL='_______________________________________________*
                       _______________________________

        *  
        DFHMDF  POS=(11,78), LENGTH=01, ATTRB=(ASKIP, PROT)
LINE06  DFHMDF  POS=(12,01), LENGTH=76, ATTRB=(UNPROT, FSET), *
        COLOR=DEFAULT, *
        INITIAL='_______________________________________________*
                       _______________________________

        *  
        DFHMDF  POS=(12,78), LENGTH=01, ATTRB=(ASKIP, PROT)
LINE07  DFHMDF  POS=(13,01), LENGTH=76, ATTRB=(UNPROT, FSET), *
        COLOR=DEFAULT, *
        INITIAL='_______________________________________________*
                       _______________________________

        *  
        DFHMDF  POS=(13,78), LENGTH=01, ATTRB=(ASKIP, PROT)
LINE08  DFHMDF  POS=(14,01), LENGTH=76, ATTRB=(UNPROT, FSET), *
        COLOR=DEFAULT, *
        INITIAL='_______________________________________________*
                       _______________________________

        *  
        DFHMDF  POS=(14,78), LENGTH=01, ATTRB=(ASKIP, PROT)
LINE09  DFHMDF  POS=(15,01), LENGTH=76, ATTRB=(UNPROT, FSET), *
        COLOR=DEFAULT, *
INITIAL='_______________________________________________*
_______________________________________________'

* DFHMDF POS=(15,78),LENGTH=Ø1,ATTRB=(ASKIP, PROT)
  LINE10 DFHMDF POS=(16,01),LENGTH=76,ATTRB=(UNPROT, FSET), *
     COLOR=DEFAULT, *
     INITIAL='_______________________________________________*

* DFHMDF POS=(16,78),LENGTH=Ø1,ATTRB=(ASKIP, PROT)
  LINE11 DFHMDF POS=(17,01),LENGTH=76,ATTRB=(UNPROT, FSET), *
     COLOR=DEFAULT, *
     INITIAL='_______________________________________________*

* DFHMDF POS=(17,78),LENGTH=Ø1,ATTRB=(ASKIP, PROT)
  LINE12 DFHMDF POS=(18,01),LENGTH=76,ATTRB=(UNPROT, FSET), *
     COLOR=DEFAULT, *
     INITIAL='_______________________________________________*

* DFHMDF POS=(18,78),LENGTH=Ø1,ATTRB=(ASKIP, PROT)
  DFHMDF POS=(2Ø,1Ø),LENGTH=52,ATTRB=(ASKIP, PROT), *
  COLOR=YELLOW, *
  INITIAL='Number of days (including today) to display mes*
  sage:'

* NDAYS  DFHMDF POS=(2Ø,63),LENGTH=Ø1,ATTRB=(NUM,FSET), *
     COLOR=RED *

* DFHMDF POS=(2Ø,65),LENGTH=Ø1,ATTRB=(ASKIP, PROT)
  DFHMDF POS=(22,1Ø),LENGTH=38,ATTRB=(ASKIP, PROT), *
  COLOR=TURQUOISE, *
  INITIAL='To remove message, clear the text area'

* DFHMDF POS=(24,05),LENGTH=15,ATTRB=(ASKIP, PROT), *
  COLOR=YELLOW, *
  INITIAL='Enter - Execute'

* DFHMDF POS=(24,63),LENGTH=12,ATTRB=(ASKIP, PROT), *
  COLOR=YELLOW, *
  INITIAL='PF3 - Cancel'

* DFHMSD TYPE=FINAL END

Luis Paulo Figueiredo Sousa Ribeiro
Systems Engineer
Edinfor (Portugal) © Xephon 2001
Displaying task storage use in a CICS region

After implementing my program MAPTCA (see article *Displaying task activity in a CICS region under stress*, *CICS Update* Issue 182, January 2001), a customer requested that I modify it to also display the storage areas allocated to each of the tasks listed.

Since this required chasing through a completely different set of CICS control blocks, I decided to implement it as a separate program, and then to merge the output from the two programs in a REXX EXEC which produces a final report.

Thus I wrote program MAPDSA and REXX SOSCHECK. The output from MAPTCA and MAPDSA are used as input to SOSCHECK. Since MAPDSA was not intended to be run stand-alone, the output is rather more cryptic than that from MAPTCA, but it can still be useful in its own right. Here is a sample of MAPDSA’s output:

```
DATE:  20010129    TIME:   142114
CICSJOB: CICSV41A
TYP <-TASK->  <-ADDR-> <-LENG->
  SCA M0000004  06FA1CC8
  SCA B0000004  06FA1C14
  SCA C0000004  06FA1B60
  SCE     0708B000  00000570
  SCA U0000004  06FA1AAC
  SCA M0000006  06FA59F8
  SCE     0005E000  00003E0
  SCA M0000021  06FACE30
  SCE     00108000  0002F00
  SCE     00101000  00003E0
  SCA B0000021  06FACEE4
  SCA C0000021  06FADF02
  SCE     07318000  0000080
  SCA U0000021  06FADF04
```

The TYP column represents CICS storage control anchor and element control blocks, the elements being actual allocated areas of storage. The task number is prefixed by M, B, C, or U – these represent storage elements that are allocated within the CICS CDSA, UDSA, ECDSA, and EUDSA respectively. The address and length of the elements are also displayed.
SOSCHECK reads MAPDSA’s and MAPTCA’s output and combines them to produce a report like the following sample:

<table>
<thead>
<tr>
<th>Tran</th>
<th>SNCF</th>
<th>ØØØØ379</th>
<th>DBUGUSER</th>
<th>EDF</th>
<th>Ø7Ø5F68Ø</th>
<th>SUSPENDED</th>
</tr>
</thead>
<tbody>
<tr>
<td>OURNET.LUTPØØØ7</td>
<td>_UDSA element ØØ14ØØ7Ø ØØØØØØ3Ø</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>_UDSA element ØØ14ØØØØ ØØØØØØ7Ø</td>
<td>112</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total (E)DSA for this transaction:</td>
<td>160</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tran</th>
<th>SNCF</th>
<th>ØØØØ379</th>
<th>DBUGUSER</th>
<th>EDF</th>
<th>Ø7Ø5F68Ø</th>
<th>SUSPENDED</th>
</tr>
</thead>
<tbody>
<tr>
<td>OURNET.LUTPØØØ7</td>
<td>EUDSA element Ø75Ø3DAØ ØØØØ9Ø6Ø</td>
<td>36960</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EUDSA element Ø75ØØØØØ ØØØØ3DAØ</td>
<td>15776</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total (E)DSA for this transaction:</td>
<td>52736</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tran</th>
<th>CEDF</th>
<th>ØØØØ386</th>
<th>DFHZARQ1</th>
<th>ZCIOWAIT</th>
<th>Ø7Ø6ØØ8Ø</th>
<th>SUSPENDED</th>
</tr>
</thead>
<tbody>
<tr>
<td>OURNET.CICSV41A</td>
<td>_CDSA element ØØØ5ØØØØ ØØØØØ7CØ</td>
<td>1984</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total (E)DSA for this transaction:</td>
<td>1984</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tran</th>
<th>CEDF</th>
<th>ØØØØ386</th>
<th>DFHZARQ1</th>
<th>ZCIOWAIT</th>
<th>Ø7Ø6ØØ8Ø</th>
<th>SUSPENDED</th>
</tr>
</thead>
<tbody>
<tr>
<td>OURNET.CICSV41A</td>
<td>ECDSA element Ø739F3EØ ØØØØ2A9Ø</td>
<td>10896</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECDSA element Ø739FØØØ ØØØØ3EØ</td>
<td>992</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total (E)DSA for this transaction:</td>
<td>11888</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Total _CDSA for all transactions: | 21952 |
| Total _UDSA for all transactions: | 160 |
| Total ECDSA for all transactions: | 39984 |
| Total EUDSA for all transactions: | 52736 |
| Total (E)DSA for all transactions: | 114832 |

For the sake of brevity I have excluded all the CICS system transactions from this sample, which is why the totals exceed the sum of the elements shown.

MAPDSA has the same ‘look but don’t touch’ philosophy as MAPTCA, so it cannot harm a running CICS region in any way, but it might get confusing results if run against a very active CICS region that is not
currently stalled due to SOS, because CICS might update control block chains faster than MAPDSA can run them.

Since the intention is to run these programs when a CICS region is in SOS status, I used an automated operations package to detect the messages DFHSM0131 (SOS below 16MB) and DFHSM0133 (SOS above 16MB) and to execute a started task which ran MAPTCA, MAPDSA, and SOSCHECK as documented below.

MAPDSA must be linked AC=1 into an APF library. The source for macro R2D is supplied in the article *Displaying task activity in a CICS region under stress*, *CICS Update* Issue 182, January 2001.

**MAPDSA SOURCE**

```
**********************************************************************
**  MAP CICS DSAS - TASK SUBPOOLS                                    **
**********************************************************************
LCLC &MODULE
&MODULE SETC 'MAPDSA'
&MODULE CSECT
&MODULE AMODE 31
&MODULE RMODE 24
YREGS
SAVE (14,12)
USING MAPDSA,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
L R10,0(R11)
MVC JOBNAME(8),2(R10)
*--------------------------------------------------------------------
* FIND ASID OF CICS REGION
*--------------------------------------------------------------------
L R11,CVTPTTR
L R11,CVTASVT-CVTMAP(R11)
USING ASVT,R11
LA R10,ASVTENTRY
L R9,ASVTMAXU
ASVT_LOOP_ROUTINE DS ØH
TM Ø(R10),ASVTAVAL
BO TRY_NEXT_ASCB
```
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=65536
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)               DFHSMANCHOR
LA   R6,28(.R6)              DFHSMANCHOR
L    R6,Ø(.R6)               LAST SCA
LR    R3,R6                  DFHKEKCB
L    R6,KEKCB                DFHKEKCB
L    R6,Ø(.R6)               DFHSMANCHOR
LA  R6,24(,R6)
L  R6,Ø(,R6)   FIRST SCA

PROCESS_SCA DS ØH
LR  R5,R6      SAVE SCA POINTER
CLC Ø(1,R6),=CL1'B'    ELEMENTS ARE IN UDSA
BE  CONTINUE_SCA
CLC Ø(1,R6),=CL1'M'    ELEMENTS ARE IN CDSA
BE  CONTINUE_SCA
CLC Ø(1,R6),=CL1'U'    ELEMENTS ARE IN EUDSA
BE  CONTINUE_SCA
CLC Ø(1,R6),=CL1'C'    ELEMENTS ARE IN ECDSA
BE  CONTINUE_SCA
B  NEXT_SCA

CONTINUE_SCA DS ØH
MVC Ø(4,R7),=CL4'SCA'  
MVC 4(8,R7),Ø(R6)     SCA NAME
ST  R6,12(R7)         SCA ADDRESS
LA  R7,16(,R7)
LA  R6,8Ø(,R6)
LR  R8,R6
L  R6,Ø(,R6)          FIRST SCE
ST  R6,SCE_FIRST
CR  R6,R8            POINTS TO ITSELF IF NO SCE
BNE  PROCESS_SCE
B  NEXT_SCA

PROCESS_SCE DS ØH
MVC Ø(4,R7),=CL4'SCE'  
MVC 4(4,R7),8(R6)     SCE ADDRESS
MVC 8(4,R7),12(R6)    SCE LENGTH
LA  R7,16(,R7)
L  R6,Ø(,R6)         NEXT SCE
CR  R6,R8            POINTS TO ITSELF IF LAST SCE
BNE  PROCESS_SCE

NEXT_SCA    DS ØH
LR  R6,R5        RESTORE SCA POINTER
CR  R6,R3       LAST SCA?
BE  NO_MORE_SCA
LA  R6,8(,R6)   NEXT SCA
L  R6,Ø(,R6)   NEXT SCA
B  PROCESS_SCA

NO_MORE_SCA DS ØH
ST  R7,TABEND

*--------------------------------------------------------------------
* OUT OF ACCESS MODE
*--------------------------------------------------------------------

OUT_OF_ACCESS DS ØH
EPAR  R2
SSAR  R2
SAC  Ø
AXSET AX=AXØ
MODESET MODE=PROB,KEY=NZERO
*-------------------------------------------------------------------*
* 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
TIMEDATE DS ØCL16       TIME AND DATE RETURNED
   DC XL16'00'
ARASID DS H
KEKCB DS ØF
   DC XL4'00006418'
SCE_FIRST DS F
SCE_NEXT  DS F
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
   IHAASCBC
   CVT DSECT=YES
END

MAPDSA JCL FOR STAND-ALONE EXECUTION

//MAPDSA EXEC PGM=MAPDSA,PARM='CICSV41A'
//STEPLIB DD DSN=YOUR.APF.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=X

SOSCHECK REXX

F***************************************************************************/
F Function : CICS SOS report
***************************************************************************/
/* M = CDSA  B = UDSA  C = ECDSA  U = EUDSA */
/* ------------------------------------------------------------------- */
numeric digits 21
tran. = ''
totl = Ø; gtot = Ø; tot_c = Ø; tot_u = Ø; totec = Ø; toteu = Ø
done = 'n'
do while done = 'n'
"execio 1 diskr tcadata"
if rc = Ø then
do
parse pull tcarec
call proc_tran
end
else
done = 'y'
end
say ''
say jobn date
say ''
done = 'n'
do while done = 'n'
"execio 1 diskr dsadata"
if rc = Ø then
do
parse pull dsarec
call proc_dsa
end
else
done = 'y'
end
if total_to_write = 'y' then do
if totl ¬= Ø then do
say ''
say '   Total (E)DSA for this transaction:' format(totl,11,Ø)
say ''
end
say ''
say '   Total _CDSA for all transactions:' format(tot_c,11,Ø)
say '   Total _UDSA for all transactions:' format(tot_u,11,Ø)
say '   Total ECDSA for all transactions:' format(totec,11,Ø)
say '   Total EUDSA for all transactions:' format(toteu,11,Ø)
say ''
say '   Total (E)DSA for all transactions:' format(gtot,11,Ø)
say ''
exit
/* ------------------------------------------------------------------- */
/* Process a transaction record */
/* ------------------------------------------------------------------- */
proc_tran:
select
when substr(tcarec,3,5) = 'DATE:' then do
date = substr(tcarec,3,40)
end
when substr(tcarec,3,8) = 'CICSJOB:' then do
jobn = substr(tcarec,3,20)
end
when substr(tcarec,9,7) = '0000000' then
nop
when substr(tcarec,9,2) = '00' then do
task = substr(tcarec,11,5)
tran.task = substr(tcarec,1,80)
end
otherwise
nop
end
return
/*-------------------------------------------------------------------*/
/* Process a dsa record                                              */
/*-------------------------------------------------------------------*/
proc_dsa:
select
when substr(dsarec,3,3) = 'SCA' then do
first = 'y'
task = substr(dsarec,10,5)
type = substr(dsarec,7,1)
if totl ¬= 0 then do
say ''
say ' Total (E)DSA for this transaction:' format(totl,11,0)
say ''
totl = 0
total_to_write = 'n'
end
if tran.task = '' then
detail = 'Task' task 'not present when SOSTCA ran'
else
detail = 'Tran' tran.task
end
when substr(dsarec,3,3) = 'SCE' then do
addr = substr(dsarec,17,8)
leng = substr(dsarec,27,8)
lend = x2d(leng)
totl = totl + lend
gtot = gtot + lend
total_to_write = 'y'
if first = 'y' then do
first = 'n'
say detail
say ''
end
select
when type = 'M' then do
```
dsa_type = '_CDSA'  
tot_c = tot_c + lend  
end  
when type = 'B' then do  
dsa_type = '_UDSA'  
tot_u = tot_u + lend  
end  
when type = 'C' then do  
dsa_type = 'ECDSA'  
totec = totec + lend  
end  
when type = 'U' then do  
dsa_type = 'EUDSA'  
toteu = toteu + lend  
end  
otherwise nop  
end  
say ' ' dsa_type 'element' addr leng format(lend,11,0)  
extend  
otherwise  
nop  
extend  
return
```

### SOSCHECK 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 DSN=&&MAPTCA,DISP=(,PASS),UNIT=SYSDA,
//     SPACE=(TRK,(1,1)),DCB=(RECFM=VB,LRECL=137)
//*/
//MAPDSA EXEC PGM=MAPDSA,PARM='CICSV41A'
//STEPLIB DD DSN=YOUR.APF.LOADLIB,DISP=SHR
//SYSPRINT DD DSN=&&MAPDSA,DISP=(,PASS),UNIT=SYSDA,
//     SPACE=(TRK,(1,1)),DCB=(RECFM=VB,LRECL=137)
//*/
//SOSCHECK EXEC PGM=IRXJCL,PARM='SOSCHECK'
//SYSEXEC DD DSN=YOUR.SYSPROC,DISP=SHR
//TCADATA DD DSN=&&MAPTCA,DISP=SHR
//DSADATA DD DSN=&&MAPDSA,DISP=SHR
//SYSTSPRT DD SYSPUT=X
//SYSTSIN DD DUMMY
```

---

**Patrick Mullen**  
*CICS Consultant (Canada)*  
© Xephon 2001
Displaying a list of terminals

The following CICS program was created to sequentially display the terminals found in a CICS region and some associated characteristics.

Below is an example of the display. You can limit the scope of the output by introducing the initial characters for any of the listed fields. This is done in line three, under any field name. In this example, the terminal name is restricted to those beginning with RC and to those with a status of ACQ, or acquired. You can set restrictions for any field, including the task number.

Apart from that, you can use F2 to inhibit or restore the display of terminals that have no real user logged on (those with the CICS default user). In this example, you see several of those terminals, two of them with an active transaction. They are, in fact, printers. If you press F2, they are removed from the display.

The COBOL program, VITERMP, has the associated transaction name in a variable at the beginning of the working storage. My transaction name is VTER, but you can change it to whatever pleases you. In order for F2 to work correctly, you must also set the variable DEFAULT-USER with the correct name defined in your system.

There is an associated BMS map, VITERMS. Its modified copybook is already included in the working storage, so when you assemble it you need only to generate the module for the CICS loadlib.

Example display:

```
+----------------------------------------------------------------------+
|   PCICSA32                                       00/10/17   15:09:12 |
| Term  Netname  Sta  Userid   Username      Tran  Taskn  Next  Rnam  Rsys |
| RC     ACQ             |                      |        |          |        |      |      |      |      |      |
| RC4B   F12R4C2B Acq  CICSUSER USER CICS          |                      |        |          |        |      |      |      |      |      |
| RC4C   F12R4C2C Acq  CICSUSER USER CICS          |                      |        |          |        |      |      |      | PPC1  37449 |
| RC4E   F12R4C2E Acq  CICSUSER USER CICS          |                      |        |          |        |      |      |      | PPC2  37677 |
| RC4F   F12R4C2F Acq  CICSUSER USER CICS          |                      |        |          |        |      |      |      |      |      |
| RC4H   F12R4C2H Acq  CICSUSER USER CICS          |                      |        |          |        |      |      |      |      |      |
| RC4I   F12R4C2I Acq  CICSUSER USER CICS          |                      |        |          |        |      |      |      |      |      |
| RC4J   L82R4C25 Acq  EALAEI  ELISABETH P. A.      |                      |        |          |        |      |      |      | TR24  |      |
| RC46   L82R4C26 Acq  MALAMAN  M.THOMAS            |                      |        |          |        |      |      |      | TR24  |      |
| RC47   L82R4C27 Acq  CALIUMJ CALIUMJ              |                      |        |          |        |      |      |      | TEU8  |      |
```
VITERMP SOURCE CODE

IDENTIFICATION DIVISION.
PROGRAM-ID. VITERMP.
* ENVIRONMENT DIVISION.
DATA DIVISION.
*==============================================================*
WORKING-STORAGE SECTION.
*================================================================
77 X            PIC S9(4)  COMP VALUE +0.
77 Z            PIC S9(4)  COMP VALUE +0.
77 SLENG        PIC S9(4)  COMP VALUE +0.
77 W-RESP       PIC S9(8)  COMP VALUE +0.
77 W-RESP2      PIC S9(8)  COMP VALUE +0.
77 COUNT-I      PIC S9(4)  COMP VALUE +0.
77 ABSTIME      PIC S9(15) COMP-3.
77 TASK-AUX     PIC 9(8)        VALUE 0.
77 DEFAULT-USER PIC X(8)        VALUE 'CICSUSER'.
77 TRANS-NAME   PIC X(4)        VALUE 'VTER'.
77 THEEND       PIC X(5)        VALUE '*END*'.

Ø1 SRCH-FIELDS.
Ø2 SRCH-RESULT PIC X(20).
Ø2 SRCH1       PIC X(20).
Ø2 SRCH-1 REDEFINES SRCH1 PIC X OCCURS 20.
Ø2 SRCH2       PIC X(20).

Ø1 COMMAREA.

Ø2 DEFAULT-FLAG PIC 9.
Ø2 COUNT-LAST   PIC S9(4).
Ø2 W-LINHA.
Ø5 W-TERMN     PIC X(04).
Ø5 W-NETNA     PIC X(08).
Ø5 W-ACQST     COMP PIC S9(8).
Ø5 W-USERI     PIC X(08).
Ø5 W-USERN     PIC X(20).
Ø5 W-TRANS PIC X(Ø4).
Ø5 W-TASKN COMP PIC S9(8).
Ø5 W-NEXTT PIC X(Ø4).
Ø5 W-RMNAME PIC X(Ø4).
Ø5 W-RMSYS PIC X(Ø4).

*  
Ø2 LINHA-NOW.
Ø5 TERMN-NOW PIC X(Ø4).
Ø5 NETNA-NOW PIC X(Ø8).
Ø5 ACQST-NOW PIC X(Ø3).
Ø5 USERI-NOW PIC X(Ø8).
Ø5 USERN-NOW PIC X(2Ø).
Ø5 TRANS-NOW PIC X(Ø4).
Ø5 TASKN-NOW PIC 9(Ø5).
Ø5 NEXTT-NOW PIC X(Ø4).
Ø5 RMNAME-NOW PIC X(Ø4).
Ø5 RMSYS-NOW PIC X(Ø4).
Ø2 LINHA-PREV.
Ø5 FILLER PIC X(64).

*  
Ø2 VITERMSI.
Ø5 FILLER PIC X(12).
Ø5 CICSNL COMP PIC S9(4).
Ø5 CICSNF PIC X(Ø1).
Ø5 CICSNI PIC X(Ø8).
Ø5 DDATEL COMP PIC S9(4).
Ø5 DDATEF PIC X(Ø1).
Ø5 DDATEI PIC X(1Ø).
Ø5 DTIMEL COMP PIC S9(4).
Ø5 DTIMEF PIC X(Ø1).
Ø5 DTIMEI PIC X(Ø8).
Ø5 TERMNXL COMP PIC S9(4).
Ø5 TERMNXA PIC X(Ø1).
Ø5 TERMNXI PIC X(Ø4).
Ø5 NETNAXL COMP PIC S9(4).
Ø5 NETNAXA PIC X(Ø1).
Ø5 NETNAXI PIC X(Ø8).
Ø5 ACQSTXL COMP PIC S9(4).
Ø5 ACQSTXA PIC X(Ø1).
Ø5 ACQSTXI PIC X(Ø3).
Ø5 USERI XL COMP PIC S9(4).
Ø5 USERIXA PIC X(Ø1).
Ø5 USERIXI PIC X(Ø8).
Ø5 USERNXL COMP PIC S9(4).
Ø5 USERNXA PIC X(Ø1).
Ø5 USERNXI PIC X(2Ø).
Ø5 TRANSXL COMP PIC S9(4).
Ø5 TRANSXA PIC X(Ø1).
Ø5 TRANSXI PIC X(Ø4).
Ø5 TASKNXL COMP PIC S9(4).
05 TASKNXA PIC X(01).
05 TASKN XI PIC 9(05).
05 NEXTTXL COMP PIC S9(4).
05 NEXTTXA PIC X(01).
05 NEXTTXI PIC X(04).
05 RMNAMXL COMP PIC S9(4).
05 RMNAMXA PIC X(01).
05 RMNAMXI PIC X(04).
05 RMSYSXL COMP PIC S9(4).
05 RMSYSXA PIC X(01).
05 RMSYSXI PIC X(04).
05 SCREEN-LINES PIC X(1692).
05 LINEI REDEFINES SCREEN-LINES OCCURS 18.
07 TERMNL COMP PIC S9(4).
07 TERMNF PIC X(01).
07 TERMNI PIC X(04).
07 NETNAL COMP PIC S9(4).
07 NETNAF PIC X(01).
07 NETNAI PIC X(08).
07 ACQSTL COMP PIC S9(4).
07 ACQSTF PIC X(01).
07 ACQSTI PIC X(03).
07 USERIL COMP PIC S9(4).
07 USERIF PIC X(01).
07 USERII PIC X(08).
07 USERNL COMP PIC S9(4).
07 USERNF PIC X(01).
07 USERNI PIC X(20).
07 TRANSL COMP PIC S9(4).
07 TRANSF PIC X(01).
07 TRANSI PIC X(04).
07 TASKNL COMP PIC S9(4).
07 TASKNF PIC X(01).
07 TASKNI PIC X(05).
07 NEXTTL COMP PIC S9(4).
07 NEXTTF PIC X(01).
07 NEXTTI PIC X(04).
07 RMNAML COMP PIC S9(4).
07 RMNAMF PIC X(01).
07 RMNAMI PIC X(04).
07 RMSYSL COMP PIC S9(4).
07 RMSYSF PIC X(01).
07 RMSYSI PIC X(04).
02 VITERMSO REDEFINES VITERMSI.
07 FILLER PIC X(1833).

*=================================================================
* LINKAGE SECTION.
*=================================================================
01 DFHCOMMAREA.
PROCEDURE DIVISION.

FIRST-TIME-ONLY.

IF EIBCALEN = Ø
  MOVE LOW-VALUES TO COMMAREA
  MOVE Ø TO DEFAULT-FLAG
  MOVE 2030 TO EIBCALEN
  PERFORM INITIATE-SCREEN
  PERFORM CHECK-RESTRICTIONS-CHANGE
  PERFORM INQUIRE-CICS
  PERFORM SEND-SCREEN-ERASE
  GO TO RETURN-TRANSID
END-IF.

OTHER-TIMES.

MOVE DFHCOMMAREA TO COMMAREA
PERFORM RECEIVE-SCREEN THRU RECEIVE-SCREEN-EXIT
PERFORM CHECK-RESTRICTIONS-CHANGE
PERFORM INQUIRE-CICS
PERFORM SEND-SCREEN
GO TO RETURN-TRANSID.

CHECK-RESTRICTIONS-CHANGE.

MOVE TERMNXI TO TERMN-NOW
MOVE NETNAXI TO NETNA-NOW
MOVE ACQSTXI TO ACQST-NOW
MOVE USERIXI TO USERI-NOW
MOVE USERNXI TO USERN-NOW
MOVE TRANSXI TO TRANS-NOW
MOVE TASKNIXI TO TASKN-NOW
MOVE NEXTTXI TO NEXTT-NOW
MOVE RMNAMXI TO RMNAM-NOW
MOVE RMSYSXI TO RMSYS-NOW
IF LINHA-PREV NOT EQUAL LINHA-NOW
  MOVE LOW-VALUES TO W-TERMN
  MOVE Ø TO COUNT-LAST
END-IF
MOVE LINHA-NOW TO LINHA-PREV.

INQUIRE-CICS.

MOVE Ø TO X.
EXEC CICS INQUIRE TERMINAL START
   PERFORM INQUIRE-CICS-LOOP THRU
       INQUIRE-CICS-LOOP-EXIT UNTIL X > 18.
   MOVE COUNT-I TO COUNT-LAST
EXEC CICS INQUIRE TERMINAL END
   END-EXEC.

* INQUIRE-CICS-LOOP.
   *==================*
EXEC CICS INQUIRE
   TERMINAL  (W-TERMN)
   ACQSTATUS  (W-ACQST)
   NETNAME   (W-NETNA)
   USERID    (W-USERI)
   USERNAME  (W-USERN)
   TRANSACTION (W-TRANS)
   TASKID    (W-TASKN)
   NEXTTRANSID  (W-NEXTT)
   REMOTENAME  (W-RMNAM)
   REMOTESYSTEM (W-RMSYS)
   RESP      (W-RESP)
   RESP2     (W-RESP2)
   NEXT
   END-EXEC

IF W-RESP2 > Ø
   ADD 1 TO X
   PERFORM CLEAN-SCREEN-LINES
   MOVE 99 TO X
   MOVE Ø TO COUNT-I
   GO TO INQUIRE-CICS-LOOP-EXIT
END-IF

ADD 1 TO COUNT-I
IF COUNT-LAST GREATER COUNT-I
   GO TO INQUIRE-CICS-LOOP-EXIT
END-IF

IF DEFAULT-FLAG = 1 AND W-USERI = DEFAULT-USER
   GO TO INQUIRE-CICS-LOOP-EXIT
END-IF

IF NOT (TERMNXI = SPACES OR = LOW-VALUES)
   MOVE 4 TO SLENG
   MOVE TERMNXI TO SRCH1
   MOVE W-TERMN TO SRCH2
   PERFORM SRCH-STRING
   IF SRCH-RESULT = Ø
      GO TO INQUIRE-CICS-LOOP-EXIT
END-IF
END-IF

IF NOT (NETNAXI = SPACES OR = LOW-VALUES)
  MOVE 8 TO SLENG
  MOVE NETNAXI TO SRCH1
  MOVE W-NETNA TO SRCH2
  PERFORM SRCH-STRING
  IF SRCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT
  END-IF
END-IF

IF NOT (ACQSTXI = SPACES OR = LOW-VALUES)
  MOVE 3 TO SLENG
  MOVE ACQSTXI TO SRCH1
  PERFORM TRANSLATE-ACQSTATUS
  PERFORM SRCH-STRING
  IF SRCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT
  END-IF
END-IF

IF NOT (USERIXI = SPACES OR = LOW-VALUES)
  MOVE 8 TO SLENG
  MOVE USERIXI TO SRCH1
  MOVE W-USERI TO SRCH2
  PERFORM SRCH-STRING
  IF SRCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT
  END-IF
END-IF

IF NOT (USERNXI = SPACES OR = LOW-VALUES)
  MOVE 2Ø TO SLENG
  MOVE USERNXI TO SRCH1
  MOVE W-USERN TO SRCH2
  PERFORM SRCH-STRING
  IF SRCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT
  END-IF
END-IF

IF NOT (TRANSXI = SPACES OR = LOW-VALUES)
  MOVE 4 TO SLENG
  MOVE TRANSXI TO SRCH1
  MOVE W-TRANS TO SRCH2
  PERFORM SRCH-STRING
  IF SRCH-RESULT = Ø
    GO TO INQUIRE-CICS-LOOP-EXIT

IF NOT (TASKNXI = SPACES OR = LOW-VALUES)
    MOVE 5 TO SLENG
    MOVE TASKNXI TO SRCH1
    MOVE W-TASKN TO TASK-AUX
    MOVE TASK-AUX(4:5) TO SRCH2
    PERFORM SRCH-STRING
    IF SRCH-RESULT = Ø
        GO TO INQUIRE-CICS-LOOP-EXIT
    END-IF
END-IF

IF NOT (NEXTTXI = SPACES OR = LOW-VALUES)
    MOVE 4 TO SLENG
    MOVE NEXTTXI TO SRCH1
    MOVE W-NEXTT TO SRCH2
    PERFORM SRCH-STRING
    IF SRCH-RESULT = Ø
        GO TO INQUIRE-CICS-LOOP-EXIT
    END-IF
END-IF

IF NOT (RMNAMXI = SPACES OR = LOW-VALUES)
    MOVE 4 TO SLENG
    MOVE RMNAMXI TO SRCH1
    MOVE W-RMNAM TO SRCH2
    PERFORM SRCH-STRING
    IF SRCH-RESULT = Ø
        GO TO INQUIRE-CICS-LOOP-EXIT
    END-IF
END-IF

IF NOT (RMSYSXI = SPACES OR = LOW-VALUES)
    MOVE 4 TO SLENG
    MOVE RMSYSXI TO SRCH1
    MOVE W-RMSYS TO SRCH2
    PERFORM SRCH-STRING
    IF SRCH-RESULT = Ø
        GO TO INQUIRE-CICS-LOOP-EXIT
    END-IF
END-IF

ADD 1 TO X
MOVE W-TERMN TO TERMNI(X)
MOVE W-NETNA TO NETNAI(X)
MOVE W-USERI TO USERII(X)
MOVE W-USERN TO USERNI(X)
MOVE W-TRANS TO TRANSI(X)
MOVE W-TASKN TO TASK-AUX
IF TASK-AUX GREATER Ø
   MOVE TASK-AUX(4:5) TO TASKNI(X)
ELSE
   MOVE SPACES TO TASKNI(X)
END-IF
MOVE W-NEXTT TO NEXTTI(X)
MOVE W-RMSYS TO RMSYSI(X)
MOVE W-RMNAM TO RMNAMI(X)
PERFORM DISPLAY-ACQSTATUS.
*
INQUIRE-CICS-LOOP-EXIT.
*======================*
EXIT.
*
TRANSLATE-ACQSTATUS.
*====================*
IF W-ACQST = DFHVALUE(ACQUIRED)
   MOVE 'ACQ' TO SRCH2
ELSE
   IF W-ACQST = DFHVALUE(ACQUIRING)
      MOVE 'AQI' TO SRCH2
   ELSE
      IF W-ACQST = DFHVALUE(RELEASED)
         MOVE 'REL' TO SRCH2
   END-IF.
*
DISPLAY-ACQSTATUS.
*==================*
IF W-ACQST = DFHVALUE(ACQUIRED)
   MOVE 'Acq' TO ACQSTI(X)
ELSE
   IF W-ACQST = DFHVALUE(ACQUIRING)
      MOVE 'Aqi' TO ACQSTI(X)
   ELSE
      IF W-ACQST = DFHVALUE(RELEASED)
         MOVE 'Rel' TO ACQSTI(X)
   END-IF.
*
SRCH-STRING.
*============*
IF SRCH-1(SLENG) = SPACE OR = '*'
   SUBTRACT 1 FROM SLENG
   GO TO SRCH-STRING
END-IF
IF SRCH1(1:SLENG) = SRCH2(1:SLENG)
   MOVE 1 TO SRCH-RESULT
ELSE
   MOVE Ø TO SRCH-RESULT
END-IF.
* CLEAN-SCREEN-LINES.
*=================================* 
PERFORM CLEAN-SCREEN-LINES-LOOP
  VARYING Z FROM X BY 1 UNTIL Z > 18.
* 
CLEAN-SCREEN-LINES-LOOP.
*=================================* 
MOVE SPACES TO TERMINI(Z) NETNAI(Z) USERII(Z)
  ACQSTI(Z) USERNI(Z) TRANSI(Z) NEXTTI(Z)
  RMSYSI(Z) RMNAMI(Z) TASKNI(Z).
* 
SEND-SCREEN-ERASE.
*=================================* 
EXEC CICS SEND MAP('VITERMS') ERASE
END-EXEC.
* 
SEND-SCREEN.
*===============* 
EXEC CICS SEND MAP('VITERMS') DATAONLY
END-EXEC.
* 
RECEIVE-SCREEN.
*================* 
EXEC CICS HANDLE CONDITION MAPFAIL(RETURN-EXIT)
END-EXEC 
EXEC CICS HANDLE AID PF3  (RETURN-EXIT)
  PF15 (RETURN-EXIT)
  PF2  (TOGGLE-DEFAULT-FLAG)
  PF14 (TOGGLE-DEFAULT-FLAG)
END-EXEC
EXEC CICS RECEIVE MAP('VITERMS')
END-EXEC
GO TO RECEIVE-SCREEN-EXIT.
* 
TOGGLE-DEFAULT-FLAG.
*====================* 
MOVE Ø TO COUNT-LAST
  IF DEFAULT-FLAG = Ø 
  MOVE 1 TO DEFAULT-FLAG
ELSE 
  MOVE Ø TO DEFAULT-FLAG
END-IF
GO TO RECEIVE-SCREEN-EXIT.
* 
RECEIVE-SCREEN-EXIT.
*====================* 
EXIT.
* 
INITIATE-SCREEN.
*================*
EXEC CICS ASSIGN APPLID (CICSNI)
END-EXEC
EXEC CICS ASKTIME ABSTIME (ABSTIME)
END-EXEC
EXEC CICS FORMATTIME
  ABSTIME (ABSTIME)
  DATE (DDATEI)
  DATESEP ('/')
  TIME (DTIMEI)
  TIMESEP (':')
END-EXEC.
*
RETURN-TRANSID.
*
EXEC CICS RETURN
  TRANSID (TRANS-NAME)
  COMMAREA (COMMAREA)
  LENGTH (EIBCALEN)
END-EXEC.
*
RETURN-EXIT.
*
EXEC CICS SEND
  FROM (THEEND)
  LENGTH (5)
  ERASE
END-EXEC
EXEC CICS RETURN
END-EXEC
GOBACK.

VITERMS SOURCE CODE

MAPSET DFHMSD TYPE=&SYSPARM,MODE=INOUT,CTRL=(FREEKB),
  LANG=COBOL,TIOAPFX=YES,EXTATT=MAPONLY
*
VITERMS DFHMDI SIZE=(24,8Ø)
*
CICSN DFHMDF POS=(Ø1,Ø4),LENGTH=Ø8,ATTRB=(ASKIP,PROT),
  COLOR=YELLOW
DDATE DFHMDF POS=(Ø1,57),LENGTH=1Ø,ATTRB=(ASKIP,PROT),
  COLOR=YELLOW
DTIME DFHMDF POS=(Ø1,68),LENGTH=Ø8,ATTRB=(ASKIP,PROT,FSET),
  COLOR=YELLOW
  *
  DFHMDF POS=(Ø2,Ø2),LENGTH=Ø4,ATTRB=(ASKIP,PROT),
  COLOR=NEUTRAL,
  INITIAL='Term'
  DFHMDF POS=(Ø2,Ø7),LENGTH=Ø7,ATTRB=(ASKIP,PROT),
  COLOR=NEUTRAL,
INITIAL='Netname'
DFHMDF POS=(Ø2,16),LENGTH=Ø3,ATTRB=(ASKIP,PROT), *
COLOR=NEUTRAL,
INITIAL='Sta'
DFHMDF POS=(Ø2,21),LENGTH=Ø6,ATTRB=(ASKIP,PROT), *
COLOR=NEUTRAL,
INITIAL='Userid'
DFHMDF POS=(Ø2,30),LENGTH=Ø8,ATTRB=(ASKIP,PROT), *
COLOR=NEUTRAL,
INITIAL='Username'
DFHMDF POS=(Ø2,52),LENGTH=Ø4,ATTRB=(ASKIP,PROT), *
COLOR=NEUTRAL,
INITIAL='Tran'
DFHMDF POS=(Ø2,57),LENGTH=Ø5,ATTRB=(ASKIP,PROT), *
COLOR=NEUTRAL,
INITIAL='Taskn'
DFHMDF POS=(Ø2,63),LENGTH=Ø4,ATTRB=(ASKIP,PROT), *
COLOR=NEUTRAL,
INITIAL='Next'
DFHMDF POS=(Ø2,69),LENGTH=Ø4,ATTRB=(ASKIP,PROT), *
COLOR=NEUTRAL,
INITIAL='Rnam'
DFHMDF POS=(Ø2,74),LENGTH=Ø4,ATTRB=(ASKIP,PROT), *
COLOR=NEUTRAL,
INITIAL='Rsys'

TERMNX DFHMDF POS=(Ø3,02),LENGTH=Ø4,ATTRB=(BRT,UNPROT,FSET,IC), *
COLOR=PINK
NETNAX DFHMDF POS=(Ø3,07),LENGTH=Ø8,ATTRB=(BRT,UNPROT,FSET), *
COLOR=PINK
ACQSTX DFHMDF POS=(Ø3,16),LENGTH=Ø3,ATTRB=(BRT,UNPROT,FSET), *
COLOR=PINK
USERIX DFHMDF POS=(Ø3,21),LENGTH=Ø8,ATTRB=(BRT,UNPROT,FSET), *
COLOR=PINK
USERNX DFHMDF POS=(Ø3,30),LENGTH=2Ø,ATTRB=(BRT,UNPROT,FSET), *
COLOR=PINK
TRANSX DFHMDF POS=(Ø3,52),LENGTH=Ø4,ATTRB=(BRT,UNPROT,FSET), *
COLOR=PINK
TASKNX DFHMDF POS=(Ø3,57),LENGTH=Ø5,ATTRB=(BRT,UNPROT,FSET,NUM), *
COLOR=PINK
NEXTTX DFHMDF POS=(Ø3,63),LENGTH=Ø4,ATTRB=(BRT,UNPROT,FSET), *
COLOR=PINK
RMNAMX DFHMDF POS=(Ø3,69),LENGTH=Ø4,ATTRB=(BRT,UNPROT,FSET), *
COLOR=PINK
RMSYSX DFHMDF POS=(Ø3,74),LENGTH=Ø4,ATTRB=(BRT,UNPROT,FSET), *
COLOR=PINK
DFHMDF POS=(Ø3,79),LENGTH=Ø1
DFHMDF POS=(Ø4,02),LENGTH=76,COLOR=RED,
INITIAL='-----------------------------------------------* 
-----------------------------------------------*
Editor's note: this code continues for many more terminals.

Luis Paulo Figueiredo Sousa Ribeiro  
Systems Engineer  
Edinfor (Portugal)  

© Xephon 2001
IBM has announced Version 2 of its CICS Transaction Server for z/OS, adding support for Enterprise JavaBeans, improved network connectivity, extensions to facilities for applications based on procedural programming models, and significant extensions to CICSPlex.

As a result, CICS TS Version 2 is now an EJB server supporting EJB 1.1 programming and functions for applications written in other languages.

Support for enterprise beans in CICS TS Version 2 includes exploitation of a new optimized JVM, enhanced CORBA function, and enhanced CICSPlex functions including workload management.

Other facilities available to enterprise beans are JDBC/SQLJ access to DB2 data, JCICS access to VSAM data, extensions to JCICS, and the CICS Connector for CICS TS, enabling interoperability between enterprise beans and applications and data using other programming models.

There are also improvements to Web enablement, including support for the external call interface (ECI) over TCP/IP. Functions aimed at application development include an enhanced 3270 bridge, an integrated CICS translator for use with COBOL and PL/I, and enhancements to file control function shipping.

Connectivity improvements include support for the VTAM LU alias facility. Availability improvements include sign-on retention for persistent sessions, support for system-managed rebuild of coupling facility structures, automatic restart of CICS data-sharing servers, and CICSPlex workload management of 3270 bridge.

Usability enhancements come via changes to the CICSPlex Web user interface.

New functions will arrive in Releases 1 and 2. CICS TS Version 2.1 is targeted specifically at early adopters of EJB technology, while Version 2.2 is the next release recommended for all CICS users for deployment in large-scale production.

Meanwhile, the company says it will enhance CICS VSAM Recovery (CICSVR) as a part of continuing customer support, addressing a number of requested enhancements and introducing new functions, which include batch logging for VSAM data and change accumulation processing.

It’s likely to be packaged with CICS TS Version 2, but a separate licence for CICSVR will be required. It will also be sold as a separately orderable product.

For further information contact your local IBM representative.


* * *