177

August 2000

3 Selective autoinstall for programs and maps
9 Interfacing CICS to SMTP – part 2
17 TCP/IP and CICS sockets
32 Resource definition display and alter commands – part 1
48 CICS news

© Xephon plc 2000
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.

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 2000. 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.
Selective autoinstall for programs and maps

This is a customized version of IBM’s DFHPGADX program and it works under CICS/ESA 4.1 and CICS TS 1.3. You can find the original program in your hlq.slq.SDFHSAMP library.

The benefit of this kind of autoinstall is that you are able to choose which program or map should be autoinstalled in an AOR or in a SERVICE-CICS, like a TOR, QOR, or FOR, and you can be sure that only those programs or maps defined via RDO or in this program are active.

To discover whether or not you are running in an AOR, you can use the SYSID(field PSYSID) and then you can activate the indicator in field PSERVICE.

All user programs and maps that should be running in an AOR are defined in the PGMTAB or MAPTAB table. All non-user programs or maps, which should be running in a SERVICE-CICS, are defined in the PGMTABS or MAPTABS table. You can work with a match-code and you don’t need to define each program or map with the full name.

Also, it is possible to autoinstall from system-resources like LE/370 and CSP.

Don’t forget that you must define the autoinstall program in your SIT.

***********************************************************************
* MODULE NAME = SELECTAI                                           *
* DESCRIPTIVE NAME = CICS/ESA 4.1      Program autoinstall program exit *
*                        CICS-TS 1.3                                             *
* STATUS = 4.1.Ø AND 1.3.Ø                                                *
* FUNCTION = Provides user input for the program autoinstall function *
*            There are ASM, PL/I, COBOL, and C versions of this program. *
*            This program is a customized version in Assembler of the pro-
*            gram autoinstall exit. The program is invoked when a program *
*            is being autoinstalled on behalf of the user and the *
*            autoinstall exit name is set in the SIT to SELECTAI.      *
*            The exit may be used to specify requirements for the *
*            program definition.                                      *
*            A parameter list is provided as input to the program. The *
*            parameter list is passed to the program via the commarea. *
*            The parameter list is defined in DFHPGACD.              *
***********************************************************************
The parameter list is addressed by the program using the normal conventions for a commarea.
The parameter list specifies the name of the program to be autoinstalled and the module type. The user may use the parameter list to return information for the program to be autoinstalled. The user may also indicate using the return_code parameter that the program should not be defined.

NOTES:

THIS IS A CUSTOMIZED VERSION OF IBM'S PROGRAM "DFHPAGDX".
REFER TO PRODUCT DOCUMENTATION.
DEPENDENCIES = S/390
MODULE TYPE = Executable
PROCESSOR = Assembler
ATTRIBUTES = Read-only, serially reusable

ENTRY POINT = SELECTAI
PURPOSE = All functions
LINKAGE =

This entry point is called by the autoinstall function to link to the program autoinstall exit program. The parameters are passed to the exit program via the commarea. The control block for the parameter list is in DFHPGACD.

INPUT =
The input parameters provide the user with the name and module type for the program to be autoinstalled. The following input parameters are passed to the program via the commarea:
PGAC_PROGRAM - name of program to be autoinstalled
PGAC_MODULE_TYPE - program, mapset, or partitionset

OUTPUT =
The output parameters may be used to specify user requirements for the program definition. The following output parameters may be returned to the autoinstall function via the commarea:
PGAC_MODEL_NAME - autoinstall model program name
PGAC_LANGUAGE - Assembler, COBOL, C370, LE370, PL/I
PGAC_CEDF_STATUS - CEDF status, yes or no
PGAC_DATA_LOCATION - data location, below or any
PGAC_EXECUTION_KEY - execution key, CICS or user
PGAC_LOAD_ATTRIBUTE - reload, transient, resident, reusable
PGAC_USE_LPA_COPY - use LPA copy, yes or no
PGAC_EXECUTION_SET - use DPL subset or full API
PGAC_REMOTE_SYSID - remote system ID
PGAC_REMOTE_PROGID - remote program name
PGAC_REMOTE_TRANSID - remote transaction ID
EXIT-NORMAL = Exit is via an EXEC CICS RETURN command.
The following return codes may be returned via the commarea:
PGAC_RETURN_CODE = PGAC_RETURN_OK
PGAC_RETURN_CODE = PGAC_RETURN_DONT_DEFINE_PROGRAM
EXIT-ERROR =
  If the program abends, an error response is returned to the autoinstall function. A message is issued by the autoinstall function and the autoinstall function is disabled.

EXTERNAL REFERENCES =
  None.
ROUTINES =
  EXEC CICS RETURN - return to the calling program.
CONTROL BLOCKS =
  The PGAC control block, which includes the input and output parameters, is in DFHPGACD.
  See INPUT and OUTPUT description above for a description of the parameters.

DESCRIPTION =
  The default program autoinstall exit simply sets the return code to OK and returns.
  The user may customize this program to provide information for the autoinstalled definition based on the program name and the module type.

CHANGE ACTIVITY :
  $MOD(SELECTAI) COMP(PROGRAM) PROD(CICS/ESA AND CICS-TS 1.3): REMARKS
  $LØ= 646 41Ø 93Ø222 HDBVDMC: Program Autoinstall
  $P1= M83159 41Ø 93Ø713 HDBVDMC: M83159: DSECTGEN changes

DFHEISTG DSECT
  Insert your own storage definitions here
PPRIVATE DS ØCL2Ø
PSYSID DS CL4
PSERVICE DS CL1
PRESP DS F
  ORG PPRIVATE+2Ø
  Copy the commarea definitions
  COPY DFHPGACD Autoinstall commarea

SELECTAI CSECT
SELECTAI AMODE 31
SELECTAI RMODE ANY
DFHREGS,
  If there is no commarea, return
OC EIBCALEN,EIBCALEN
BZ RETURNØ
  Address the commarea
  L R2,DFHEICAP
USING PGAC,R2

* Add user specific code here
   CLI   PGAC_MODULE_TYPE,PGAC_TYPE_PARTITIONSET
   BE    RETURNDD               Accept only programs and maps
   MVI   PPRIVATE,'X'ØØ'        FORMAT WORKINGSET
   MVC   PPRIVATE+1(L'PPRIVATE-1),PPRIVATE

* Assign the sysid
   EXEC CICS ASSIGN SYSID(PSYSID) RESP(PRESP)
   CLC   PRESP,DFHRESP(NORMAL)  ANY ERRORS DETEDED ?
   BNE   RETURNDD              IF YES: DON'T AUTOINSTALL
   LA    R7,SERTAB            LOAD SERVICE-TAB
   SERV1000 DS 0H
   CLI   0(R7),C'*'           END OF TABLE ?
   BE    SERV9000            YES: IT'S NOT A SERVICE-CICS
   CLC   PSYSID+2(1),0(R7)    ENTRY IN TABLE ?
   BE    SERV1900           YES: IT'S A SERVICE-CICS
   LA    R7,L'SERTAB(R7)     NEXT ENTRY
   B     SERV1000

SERV1900 DS 0H
   MVI   PSERVICE,C'1'

SERV9000 DS 0H

* ---- Program ------------
   CLI   PSERVICE,C'1'       is it a service-cics ?
   BNE   PGM05000             no... load aor-table
   LA    R7,PGMTABS         first entry in table
   LA    R10,PGMCNTS        number of programs
   B     PGM1000

PGM05000 DS 0H
   LA    R7,PGMTAB          first entry in table
   LA    R10,PGMCNT         number of programs

PGM1000 DS 0H
   LA    R8,7               max.-length -1 (EX|)
   LR    R9,R7              addr. r9 eq addr. r7
   LA    R9,7(R9)           last possible character

PGM2000 DS 0H
   CLI   0(R9),C' '         true only gt blank
   BH    PGM3000             if gt..compare
   BCTR  R9,0                next column
   BCT   R8,PGM2000          go on

PGM3000 DS 0H
   EX    R8,COMPPGM        compare
   BE    PROCESS          yes... go on
   LA    R7,L'PGMTAB(,R7)  next entry
   BCT   R10,PGM1000      go on

* ---- Maps ---------------
   CLI   PSERVICE,C'1'       is it a service-cics ?
   BNE   MAP05000            no... load aor-table
   LA    R7,MAPTABS         first entry in table
   LA    R10,MAPCNTS        number of maps
   B     MAP1000
MAP0500 DS 0H
LA R7, MAPTAB first entry in table
LA R10, MAPCNT number of maps

MAP1000 DS 0H
LA R8, 7 max.-length -1 (EX])
LR R9, R7 addr. r9 eq addr. r7
LA R9, 7(R9) last possible character

MAP2000 DS 0H
CLI 0(R9), C' ' true only gt blank
BH MAP3000 if gt..compare
BCTR R9, 0 next column
BCT R8, MAP2000 go on

MAP3000 DS 0H
EX R8, COMPPGM compare
BE MAP yes.. go on
LA R7, L'MAPTAB(, R7) next entry
BCT R10, MAP1000 go on
B RETURNDD no map ... goback

MAP DS 0H
MVC PGAC_MODEL_NAME, =CL8'DFHPGAMP' SET DEFAULT-TYPE
B RETURNOK

PROCESS DS 0H
MVC PGAC_MODEL_NAME, =CL8'DFHPGAPG' SET DEFAULT-TYPE
MVI PGAC_CEDF_STATUS, PGAC_CEDF_YES "CEDF = YES"
MVI PGAC_DATA_LOCATION, PGAC_LOCATION_ANY TASKDATALOC=ANY
MVI PGAC_EXECUTION_KEY, PGAC_USER_KEY "EXECKEY=USER"
LA R7, RESTAB LOAD FIRST ENTRY

PROC1000 DS 0H
CLI 0(R7), C' ' END OF TABLE ?
BE RETURNOK YES: LOAD WITH "RESIDENT=NO"
CLC PGAC_PROGRAM, 0(R7) PROGRAM IN TABLE ?
BE PROC1900 YES: LOAD WITH "RESIDENT=YES"
LA R7, L'RESTAB(R7) NEXT ENTRY
B PROC1000

COMPPGM CLC PGAC_PROGRAM(0), 0(R7) PROGRAM TRUE?
PROC1900 DS 0H LOAD PROGRAM RESIDENT
MVI PGAC_LOAD_ATTRIBUTE, PGAC_RESIDENT
* Set the return code to OK
RETURNOK DS 0H
MVI PGAC_RETURN_CODE, PGAC_RETURN_OK
B RETURN0
* Branch to this label if you elect not to define the program
RETURNDD DS 0H
MVI PGAC_RETURN_CODE, PGAC_RETURN_DONT_DEFINE_PROGRAM
RETURN0 DS 0H
EXEC CICS RETURN
EJECT
*
PRIVATE DEFINITIONS
RESTAB DS 0CL8 TABLE FOR PROGRAMS, WHICH
DC CLB'CI1190 ' MUST BE LOADED RESIDENT |
SERTAB DS ØCL1            SERVICE-CICS-TABLE
DC C'D'             DOR
DC C'S'             SHARDED (QOR)
DC C'T'             TOR
DC C'V'             FOR
DC C'*'
PGMTAB DS ØCL8            TABLE FOR PROGRAMS, WHICH
DC CL8'CEE'         LE/370
DC CL8'CI'          APPLICATION-NAME
DC CL8'DN'          APPLICATION-NAME
DC CL8'DZ'          APPLICATION-NAME
DC CL8'DØ'          APPLICATION-NAME
DC CL8'D1'          APPLICATION-NAME
DC CL8'D2'          APPLICATION-NAME
DC CL8'D3'          APPLICATION-NAME
DC CL8'D4'          APPLICATION-NAME
DC CL8'D5'          APPLICATION-NAME
DC CL8'D6'          APPLICATION-NAME
DC CL8'D7'          APPLICATION-NAME
DC CL8'D8'          APPLICATION-NAME
DC CL8'D9'          APPLICATION-NAME
DC CL8'EDC'         C
DC CL8'FSN'         ASF
DC CL8'IBM'         PL1
DC CL8'IED'         C
DC CL8'IGZ'         COBOL
DC CL8'IIBM'        PL1
DC CL8'IIGZ'        COBOL
DC CL8'KA'          CSP 4.1
DC CL8'KL'          CSP 4.1
DC CL8'KS'          CSP 4.1
DC CL8'KT'          CSP 4.1
DC CL8'KU'          CSP 4.1
DC CL8'MCP'         APPLICATION-NAME
DC CL8'MSD'         APPLICATION-NAME
DC CL8'MSG'         APPLICATION-NAME
DC CL8'MSU'         APPLICATION-NAME
DC CL8'ZB'          CSP 4.1
PGMTABE DC C'*'       ARE SUPPORTED BY AUTOINST.
PGMCNT EQU (PGMTABE-PGMTAB)/8
MAPTAB DS ØCL8            TABLE FOR MAPS, WHICH
DC CL8'CM'          MAP-NAME
DC CL8'CP'          MAP-NAME
DC CL8'SD'          MAP-NAME
DC CL8'SU'          MAP-NAME
DC CL8'TS'          MAP-NAME
MAPTABE DC C'*'       ARE SUPPORTED BY AUTOINST.
MAPCNT EQU (MAPTABE-MAPTAB)/8
PGMTABS DS ØCL8            TABLE FOR PROGRAMS, WHICH
Interfacing CICS to SMTP – part 2

This month we conclude the package to interface CICS with the SMTP mail capability of TCP/IP for MVS.

*   REGISTER USAGE:                                                  *
*     RØ                                                              *
*     R1                                                              *
*     R2                                                              *
*     R3   COMMAREA POINTER                                           *
*     R4   COMMAREA STOPPER                                           *
*     R5   BAL                                                        *
*     R6                                                              *
*     R7                                                              *
*     R8                                                              *
*     R9                                                              *
*     R1Ø                                                             *
*     R11  EIB BASE REGISTER                                          *
*     R12  CODE BASE REGISTER                                         *
*     R13  DATA BASE REGISTER                                         *
*     R14                                                             *
*     R15                                                             *
***********************************************************************
SPACE 2
EJECT
***********************************************************************
* PROGRAM VARIABLES *
***********************************************************************
DFHEISTG DSECT
   SPACE 1
RESP   DS   F
TOKEN   DS   CL8
WORKLINE DS   CL80
   SPACE 1
***********************************************************************
* COMAREA DSECT *
***********************************************************************
COMAREA DSECT
CARCPT   DS   CL40
CAFROM   DS   CL40
ORG   COMMAREA
CAMSG   DS   CL80
CANEXT   EQU  *
   ORG   COMMAREA
CARESP   DS   XL1
   ORG
   SPACE 3
***********************************************************************
EJECT
SENDMAIL DFHEIENT CODEREG=(12),DATAREG=(13),EIBREG=(11)
SENDMAIL AMODE 31
SENDMAIL RMODE ANY
XEQU
EJECT
SPACE 2
START DS   0H
   ** INITIALIZATION **
L   R3,DFHEICAP GET ADDRESS OF COMMAREA
LA   R3,0(,R3) CLEAR VL bit
LTR   R3,R3 Q-IS COMMAREA AVAILABLE
BZ   RETURN8 BIF NO
   USING COMMAREA,R3 SET ADDRESSABILITY
   CLC EIBCALLEN=H'400' Q-IS COMMAREA OF PROPER LENGTH
   BL   RETURN8 YES
   LR   R4,R3 ADDRESS THE
   AH   R4,EIBCALLEN END OF THE COMMAREA
* BUILD EMAIL MESSAGES
   EXEC CICS SPOOLOPEN OUTPUT TOKEN(TOKEN) USERID(TCPSMTP) *
   NODE(NODE) CLASS(CLASSB) NOCC PUNCH RESP(RESP)
   CLC   RESP,DFHRESP(NORMAL)
   BNE   RETURN8
   MVC   WORKLINE,HELLO
   BAL   R5,WRITE
   MVC   WORKLINE,MAILFROM
   MVC   WORKLINE+12(L'CAFROM),CAFROM
   BAL   R5,WRITE
MVC WORKLINE,RCPTTO
MVC WORKLINE+10(L’CARCPT),CARCPT
BAL R5,WRITE
MVC WORKLINE,DATA
BAL R5,WRITE
LA R3,CANEXT
SENDLOOP DS ØH
MVC WORKLINE,CAMSG
BAL R5,WRITE
LA R3,CANEXT
CR R3,R4
BL SENDLOOP
EXEC CICS SPOOLCLOSE TOKEN(TOKEN) KEEP RESP(RESP)
CLC RESP,DFHRESP(NORMAL)
BE RETURN
RETURNø DS ØH ** INVALID COMMAREA LENGTH **
MVI CARESP,X’FF’ SET RC=ERROR
RETURN DS ØH
EXEC CICS RETURN
WRITE DS ØH
EXEC CICS SPOOLWRITE TOKEN(TOKEN) FROM(WORKLINE) RESP(RESP)
CLC RESP,DFHRESP(NORMAL)
BNE RETURNø
BR R5
SPACE 2
***********************************************************************
EJECT
***********************************************************************
*        P R O G R A M     C O N S T A N T S                          *
***********************************************************************
SPACE 1
TCPSMTP DC CL8’TCPSMTP’
NODE DC CL8’HLNCTR’ <=== change this to your JES node
CLASSB DC CL1’B’ <=== verify the class is OK
HELLO DC CL8’HELO HLNCTR’ <=== change this to your jes node
MAILFROM DC CL8’MAIL FROM: <12345678901234567890123456789012345678901234567890>* Ø’
RCPTTO DC CL8’RCPT TO: <12345678901234567890123456789012345678901234567890>* Ø’
DATA DC CL8’DATA’
SPACE 3
***********************************************************************
SPACE 1
END

MAILMAP.BMS
MAILMAP DFHMSD TYPE=DSECT, C
     MODE=INOUT, C
     LANG=COBOL, C

TIOAPFX=YES

CONFIRM  DFHMDI COLUMN=Ø22,LINE=Ø09,
CTRL=(FRSET,FREEKB),
SIZE=(Ø04,Ø32)
DFHMDF LENGTH=Ø31,POS=(Ø01,Ø01),
INITIAL='+-----------------------------+',
ATTRB=(ASKIP,NORM)

ERROR   DFHMDI COLUMN=Ø22,LINE=Ø09,
CTRL=(FRSET,FREEKB),
SIZE=(Ø04,Ø39)
DFHMDF LENGTH=Ø38,POS=(Ø01,Ø01),
INITIAL='+------------------------------------+',
ATTRB=(ASKIP,NORM)

MAILMAP  DFHMDI COLUMN=NEXT,LINE=NEXT,
CTRL=(FRSET,FREEKB),
SIZE=(Ø24,Ø80)
MAIL_DATE DFHMDF LENGTH=Ø10,POS=(Ø01,Ø01),
INITIAL='MM/DD/YYYY',
ATTRB=(ASKIP,NORM)
MAIL_APPLID DFHMDF LENGTH=ØØ8,POS=(Ø01,Ø71),
INITIAL='APPLID  ',
ATTRB=(ASKIP,NORM)
MAIL_TIME DFHMDF LENGTH=Ø10,POS=(Ø01,Ø01),
INITIAL='HH:MM:SS',
ATTRB=(ASKIP,NORM)
MAIL_USERID DFHMDF LENGTH=Ø10,POS=(Ø01,Ø01),
INITIAL='USERID  ',
ATTRB=(ASKIP,NORM)
MAIL_TO DFHMDF LENGTH=Ø40,POS=(Ø01,Ø11),
INITIAL='                                        ',
ATTRB=(UNPROT,NORM,FSET,IC)
DFHMDF LENGTH=Ø00,POS=(Ø04,Ø52),
ATTRB=(ASKIP,NORM)
DFHMDF LENGTH=024,POS=(004.053), INITIAL='<< Recipient''s Email ID', ATTRB=(ASKIP,NORM)

DFHMDF LENGTH=009,POS=(005.001), INITIAL='Reply to:', ATTRB=(ASKIP,NORM)

MAIL_REPLY DFHMDF LENGTH=040,POS=(005.011), INITIAL=' ', ATTRB=(UNPROT,NORM,FSET)

MAIL_FROM DFHMDF LENGTH=020,POS=(006.011), ATTRB=(UNPROT,NORM,FSET)

MAIL_SUBJECT DFHMDF LENGTH=069,POS=(007.011), ATTRB=(UNPROT,NORM,FSET)

MAIL_BODY DFHMDF LENGTH=079,POS=(009.001), INITIAL=' ', ATTRB=(UNPROT,NORM,FSET)

ARRY001 DFHMDF LENGTH=079,POS=(010.001), INITIAL=' ', ATTRB=(UNPROT,NORM,FSET)

ARRY002 DFHMDF LENGTH=079,POS=(011.001), INITIAL=' ', ATTRB=(UNPROT,NORM,FSET)

ARRY003 DFHMDF LENGTH=079,POS=(012.001), INITIAL=' ', ATTRB=(UNPROT,NORM,FSET)

ARRY004 DFHMDF LENGTH=079,POS=(013.001), INITIAL=' ',
ARRY005 DFHMDF LENGTH=079, POS=(014,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY006 DFHMDF LENGTH=079, POS=(015,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY007 DFHMDF LENGTH=079, POS=(016,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY008 DFHMDF LENGTH=079, POS=(017,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY009 DFHMDF LENGTH=079, POS=(018,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY010 DFHMDF LENGTH=079, POS=(019,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY011 DFHMDF LENGTH=079, POS=(020,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY012 DFHMDF LENGTH=079, POS=(021,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY013 DFHMDF LENGTH=079, POS=(022,001),
   INITIAL=' ',
   ATTRB=(UNPROT,NORM,FSET)

ARRY014 DFHMDF LENGTH=079, POS=(023,001),
   ATTRB=(UNPROT,NORM,FSET)
   DFHMDF LENGTH=003, POS=(024,001),
   INITIAL='PF:',
   ATTRB=(ASKIP,NORM)

MAIL_F1 DFHMDF LENGTH=005, POS=(024,005),
   INITIAL='1=Hlp',
   ATTRB=(ASKIP,NORM)

MAIL_F2 DFHMDF LENGTH=005, POS=(024,011),
   INITIAL='2=???',
   ATTRB=(ASKIP,DRK)

MAIL_F3 DFHMDF LENGTH=005, POS=(024,017),
   INITIAL='3=End',
   ATTRB=(UNPROT,NORM,FSET)
The advent of TCP/IP on OS/390 allowed the use of SMTP mail from TSO and batch. We regularly use the SMTP process at the tail end of batch jobs to facilitate mailing reports or abend notifications to designated users. The following example shows how to use the IEBGENER utility to send an e-mail message from a batch job. The message could be entirely inline or included from concatenated DD statements. The data in this example is formatted as 80-byte records; however, I have successfully tested it with records up to 512 bytes in length.

To accomplish the same thing from CICS, you could simply code up the JCL in working storage and submit it as a batch job. JCL submission is usually accomplished via extra-partition transient data queue output going to a DD statement defined with //JCL DD SYSOUT=(B,INTRDR). Alternatively, you can use the CICS spool interface to queue the e-mail message directly to the TCPSMTP program.
The CICS SENDMAIL program takes as input a variable length COMMAREA consisting of 80-byte records containing the recipient and sender e-mail addresses and the message to be sent. The program then formats this information into acceptable input to the TCPSMTP process and spools it out to JES. TCPSMTP does the rest.

To properly utilize this program you must have TCPSMTP properly configured on your OS/390 system. You must also have the CICS SPOOL interface active by specifying SPOOL=YES in the SIT or as a start-up override. Following the SENDMAIL program is a small COBOL program and BMS map source that can be used to test the SENDMAIL function from a formatted screen.

```
//IEBGENER EXEC PGM=IEBGENER
//SYSIN DD DUMMY
/**
//SYSUT1 DD *
HELO HLNCTR
MAIL FROM: <CZØØ55@HLNCTR.STATE.MT.US>
RCPT TO: <DGRINSELL@STATE.MT.US>
DATA
TO: Donald Grinsell
FROM: "Grinsell, Donald"<czØØ55@hlnctr.state.mt.us>
SUBJECT: TCP/IP Mail from MVS batch on SYSA

This is a test message from batch.

/*
// DD DISP=SHR,DSN=MORE.EMAIL.DATA
/**
//SYSUT2 DD SYSOUT=(B,TCPSMTP)
/**

Editor’s note: the MAILMAP BMS from page 11 goes here.

The mailtext COBOL program (from page 42 of last month’s issue) goes next.

The send mail Assembler program, started at the end of last month’s issue and concluded at the start of this article, goes last.

Donald A Grinsell
CICS Systems Programmer
State of Montana (USA) © Donald A Grinsell 2000
TCP/IP and CICS sockets

At our installation we needed to develop a way for a batch program to communicate with CICS without using APPC. We were interested in a solution which extends beyond our mainframe environment that would or could include alternative platforms such as the RS/6000. We have now accomplished our mission by developing a CICS sockets child server program named TCHSRV1. When we receive an incoming request on a CICS Internet socket from a client, CICS sockets starts our CICS child server transaction SRV1, as indicated by the first four bytes of the incoming datastream. Program TCHSRV1 begins processing by retrieving the CICS sockets parameters and taking control of the socket from CICS sockets. Program TCHSRV1 will acquire a work area and retrieve a 4000-byte data area. From this it will extract the program name which is to process the user data and link to that program with the user data pointed to as a COMMAREA after writing a link message to the CICS log. After the linked-to program has completed processing, the modified user data area and response codes are also written to the CICS log. Finally, program TCHSRV1 sends the modified user data back to the client, closes the socket, and returns control to CICS.

The TCHSRV1 and TCHCLOSE CICS log messages are shown below:

```
TCHSRV1 - LINK TO PROGRAM TCHCLOSE
TCHCLOSE - FILE PROTHLF  CLOSED / DISABLED RESP CODE = 00000000
TCHCLOSE - FILE PROTSYM  CLOSED / DISABLED RESP CODE = 00000000
TCHCLOSE - FILE TESTFILE CLOSED / DISABLED RESP CODE = 0000012
TCHCLOSE RESPONSE CODE = 0000012
PROTHLF  RESP=00000000
PROTSYM  RESP=00000000
TESTFILE RESP=0000012
```

The batch OS/390 client program that we are using to communicate with CICS T/S 1.3 is named TCHTCPIP. This program begins by reading the target IP address and port number from the job step parameters, the user’s data from the SYSIN DD is then stored in a GETMAINed storage area, and the TCP/IP API is initialized. Next, TCHTCPIP obtains a TCP/IP socket, connects to that socket, and gets the name of the remote socket to which the local socket is connected.
The user’s data is now sent to the target CICS system and the modified user data is read back after being processed and written out to the TCHTCPPIP job’s sysout DD. Finally, this program issues a shutdown to terminate all communications, closes the socket, and terminates the TCP/IP API. Prior to passing control back to OS/390, the highest processing return code is retrieved and placed in register 15.

An example of TCHTCPPIP SYSOUT messages resulting in the following job completing processing with a condition code of 12 is:

```
TCHCLOSE RESPONSE CODE = 00000012
PROTHLF RESP=00000000
PROTSYM RESP=00000000
TESTFILE RESP=00000012
```

The following is an example of a job that can be submitted to execute the above documented programs and processes. In this example a CICS program, TCHCLOSE, is called to close files PROTHLF, PROTSYM, and TESTFILE.

```
//JOBCARD
//*********************************************************************
//*      T C P / I P   -   C I C S   C L I E N T   I N T E R F A C E  *
//*        JOBSTEP PARMS -                                            *
//*       ENCLOSED IN SINGLE QUOTES MUST BE THE DECIMAL IP ADDRESS    *
//*          IMMEDIATELY FOLLOWED BY A COMMA AND THE DECIMAL          *
//*          TARGET PORT.                                             *
//*        SYSIN CARDS -                                              *
//*          THE FIRST CARD MUST BE THE CICS TRANSACTION NAME OF THE  *
//*          CHILD CICS SERVER TO BE STARTED BY CICS SOCKETS.         *
//*          THE SECOND CARD MUST BE THE NAME OF THE CICS PROGRAM     *
//*          THAT IS TO BE LINKED TO FOR DATA PROCESSING.             *
//*          THE THIRD TO THE FIFTIETH CARDS ARE THE DATA INPUT TO    *
//*          BE USED BY THE PROGRAM DEFINED IN CARD TWO VIA A CICS    *
//*          COMMAREA.                                                *
//*********************************************************************
//TCHTCPPIP EXEC PGM=TCHTCPPIP,
//         PARM='123.456.789.Ø12,1111'
//STEPLIB     DD DISP=SHR,DSN=CICS.TEST.LOADLIB
//SYSPRINT    DD SYSOUT=T,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYSUDUMP    DD SYSOUT=T
//SYSPRINT    DD SYSOUT=T
//SYSOUT      DD SYSOUT=T
//SNAP        DD SYSOUT=T
//SYSIN       DD *
SRV1
TCHCLOSE
```
The following program, TCHSRV1, executes within the CICS TS address space with CICS sockets:

```
TITLE 'TCHSRV1 - CICS TCPIP SERVER PROGRAM'
***********************************************************************
* TECHNICAL SUPPORT - DAVE MUNGER                                     *
* MODULE NAME - TCHSRV1                                               *
* LANGUAGE - ASSEMBLER                                                *
* DESCRIPTION - THIS PROGRAM EXECUTES AS A CHILD SERVER FOR TCPIP     *
*               CICS SOCKETS REQUESTS.                                *
*               INPUT FORMAT -                                        *
*               LINE 1 - PROGRAM NAME TO EXECUTE VIA A CICS LINK       *
*               LINE 2 - 49 PROGRAM DATA                             *
*               OUTPUT FORMAT -                                      *
*               LINE 1 - RETURNS HIGHEST RETURN CODE                  *
*               LINE 2 - 49 PROGRAM DATA WITH RETURN CODE             *
***********************************************************************

DFHEJECT
DFHREGS
DFHEISTG
DBLWRD   DS    D                  DATA CONVERSION DOUBLEWORD
PARMLIST DS    3ØA                TCP/IP PARAMETER LIST
*        C I C S   S O C K E T   P A R A M E T E R S
INPUT    DS    ØCL72
DESCRIPT DS    F                  SOCKET DESCRIPTOR
CICSNAME DS    CL8                NAME OF CICS LISTNER
TASKID   DS    CL8                LISTNER'S TASK IDENTIFIER
DATAAREA DS    CL35               INPUT AREA FROM LISTNER
   DS    CL1                RESERVED
FAMILY   DS    H                  TCP/IP ADDRESSING FAMILY
PORT     DS    H                  TCP/IP PORT NUMBER
IPADDR   DS    F                  TCP/IP ADDRESS
   DS    D                RESERVED
   DS    CL8                UNUSED
*        T A K E   S O C K E T   P A R A M E T E R S
TSOCKET  DS    ØCL4Ø               SOCKET DESCRIPTOR
DOMAIN   DS    F                  AF_INET
TNAME    DS    CL8                NAME OF CICS LISTNER
TTASK    DS    CL8                LISTNER'S TASK IDENTIFIER
   DS    CL2Ø               RESERVED
FUNCTION DS    CL16               TCP/IP FUNCTION
RETCODE  DS    F                  RETURN CODE
ERRNO    DS    F                  ERROR NUMBER
RESP     DS    F                  CICS RESPONSE CODE
SOCKET   DS    H                  SOCKET DESCRIPTOR
LENGTH   DS    H                  LENGTH FIELD
```

BUFFER  DS    F                  BUFFER ADDRESS
OUTPUT  DS    CL8Ø               OUTPUT BUFFER
PROGRAM  DS    CL8                PROGRAM NAME
TCHSRV1 AMODE  31
TCHSRV1 RMODE  ANY
TCHSRV1 CSECT
B     START              MODULE HISTORY
DC    CL8'TCHSRV1'       MODULE NAME
DC    CL8'VER - 01'      MODULE VERSION NUMBER
DC    CL8'&SYSDATE'      SYSTEM DATE
DC    CL8'&SYSTIME'      SYSTEM TIME
START    EQU   * 
MVC   LENGTH(2),=H'72'   SET LENGTH = 72
EXEC CICS RETRIEVE INTO(INPUT) LENGTH(LENGTH) RESP(RESP)
MVC   FUNCTION(16),TAKESOCK SET FUNCTION
MVC   SOCKET(2),DESCRIPT+2  SET SOCKET DESCRIPTOR
MVC   TDOMAIN(4),=F'2'      SET IF_INET
MVC   TNAME(8),CICSNAME SET NAME OF CICS LISTNER
MVC   TTASK(8),TASKID     SET LISTNER'S TASK IDENTIFIER
CALL EZASOKET,(FUNCTION,SOCKET,TSOCKET,ERRNO,RETCODE),        X
VL,MF=(E,PARMLIST)
CLI   RETCODE,X'FF'         Q. ANY ERRORS DETECTED ?
BE    ERROR                 A. YES - GO TO ERROR RTN
MVC   SOCKET(2),RETCODE+2  GET NEW SOCKET DESCRIPTOR
EXEC CICS GETMAIN SET(R4) FLENGTH(4000) INITIMG(=X'00')
ST    R4,BUFFER             SAVE BUFFER ADDRESS
MVC   FUNCTION(16),READX    SET FUNCTION = READ
MVC   LENGTH(2),=H'4000'    SET READ LENGTH = 4000
MVC   SOCKET(2),=H'4000'    GET NEW SOCKET DESCRIPTOR
CALL EZASOKET,(FUNCTION,SOCKET,LENGTH,R4),, X
ERRNO,RETCODE,VL,MF=(E,PARMLIST)
CLI   RETCODE,X'FF'         Q. ANY ERRORS DETECTED ?
BE    ERROR                 A. YES - GO TO ERROR RTN
L     R4,BUFFER             R4 => BUFFER DATA
LA    R4,30(R4)             R4 => USER DATA
XC    OUTPUT(8Ø),OUTPUT     CLEAR OUTPUT BUFFER
MVC   OUTPUT(L'LINKMSG'),LINKMSG SET LINK MESSAGE INTO BUFFER
MVC   OUTPUT+L'LINKMSG(8),R4 SET PROGRAM NAME INTO BUFFER
MVC   PROGRAM(8),R4 SET PROGRAM NAME FOR CICS LINK
EXEC CICS WRITEQ TD QUEUE('CSSL') FROM(OUTPUT) LENGTH(8Ø) X
RESP(RESP)
EXEC CICS LINK PROGRAM(PROGRAM) COMMAREA(Ø(R4)) LENGTH(4000) X
RESP(RESP)
L     R5,=F'49'             R5 = 49
LOOP     EQU   * 
EXEC CICS WRITEQ TD QUEUE('CSSL') FROM(Ø(R4)) LENGTH(8Ø) X
RESP(RESP)
LA    R4,8Ø(R4)             R4 => USER DATA
CLI   Ø(R4),X'40'          Q. SPACES FOUND ?
BE    WRITE                 A. YES - GO WRITE RESP TO CLIENT
CLI   Ø(R4),X'00'          Q. LOW VALUES FOUND ?
BE    WRITE                 A. YES - GO WRITE RESP TO CLIENT
BCT R5,LOOP          LOOP CONTROL

WRITE EQU *            WRITE EQUATION

MVC FUNCTION(16),WRITEX SET FUNCTION = WRITE
L   R4,BUFFER         R4 => DATA BUFFER
MVC LENGTH(2),=H'4000' LENGTH = DATA BUFFER LENGTH
CALL EZASOKET,(FUNCTION,SOCKET,LENGTH,Ø(R4), X
  ERRNO,RETCODE),VL,MF=(E,PARMLIST)
CLI RETCODE,X'FF'     Q. ANY ERRORS DETECTED ?
BE ERROR              A. YES - GO TO ERROR RTN
MVC FUNCTION(16),CLOSEX SET FUNCTION = CLOSE

CLOSE EQU *            CLOSE EQUATION

CALL EZASOKET,(FUNCTION,SOCKET,ERRNO,RETCODE), X
  VL,MF=(E,PARMLIST)
B   EXIT              GO EXIT

**********************************************************************
*        E R R O R   R O U T I N E                                    *
**********************************************************************

ERROR EQU *            ERROR EQUATION

MVI OUTPUT,X'40'       CLEAR OUTPUT
MVC OUTPUT+1(79),OUTPUT BUFFER
MVC OUTPUT(61),MESSAGE SET MESSAGE INTO OUTPUT BUFFER
MVC OUTPUT+100(16),FUNCTION SET FUNCTION INTO OUTPUT BUFFER
MVI OUTPUT+36,C'+' SET RETURN CODE SIGN
L   R5,RETCODE         Q. RETURN CODE POSITIVE ?
LTR R5,R5              SET CONDITION CODE
BNM POSITIVE           A. YES - BYPASS SETTING NEGATIVE
MVI OUTPUT+36,C'-' A. NO - SET RESPONSE CODE NEGATIVE

POSITIVE EQU *         POSITIVE EQUATION

CVD R5,DBLWRD          CONVERT RESPONSE CODE TO DECIMAL
UNPK OUTPUT+37(7),DBLWRD+4(4) UNPACK RESPONSE CODE
OI OUTPUT+43,X'F0' FORCE NUMERIC ZONE
L   R5,ERROR           R5 = ERROR NUMBER
CVD R5,DBLWRD          CONVERT ERROR NUMBER TO DECIMAL
UNPK OUTPUT+51(7),DBLWRD+4(4) UNPACK ERROR CODE
OI OUTPUT+57,X'F0' FORCE NUMERIC ZONE
EXEC CICS WRITEQ TD QUEUE('CSSL') FROM(OUTPUT) LENGTH(80) X
  RESP(RESP)
CLC FUNCTION(16),TAKESOCK Q. TAKESOCKET ERROR ?
BE   EXIT              A. YES - GO EXIT
CLC FUNCTION(16),READX Q. READ ERROR ?
BE   CLOSE             A. YES - GO CLOSE SOCKET
CLC FUNCTION(16),WRITEX Q. WRITE ERROR ?
BE   CLOSE             A. YES - GO CLOSE SOCKET
CLC FUNCTION(16),CLOSEX Q. CLOSE ERROR ?
BE   EXIT              A. YES - GO EXIT

EXIT EQU *             EXIT EQUATION

EXEC CICS RETURN
EJECT
* PROGRAM VARIABLES *
***********************************************************************
LINKMSG DC CL26'TCHSRV1 - LINK TO PROGRAM '
TAKESOCK DC CL16'TAKESOCKET'
READX DC CL16'READ'
WRITEX DC CL16'WRITE'
CLOSEX DC CL16'CLOSE'
MESSAGE DC CL10'TCHSRV1 :
   DC CL16'
   DC CL10' RETCODE= '
   DC CL8'
   DC CL10' ERRNO='
   DC CL7'
LTORG
END

The following program, TCHTCPIP, executes within a batch OS/390 job:

TITLE 'TCHTCPIP - TCP/IP BATCH - CICS INTERFACE'
EJECT
YREGS
EJECT
***********************************************************************
* TECHNICAL SUPPORT - DAVE MUNGER                                     *
* MODULE NAME - TCHTCPIP                                              *
* LANGUAGE - ASSEMBLER                                                *
* DESCRIPTION - THIS PROGRAM READS IN THE JOBSTEP PARAMETERS AND      *
  * INITIATES A TCP/IP CONVERSATION WITH CICS VIA CICS Sockets. The jobstep sysin data is read and sent to cics for processing. After CICS processing, a data buffer is returned to this program and the highest recorded return code from CICS processing is used as the return code of this program. *
***********************************************************************
TCHTCPIP CSECT
TCHTCPIP AMODE 24
TCHTCPIP RMODE 24
STM R14,R12,12(R13) STORE ENTRY REGISTERS
BALR R3,0 SET FIRST BASE REGISTER
USING *,R3,R4 ESTABLISH ADDRESSABILITY
BASE1 L R4,BASE2 SET SECOND BASE REGISTER
B CONTINUE BRANCH AROUND EYE CATCHER
DC CL8'TCHTCPIP' MODULE NAME
DC CL8'Ver - 01' MODULE VERSION NUMBER
DC CL8'&SYSDATE' SYSTEM DATE
DC CL8'&SYSTIME' SYSTEM TIME
BASE2 DC A(BASE1+4096) 2ND BASE REGISTER ADDRESS
CONTINUE ST R13,SAVE+4 STORE MVS SAVE AREA ADDRESS
LA R13,SAVE R13 => NEW SAVE AREA
EJECT
INITIALIZATION

SPACE 2
ST R1,PARMLST  SAVE PARAMETER LIST POINTER ADDRESS
MVC STEPRC(4),FULLWRD0 INITIALIZE STEP RETURN CODE
GETMAIN R,LV=4000
LTR R15,R15  Q. GETMAIN SUCCESSFUL ?
BNZ STGERR  A. NO - GO TO ERROR RTN
LR R6,R1  R6 => GETMAIN BUFFER AREA
ST R6,BUFFER  SAVE SEND BUFFER ADDRESS
L R7,=F'4000'  R7 = DATA LENGTH
LR R8,R6  R8 => DATA BUFFER
SR R9,R9  R9 = ZERO
MVCL R6,R8  INITIALIZE DATA BUFFER
EJECT

JOB STEP PARAMETER PROCESSING

SPACE 2
L R5,PARMLST  R5 => INPUT PARAMETER AREA
LH R5,2(R5)  R5 => INPUT PARM AREA
CLC Ø(2,R5),=XL2'0000'  Q. INPUT PARM SUPPLIED ?
BE PARMERR1  A. NO - GO TO ERROR RTN
LA R7,2(R5)  R7 => BEGINNING OF JOB STEP PARMS
LH R6,Ø(R5)  R6 = INPUT PARAMETER AREA LENGTH
AR R5,R6  R5 => END OF INPUT PARAMETER LIST
LA R5,2(R5)  ADD LENGTH OF LENGTH FIELD TO R5
LA R6,IPADDRS  R6 => IP ADDRESS FIELDS
L R8,=F'4'  R8 = LOOP CONTROL
EJECT

ADDRESS PROCESSING

SPACE 2
EQU *
IPLOOP  EQU *
XC NUMFIELD(4),NUMFIELD SET NUMFIELD = BINARY ZEROES
MVZ NUMFIELD(3),Ø(R7)  GET ZONES
CLC NUMFIELD(3),=XL3'F0F0F0'  Q. NUMERIC ZONES FOUND ?
BNE PARMERR2  A. NO - GO TO ERROR RTN
XC DBLWRD(8),DBLWRD SET DBLWRD = BINARY ZEROES
PACK DBLWRD+6(2),Ø(3,R7) PACK IP ADDRESS
CP DBLWRD+6(2),=PL2'255'  Q. VALID IP ADDRESS ?
BH PARMERR2  A. NO - GO TO ERROR RTN
CVB R9,DLBWRD  CONVERT IP ADDRESS TO BINARY
STCM R9,'X01',Ø(R6)  STORE IP ADDRESS
LA R6,1(R6)  INCREMENT OUTPUT IP ADDRESS PTR
LA R7,4(R7)  INCREMENT INPUT IP ADDRESS PTR
CR R7,R5  Q. END OF JOB STEP INPUT PARMS ?
BH PARMERR2  A. YES - GO TO ERROR RTN
BCT R8,IPLOOP  LOOP CONTROL

EJECT
***********************************************************************
*        PORT NUMBER PROCESSING                               *
***********************************************************************
SPACE 2
XC    NUMFIELD(4),NUMFIELD SET NUMFIELD = BINARY ZEROES
MVZ   NUMFIELD(4),Ø(R7)  GET ZONES
CLC   NUMFIELD(4),=XL4'FØFØFØFØ' Q. NUMERIC ZONES FOUND ?
BNE   PARMERR3           A. NO - GO TO ERROR RTN
XC    DBLWRD(8),DBLWRD SET DBLWRD = BINARY ZEROES
PACK  DBLWRD+5(3),Ø(4,R7) PACK IP ADDRESS
CVB   R9, DBLWRD CONVERT PORT NUMBER TO BINARY
STH   R9,PORT STORE PORT NUMBER
EJECT
OPEN  (SYSINDCB,(INPUT)) OPEN SYSIN DCB
LTR   R15,R15            Q. ANY ERRORS ?
BNZ   DCBERR1            A. YES - GO TO ERROR RTN
L     R5,=F'5Ø'          SET LOOP COUNTER
L     R6,BUFFER          R6 => SEND BUFFER
READ   EQU   *
GET   SYSINDCB           READ JOB INPUT
MVC   Ø(Ø(R6),Ø(R1)) PLACE COMMAND IN USER STORAGE
LA    R6,8Ø(R6)          INCREMENT COMMAND POINTER
BCT   R5,READ            LOOP CONTROL
CLOSEDCB EQU   *
CLOSE SYSINDCB           CLOSE DCB
EXTRACT TIOTTAB,FIELDS=(TIOT) GET TIOT ADDRESS
L     R1,TIOTTAB         R1 => TIOT ADDRESS
MVC   JOBNAME(8),Ø(R1)   SET JOBNAME
EJECT
***********************************************************************
*        CONNECT TO PROGRAM INTERFACE                           *
***********************************************************************
SPACE 2
MVC    MSGCMD(8),=CL8'INITAPI' SET TCP/IP COMMAND
EZASMI TYPE=INITAPI, ISSUE INITAPI MACRO X
SUBTASK=JOBNAME, SPECIFY SUBTASK IDENTIFIER X
MAXSOC=MAXSOC, SPECIFY MAXIMUM NUMBER OF SOCKETS X
MAXSNO=MAXSNO, RECEIVE SOCKET NUMBER ASSIGNED X
ERRNO=ERRNO, ERROR NUMBER FIELD X
RETCODE=RETCODE, RETURN CODE FIELD X
APITYPE=APITYPE AF_INET DEFAULT
CLI   RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE    ERROR A. YES - GO TO ERROR RTN
EJECT
***********************************************************************
*        OBTAIN TCP/IP SOCKET DESCRIPTOR                          *
***********************************************************************
SPACE 2
MVC    MSGCMD(8),=CL8'SOCKET ' SET TCP/IP COMMAND
EZASMI TYPE=SOCKET, ISSUE SOCKET MACRO X
AF='INET', INET X
SOCTYPE='STREAM', STREAM COMMUNICATIONS X
ERRNO=ERRNO, ERROR NUMBER FIELD X
RETCODE=RETCODE RETURN CODE FIELD

CLI RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE ERROR A. YES - GO TO ERROR RTN
MVC SOCKET(2),RETCODE+2 SET SOCKET DESCRIPTOR
EJECT

***********************************************************************
*        ISSUE CONNECT SOCKET                                    *
*                                                                      *
SPACE 2
MVC MSGCMD(8),=CL8'CONNECT' SET TCP/IP COMMAND
EZASMI TYPE=CONNECT, ISSUE CONNECT MACRO X
S=SOCKET, SOCKET DESCRIPTOR X
NAME=CONNPARM, SOCKET NAME STRUCTURE X
ERRNO=ERRNO, ERROR NUMBER FIELD X
RETCODE=RETCODE RETURN CODE FIELD

CLI RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE ERROR A. YES - GO TO ERROR RTN
EJECT

***********************************************************************
*        ISSUE GETPEERNAME                                      *
*                                                                      *
SPACE 2
MVC MSGCMD(8),=CL8'GETPEER' SET TCP/IP COMMAND
EZASMI TYPE=GETPEERNAME, ISSUE GETPEERNAME MACRO X
S=SOCKET, SOCKET DESCRIPTOR X
NAME=CONNPARM, SOCKET NAME STRUCTURE X
ERRNO=ERRNO, ERROR NUMBER FIELD X
RETCODE=RETCODE RETURN CODE FIELD

CLI RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE ERROR A. YES - GO TO ERROR RTN
EJECT

***********************************************************************
*        ISSUE WRITE DATA FROM BUFFER                           *
*                                                                      *
SPACE 2

MVC MSGCMD(8),=CL8'WRITE' SET TCP/IP COMMAND
L R5,BUFFER R5 => DATA BUFFER
EZASMI TYPE=WRITE, ISSUE WRITE MACRO X
S=SOCKET, SOCKET DESCRIPTOR X
NBYTE=4000, SIZE OF BUFFER X
BUF=(R5), BUFFER ADDRESS X
ERRNO=ERRNO, ERROR NUMBER FIELD X
RETCODE=RETCODE RETURN CODE FIELD

CLI RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE ERROR A. YES - GO TO ERROR RTN
EJECT

* ISSUE READ - READ DATA AND STORE IN BUFFER *

SPACE 2
MVC MSGCMD(8),=CL8'READ' SET TCP/IP COMMAND
L R6,BUFFER R6 => SEND BUFFER
L R7,='4000' R7 = DATA LENGTH
LR R8,R6 R8 => DATA BUFFER
SR R9,R9 R9 = ZERO
MVCL R6,R8 INITIALIZE DATA BUFFER
L R5,BUFFER R5 => BUFFER
EZASMI TYPE=READ, ISSUE READ MACRO
S=SOCKET, SOCKET DESCRIPTOR
NBYTE=4000, BUFFER SIZE
BUF=(R5), BUFFER ADDRESS
ERRNO=ERRNO, ERROR NUMBER FIELD
RETCODE=RETCODE RETURN CODE FIELD
CLI RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE ERROR A. YES - GO TO ERROR RTN

SYSOUT EQU *
OPEN (SYOUTD CB,(OUTPUT)) OPEN SYSOUT DCB
LTR R15,R15 Q. ANY ERRORS ?
BNZ DCBERR2 A. YES - GO TO ERROR RTN
L R5,='50' SET LOOP COUNTER
L R6,BUFFER R6 => BUFFER
LA R6,30(R6) R6 => USER DATA
PUT SYOUTDCB GET FIRST BUFFER ADDRESS

WRITE EQU *
MVC 0(80,R1),0(R6) SET DATA INTO BUFFER
LA R6,80(R6) INCREMENT DATA POINTER
CLI 0(R6),X'00' Q. DATA FOUND ?
BE CLOSEO A. NO - GO CLOSE DCB
CLI 0(R6),X'40' Q. DATA FOUND ?
BE CLOSEO A. NO - GO CLOSE DCB
PUT SYOUTDCB WRITE BUFFERED DATA
BCT R5,WRITE LOOP CONTROL

CLOSEO EQU *
CLOSE SYOUTDCB CLOSE DCB

EJECT

* ISSUE SHUTDOWN *

SPACE 2
SHUTDOWN EQU *
MVC MSGCMD(8),=CL8'SHUTDOWN' SET TCP/IP COMMAND
EZASMI TYPE=SHUTDOWN, ISSUE SHUTDOWN MACRO
S=SOCKET, SOCKET DESCRIPTOR
HOW=STOPALL, END COMMUNICATION
ERRNO=ERRNO, ERROR NUMBER FIELD
RETCODE=RETCODE RETURN CODE FIELD
CLI RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE    ERROR              A. YES - GO TO ERROR RTN
EJECT

***********************************************************************
*        C L O S E   C O N N E C T I O N                              *
***********************************************************************
SPACE 2
CLOSE    EQU *
MVC   MSGCMD(8),=CL8'CLOSE ' SET TCP/IP COMMAND
EZASMI TYPE=CLOSE, ISSUE CLOSE MACRO X
S=SOCKET, SOCKET DESCRIPTOR X
ERRNO=ERRNO, ERROR NUMBER FIELD X
RETCODE=RETCODE RETURN CODE FIELD
CLI   RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE    ERROR              A. YES - GO TO ERROR RTN
EJECT

***********************************************************************
*        T E R M I N A T E   C O N N E C T I O N   T O   A P I          *
***********************************************************************
SPACE 2
TERMAPI  EQU *
MVC   MSGCMD(8),=CL8'TERMAPI ' SET TCP/IP COMMAND
EZASMI TYPE=TERMAPI ISSUE TERMAPI MACRO
CLI   RETCODE,X'FF' Q. ANY ERRORS DETECTED ?
BE    ERROR              A. YES - GO TO ERROR RTN
EJECT

***********************************************************************
*        R E T U R N   T O   C A L L E R                                             *
***********************************************************************
SPACE 2
EXIT     EQU *
L     R6,BUFFER          R6 => DATA BUFFER
LA    R6,3Ø(R6)          R6 => USER DATA
L     R7,=F'1Ø'          R7 = 1Ø
EXITLOOP EQU *
CLC   Ø(16,R6),=CL16'RESPONSE CODE = '
BE    EXITEND
LA    R6,1(R6)           INCREMENT DATA POINTER
BCT   R7,EXITLOOP        LOOP CONTROL
B     FREEMAIN           GO FREEMAIN BUFFER
EXITEND  EQU *
LA    R6,16(R6)          R6 => RETURN CODE
XC    DBLWRD(8),DBLWRD SET DBLWRD = BINARY ZEROES
PACK   DBLWRD+3(5),Ø(8,R6) PACK SERVER RETURN CODE
CVB   R7,DBLWRD CONVERT RETURN CODE TO BINARY
C     R7,STEPRC          Q. SERVER RC > STEP RC ?
BNH   FREEMAIN           A. NO - GO FREEMAIN BUFFER
ST    R7,STEPRC          A. YES - SET NEW STEP RETURN CODE
FREEMAIN EQU *
L     R2,BUFFER          R2 => BUFFER
FREEMAIN R,LV=4ØØØ,A=(R2)
SETRC EQU *
L R15,STEPRC SET STEP RETURN CODE
L R13,SAVE+4 R13 => CALLER'S SAVE AREA
LM R0,R12,20(R13) RELOAD CALLER'S REGISTERS
L R14,12(13) R14 = CALLER'S RETURN ADDRESS
BR R14 RETURN TO CALLER
EJECT

***********************************************************************
*        ERROR ROUTINES                                               *
***********************************************************************

PARMERR1 EQU *
WTO 'JOB PARM LENGTH ERROR DETECTED - PROCESSING TERMINATED'
MVC STEP RC(4),=F'8' SET ERROR RETURN CODE
B FREEMAIN GO ISSUE FREEMAIN AND EXIT

PARMERR2 EQU *
WTO 'IP ADDRESS ERROR DETECTED - PROCESSING TERMINATED'
MVC STEPRC(4),=F'8' SET ERROR RETURN CODE
B FREEMAIN GO ISSUE FREEMAIN AND EXIT

PARMERR3 EQU *
WTO 'PORT NUMBER ERROR DETECTED - PROCESSING TERMINATED'
MVC STEPRC(4),=F'8' SET ERROR RETURN CODE
B FREEMAIN GO ISSUE FREEMAIN AND EXIT

STGERR EQU *
WTO 'GETMAIN ERROR DETECTED - PROCESSING TERMINATED'
MVC STEPRC(4),=F'8' SET ERROR RETURN CODE
B SETRC GO SET RETURN CODE

DCBERR1 EQU *
WTO 'DCB OPEN ERROR DETECTED - PROCESSING TERMINATED'
MVC STEPRC(4),=F'8' SET ERROR RETURN CODE
B FREEMAIN

DCBERR2 EQU *
WTO 'DCB OPEN ERROR DETECTED - PROCESSING TERMINATED'
MVC STEPRC(4),=F'8' SET ERROR RETURN CODE
B SHUTDOWN GO STOP ALL DATA COMMUNICATIONS

ERROR EQU *
MVI MSGRCS,C'+' SET RETURN CODE SIGN
L R5,RETCODE R5 = RETURN CODE
LTR R5,R5 Q. RETURN CODE POSITIVE ?
BNM POSITIVE A. YES - GO CONVERT RETURN CODE
MVI MSGRCS,C'-' A. NO - SET RETURN CODE SIGN TO NEG

POSITIVE EQU *
CVD R5,DBLWRD CONVERT RETURN CODE TO DECIMAL
UNPK MSGRC,DBLWRD+4(4) UNPACK RETURN CODE
OI MSGRC+6,X'F0' SET ZONE
L R5,ERRNO R5 = ERROR NUMBER
CVD R6,DBLWRD CONVERT ERROR NUMBER TO DECIMAL
UNPK MSGERROR,DBLWRD+4(4) UNPACK ERROR NUMBER
OI MSGERROR+6,X'F0' SET ZONE
WTO TEXT=MESSAGE
MVC  STEPRC(4),=F'8'    SET JOBSTEP RETURN CODE
CLC  MSGCMD(8),=CL8'INITAPI'  Q. INITAPI ERROR DETECTED ?
BE   FREEMAIN                 A. YES - GO FREEMAIN BUFFER
CLC  MSGCMD(8),=CL8'SOCKET '  Q. SOCKET ERROR DETECTED ?
BE   TERMAPI                  A. YES - GO TERMINATE API
CLC  MSGCMD(8),=CL8'CONNECT ' Q. CONNECT ERROR ?
BE   CLOSE                    A. YES - GO CLOSE SOCKET
CLC  MSGCMD(8),=CL8'GETPEER ' Q. GETPEER ERROR ?
BE   CLOSE                    A. YES - GO CLOSE SOCKET
CLC  MSGCMD(8),=CL8'WRITE   ' Q. WRITE ERROR ?
BE   SHUTDOWN                 A. YES - GO SHUTDOWN COMM
CLC  MSGCMD(8),=CL8'READ    ' Q. READ ERROR ?
BE   SHUTDOWN                 A. YES - GO SHUTDOWN COMM
CLC  MSGCMD(8),=CL8'SHUTDOWN' Q. SHUTDOWN ERROR ?
BE   CLOSE                    A. YES - GO CLOSE SOCKET
CLC  MSGCMD(8),=CL8'CLOSE   ' Q. CLOSE ERROR ?
BE   TERMAPI                  A. YES - GO TERMINATE API
CLC  MSGCMD(8),=CL8'TERMAPI ' Q. TERMAPI ERROR ?
BE   FREEMAIN                 A. YES - GO TO FREEMAIN
EJECT
***********************************************************************
*        G L O B A L   V A R I A B L E   S T O R A G E   A R E A      *
***********************************************************************
SPACE 2
EZASMGW  EZASMI TYPE=GLOBAL,STORAGE=CSECT
EJECT
***********************************************************************
*        T A S K   V A R I A B L E   S T O R A G E   A R E A          *
***********************************************************************
SPACE 2
EZASMI TYPE=TASK,STORAGE=CSECT
SPACE 2
DBLWRD   DS    D                  WORK AREA
SAVE     DS    18F                SAVE AREA
STEPRC   DS    F                  STEP RETURN CODE
TIOTTAB  DS    F                  TIOT ADDRESS
PARMLST  DS    F                  PARAMETER LIST POINTER
JOBNAME  DS    CL8                SUBTASK PARM VALUE
MAXSNO   DC    F'Ø'               (HIGHEST SOCKET DESCRIPTOR AVAILABLE)
MAXSOC   DC    H'1Ø'              MAXSOC PARM VALUE
SOCKET   DS    H                  PORT NUMBER
APITYPE  DC    H'2'               OR PUT A 3 HERE
CNOP  Ø,4
CONNPARM DS    ØCL16             SOCKET NAME STRUCTURE
   DC    AL2(2)             FAMILY
PORT     DS    H                  PORT NUMBER
IPADDRS  DS    F                  IP ADDRESS
   DC    XL8'Ø'             RESERVED
FULLWRDØ DC    F'Ø'               STEP RETURN CODE
BUFFER   DS    F                  GETMAINED SEND BUFFER ADDRESS
We use this process to close files for batch processing. The following program, **TCHCLOSE**, executes within the CICS TS address to close files:

```assembly
TITLE 'TCHCLOSE - FILE CLOSE PROGRAM'
EJECT
DFHREGS
EJECT
***********************************************************************
* TECHNICAL SUPPORT - DAVE MUNGER                                    *
* MODULE NAME - TCHCLOSE                                              *
* LANGUAGE - ASSEMBLER                                                *
* DESCRIPTION - THIS PROGRAM ADDRESSES A LIST OF FILES TO BE CLOSED   *
*               IN A COMMAREA AND ISSUES THE CICS CLOSE COMMANDS.       *
*               THE RESPONSE CODES FOR EACH FILE CLOSE COMMAND IS      *
*               PLACED TO THE RIGHT OF EACH FILE NAME IN THE COMMAREA. *
*               AS THIS PROGRAM CLOSES FILES THE HIGHEST RETURN CODE  *
*               RETURNED IS RECORDED IN THE FIRST 80-BYTE LOGICAL      *
*               RECORD IN THE COMMAREA.                               *
***********************************************************************
DFHEISTG DSECT ,
BCD DS D WORK AREA
RESPCODE DS F COMMAREA RECORD RETURN CODE
RESP DS F CICS RESPONSE CODE
FILENAME DS CL8 CICS FILE NAME
OUTPUT DS CL80 OUTPUT DATA BUFFER
TCHCLOSE CSECT
TCHCLOSE AMODE 31
TCHCLOSE RMODE ANY
```
L R4,DFHEICAP  R4 -> COMMAREA
XC RESPCODE(4),RESPCODE  RESPCODE = ZEROES

LOOP EQU *
LA R4,8Ø(R4)  R4 => FILE NAME
MVC FILENAME(8),Ø(R4)  SET FILENAME
CLI FILENAME,X'4Ø'  Q. FILENAME FOUND ?
BE EXIT  A. NO - GO EXIT
CLI FILENAME,X'0Ø'  Q. FILENAME FOUND ?
BE EXIT  A. NO - GO EXIT
EXEC CICS SET FILE(FILENAME) CLOSED DISABLED RESP(RESP)
MVC OUTPUT(L'MESSAGE),MESSAGE  SET MESSAGE INTO DATA BUFFER
MVC OUTPUT+16(8),FILENAME  SET FILENAME INTO MESSAGE
L R5,RESP  R5 = RESP
CVD R5,BCD  CONVERT TO DECIMAL
UNPK OUTPUT+55(8),BCD  SET RETURN CODE INTO OUTPUT
O1 OUTPUT+62,X'FØ'  SET SIGN
EXEC CICS WRITEQ TD QUEUE('CSSL') FROM(OUTPUT) LENGTH(8Ø)
MVC 9(R5,R4),=CL5'RESP='  SET 'RESP=' INTO COMMAREA
MVC 14(R5,R4),OUTPUT+55  SET RESP CODE INTO COMMAREA
CLC RESPCODE(4),RESP  Q. RESPCODE > RESP
BH LOOP  A. YES - CONTINUE PROCESSING
MVC RESPCODE(4),RESP  SET RESPCODE = RESP
B LOOP  CONTINUE PROCESSING FILE REQ

********************************************************************
* * *    E X I T   P R O C E S S I N G                         * * *
********************************************************************
SPACE
EXIT EQU *
L R4,DFHEICAP  R4 => COMMAREA
MVC Ø(25,R4),=CL25'TCHCLOSE RESPONSE CODE = '
L R5,RESPCODE  R5 = RESPCODE
CVD R5,BCD  CONVERT TO DECIMAL
UNPK 25(R5,R4),BCD  SET RETURN CODE INTO OUTPUT
O1 32(R5,R4),X'FØ'  SET SIGN
EXEC CICS RETURN

********************************************************************
* * *    P R O G R A M   C O N S T A N T S                     * * *
********************************************************************
SPACE
MESSAGE DC  CL63'TCHCLOSE - FILE XXXXXXXX CLOSED / DISABLED RESPX
           ONCE = XXXXXXXX'
END

David M Munger
Technical CICS Consultant
Time Customer Service (USA)

© David M Munger 2000
Resource definition display and alter commands – part 1

The resource definition displays provided by CICS standard facilities do not always provide the detail or the summary level information desired. Here is a utility that will list files by various categories and provide the ability to alter individual file and transaction attributes dynamically. It does not use RDO but the dynamic create facility introduced with CICS Version 4.

At the summary level (see Figure 1), files are grouped in the following categories:

• By first three bytes of the datasetname.
• By remote system.
• By LSR buffer pool or NSR.

From the summary screen an individual category can be selected using the cursor to display all the files within that category (see Figure 2). The file list can be browsed forwards and backwards.

Individual files can then be selected by cursor from the category lists in order to display the detailed file attributes (see Figure 3). If desired, the attributes can be altered. Files are automatically closed and re-opened in order to change attributes. If an existing file is entered as a remote name or vice versa at the detail level, the transaction will switch to the correct mode.

Useful applications of this utility include:

• Identifying files using NSR and moving them into LSR and back easily.
• Finding LRECL and KEYLENGTH easily when preparing remote definitions.
• Creating new files in test environments using existing models.
• Tuning LSR dynamically by moving files from one bufferpool to another.
<table>
<thead>
<tr>
<th>Dataset Type</th>
<th>Open</th>
<th>47</th>
<th>95</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>94</th>
<th>HLQ Total</th>
<th>HLQ Total</th>
<th>HLQ Total</th>
<th>HLQ Total</th>
<th>HLQ Total</th>
<th>HLQ Total</th>
<th>HLQ Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote</td>
<td>189</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>LSR1</td>
<td>LSR2</td>
<td>LSR3</td>
<td>LSR4</td>
<td>LSR5</td>
<td>LSR6</td>
<td></td>
</tr>
<tr>
<td>Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KPA</td>
<td>DFV</td>
<td>TPA</td>
<td>DCW</td>
<td>DFW</td>
<td>DFU</td>
<td></td>
</tr>
<tr>
<td>HLQ Total</td>
<td>167</td>
<td>3</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>SCW</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To Exit Clear/PF3 For details select HLQ/RSYS or LSRPOOL
### Figure 2: Selected detail display

**SYG9**  
Partition: pppp  
--- CICS Region: aaaaaaa  
DATE: 30/06/1999  
TIME: 13:19:00  
--- ccccccc ---  

<table>
<thead>
<tr>
<th>Remote datasets</th>
<th>4 DDAS 4</th>
<th>LSR1</th>
<th>LSR2</th>
<th>LSR3</th>
<th>LSR4</th>
<th>LSR5</th>
<th>LSR6</th>
<th>LSR7</th>
<th>LSR8</th>
<th>NSR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rsys Total</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Local datasets</th>
<th>189 Open</th>
<th>47</th>
<th>95</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>94</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files with HLQ of - &quot;KPA&quot;</td>
<td>Total 167</td>
<td>Total Open 38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fileid</th>
<th>Datasetname</th>
<th>Status</th>
<th>Key</th>
<th>Recl</th>
</tr>
</thead>
<tbody>
<tr>
<td>A000LOG</td>
<td>KPACO.DPACICS.EA000LOG.CLS</td>
<td>Open</td>
<td>N/A</td>
<td>4089</td>
</tr>
<tr>
<td>BDI</td>
<td>KPACO.DPACICS.KBIREC.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B000LOG</td>
<td>KPACO.DPACICS.EB000LOG.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAA</td>
<td>KPACO.DPACICS.ECAA.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAF</td>
<td>KPACO.DPACICS.KCAF.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAI</td>
<td>KPACO.DPACICS.KCAI.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAIH</td>
<td>KPACO.DPACICS.KCAIH.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAIP</td>
<td>KPACO.DPACICS.KCAI.PTH</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAM</td>
<td>KPACO.DPACICS.KCAM.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAMH</td>
<td>KPACO.DPACICS.KCAMH.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAP</td>
<td>KPACO.DPACICS.KCAP.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAPH</td>
<td>KPACO.DPACICS.KCAPH.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAS</td>
<td>KPACO.DPACICS.KCAS.CLS</td>
<td>Closed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CASH</td>
<td>KPACO.DPACICS.KCASH.CLS</td>
<td>Closed</td>
<td></td>
<td>FWD(PF8)</td>
</tr>
</tbody>
</table>

Return to summary with PF3/Clear, File Display/Alter PF5
Dynamic Create Facility

Transaction( ___ )  Program( ___ )  Taskdataloc( A )  DTimeout ( 300 )
SPurge ( Y )  TPurge ( Y )
TClass ( N )  TWASize ( 0000 )

Fileid ( COPOPT )  DSNName ( DFV.SOPV3011.COPOPT )
Recordformat( V )  STRNO( 02 )  LSRPool( 1 )
Status( E )  Recovery( N )
Browse( N )  Add( N )  Delete( N )  Update( N )

Remote ( ___ )  SYSid ( ___ )  KEYLen ( 255 )  RECLen ( 9999 )

File exists Press ENTER to replace
Enter Transaction, Fileid or Remote.

Clear or PF3 to EXIT, PF4 to CANCEL
The detail level transaction also allows transactions to be displayed, altered, and added. If TCLASS(Y) is chosen, a TRANCLASS is automatically created using the naming convention CMXTtttt, where tttt is the transaction identifier.

All parameters are validated and some rules applied. For example, transactions beginning ‘C’ or programs beginning ‘DFH’ are not allowed. Certain defaults are supplied for new definitions (which can be overridden). These defaults may differ from IBM defaults, for example TASKDATALOC takes the default A (Any).

When CICS create is used, all changes can be logged to the CSSL destination, which can act as an audit trail.

INSTALLING THE UTILITY
There are two Assembler programs, SYG8 and SYG9, and two maps, SYMSYG8 and SYMSYG9, that need to be defined to RDO. Two transactions, SYG8 and SYG9, should be defined pointing to programs SYG8 and SYG9 respectively.

The programs and maps supplied should be assembled/linked into an appropriate library in the CICS RPL chain.

SYG9 – FILE SUMMARY DISPLAY
Adjust the code in TOPMODE to suit your own LPAR naming convention.

*ASM XOPTS(SP)
SYG9 RMODE ANY
*------------------------------------------------------------------
*             PROGRAM      : SYG9
* DESCRIPTION  : This module counts and displays the number
*                 of local files, remote files, and open files.
*                 It also counts them by HLQ of the DSN,
*                 remote sysid, and LSRPOOL.
*                 HLQs LSRPOOLS and remote sysids can be selected by
*                 cursor in order to display a list of files.
*------------------------------------------------------------------
R1       EQU   1
R2       EQU   2
EIBREG EQU 3
DATAREG EQU 4
BASE1 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
RA EQU 10
LINKREG EQU 11
RC EQU 12
COPY DFHAID
DFHEISTG DSECT
COMMAS DS ØH
OFFSET DS H
TSTAT DS CL1
HLQPOS DS PL4
MOREBWD DS CL1
MOREFWD DS CL1
MODEFLG DS CL1
OPENSTAT DS F
FTYPE DS F
LSRNUM DS F
RSYS DS CL4
WDDN DS CL8
WDSN DS CL48
ATIME DS PL8
APPL DS CL8
DWORK DS D
INQKLEN DS F
INQRLEN DS F
INLENG DS H
TOTLOCL DS PL3
TOTOPEN DS PL3
TOTREMT DS PL3
TOTNSR DS PL3
TOTLSR1 DS PL3
TOTLSR2 DS PL3
TOTLSR3 DS PL3
TOTLSR4 DS PL3
TOTLSR5 DS PL3
TOTLSR6 DS PL3
TOTLSR7 DS PL3
TOTLSR8 DS PL3
COLCNT DS PL1
RSYSTAB DS CL35
HLQTAB DS CL168
HLQTAB1 DS CL168
HLQSEL DS CL3
RSYSSEL DS CL4
HLQCNTR DS PL3

**HLOCNT0** DS PL3
**FILETAB** DS CL980
**FILECNT** DS PL2
**FILPOS** DS PL4
**CF** DS CL9
COPY SYMSYG9
*
**SYG9** DFHEIENT CODEREG=(BASE1), EIBREG=(EIBREG), *
**DATAREG=(DATAREG)**

B BEGIN
DC CL12'PROGRAM ID: '
DC CL8'SYG9 '
DC CL4': '
DC CL24'ASSEMBLY TIME AND DATE:'
DC CL8'&SYSTIME'
DC CL8'&SYSDATE'
BEGIN DS ØH
CLC EIBCALEN,=H'Ø' Any Commarea?
BE BEGIN1 First Time Through
L RA,DFHEICAP Address Commarea
MVC COMMAS(10),Ø(RA) Move into dynamic
EXEC CICS HANDLE AID PF3(RETURN1) CLEAR(RETURN1) *
PF7(PAGEBWD) PF8(PAGEFWD)
EXEC CICS RECEIVE LENGTH(INLENG)
CLI TSTAT,X'2Ø' At detail level ?
BE UPORDWN Hold position
CLC EIBCPOSN,=H'344' Cursor before sysid area
BL BEGIN1 So ignore
CLC EIBCPOSN,=H'387' Cursor beyond sysid area
BH LSRCUR So check HLQ area
XR R8,R8 Clear offset reg
MVI MODEFLG,C'R' Set remote mode
CLC EIBCPOSN,=H'355' 1st sysid ?
BL SELEND Yes
LA R8,7(R8)
CLC EIBCPOSN,=H'366' 2nd sysid ?
BL SELEND Yes
LA R8,7(R8)
CLC EIBCPOSN,=H'377' 3rd sysid ?
BL SELEND Yes
LA R8,7(R8)
B SELEND Must be 4th then
LSRCUR DS ØH
CLC EIBCPOSN,=H'515' Cursor before LSR area
BL BEGIN1 So ignore
CLC EIBCPOSN,=H'559' Cursor beyond LSR area
BH HLOCUR So check HLQ area
XR R8,R8 Clear offset reg
MVI MODEFLG,C'L'               Set LSR mode
CLC EIBCP OSN,=H'554'           Cursor In NSR area
BH SELEND                      So Finished
LA R8,1(R8)                    LSRPOOL1
CLC EIBCP OSN,=H'520'           Cursor in LSR1 area
BL SELEND                      So Finished
LA R8,1(R8)                    LSRPOOL2
CLC EIBCP OSN,=H'525'           Cursor in LSR2 area
BL SELEND                      So Finished
LA R8,1(R8)                    LSRPOOL3
CLC EIBCP OSN,=H'530'           Cursor in LSR3 area
BL SELEND                      So Finished
LA R8,1(R8)                    LSRPOOL4
CLC EIBCP OSN,=H'535'           Cursor in LSR4 area
BL SELEND                      So Finished
LA R8,1(R8)                    LSRPOOL5
CLC EIBCP OSN,=H'540'           Cursor in LSR5 area
BL SELEND                      So Finished
LA R8,1(R8)                    LSRPOOL6
CLC EIBCP OSN,=H'545'           Cursor in LSR6 area
BL SELEND                      So Finished
LA R8,1(R8)                    LSRPOOL7
CLC EIBCP OSN,=H'550'           Cursor in LSR7 area
BL SELEND                      So Finished
LA R8,1(R8)                    LSRPOOL8
B    SELEND                    Finished

HLQCUR    DS   ØH
CLC EIBCP OSN,=H'641'           Cursor before HLQ area
BL BEGIN1                      So ignore
CLC EIBCP OSN,=H'182Ø'          Cursor After last HLQ
BH BEGIN1                      So ignore
XR R6,R6                       Clear R6
LH R7,EIBCP OSN                Cursor position
SH R7,=H'641'                  Minus lines above HLQ
XR R8,R8                       Clear R8 (offset)

SELEND    DS   ØH
CH R7,=H'159'                  On this line ?
BH SELNEXT                     Try next
CH R7,=H'69'                   Beyond HLQ on this line ?
BH BEGIN1                      So ignore
LH R9,=H'10'                   Divide by ten
DR R6,R9                       Multiply by six
AR R8,R7                       Add to offset
B    SELEND                    As if by magic !

SELEND    DS   ØH
SH R7,=H'160'                  Take off a couple of lines
LA R8,42(R8)                   Add in the offset
B    SELNEXT                   And round we go

STH R8,OFFSET                  Save for later
MVI TSTAT,X'10' Set flag for later
ZAP HLQPOS,=P'Ø' Set position
MVI MOREBWD,X'00' Clear flags
MVI MOREFWD,X'00'
B BEGIN2 Run normal Scan

PAGEBWD DS ØH
CLI TSTAT,X'20' Only at detail level
BNE BEGIN1 Ignore request
CLI MOREBWD,X'10' Are there any?
BNE UPORDWN No ignore request
SP HLQPOS,=P'14' Prev 14 Files
CP HLQPOS,=P'Ø' Now at start?
BNE UPORDWN No
MVI MOREBWD,X'00' Stop going back
B UPORDWN Common code

PAGEFWD DS ØH
CLI TSTAT,X'20' Only at detail level
BNE BEGIN1 Ignore request
CLI MOREFWD,X'10' Are there any?
BNE UPORDWN No ignore request
AP HLQPOS,=P'14' Next 14 Files
MVI MOREBWD,X'10' Must be now
MVI MOREFWD,X'00' Reset for counter

UPORDWN DS ØH
ZAP FILPOS,=P'Ø'
LA R8,FILETAB To store file info
ZAP FILECNT,=P'Ø' and a counter
MVI TSTAT,X'10'
B BEGIN2 Run normal Scan

BEGIN1 DS ØH
MVI TSTAT,X'00'

BEGIN2 DS ØH
ZAP TOTLOCL,=P'Ø' Zero Local count
ZAP TOTREMT,=P'Ø' Zero Remote count
ZAP TOTOPEN,=P'Ø' Zero Open count
ZAP TOTNSR,=P'Ø' Zero NSR count
ZAP TOTLSR1,=P'Ø' Zero LSR1 count
ZAP TOTLSR2,=P'Ø' Zero LSR2 count
ZAP TOTLSR3,=P'Ø' Zero LSR3 count
ZAP TOTLSR4,=P'Ø' Zero LSR4 count
ZAP TOTLSR5,=P'Ø' Zero LSR5 count
ZAP TOTLSR6,=P'Ø' Zero LSR6 count
ZAP TOTLSR7,=P'Ø' Zero LSR7 count
ZAP TOTLSR8,=P'Ø' Zero LSR8 count
MVC RSYSTAB,RSYITAB Initialize RSYSTAB
MVC HLQTAB(168),HLQITAB Initialize
MVC HLQTABLE(162),HLQITAB Table
EXEC CICS HANDLE CONDITION END(FILEND)
EXEC CICS INQUIRE FILE START

FILLOOP DS ØH
EXEC CICS INQUIRE FILE(WDDN) NEXT * 
DSNAME(WDSN) REMOTESYSTEM(RSYS) * 
OPENSTATUS(OPENSTAT) LSRPOOLID(LSRNUM) * 
KEYLENGTH(INQKLEN) RECORDSIZE(INQRLEN) * 
ACCESSMETHOD(FTYPE) 

CLC FTYPE,DFHVALUE(REMOTE) Is it Remote?  
BE CNTREM Count it  
AP TOTLOCL,=P'1' Must be Local then  
CLC LSRNUM,=F'1' LSR Pool count  
BNE CLSR8  
AP TOTLSR1,=P'1'  
B CNTOPEN  

CLSR8 DS øH  
CLC LSRNUM,=F'8' LSR Pool count  
BNE CLSRØ  
AP TOTLSR8,=P'1'  
B CNTOPEN  

CLSRØ DS øH  
CLC LSRNUM,=F'0' LSR Pool count  
BNE CLSR2  
AP TOTNSR,=P'1'  
B CNTOPEN  

CLSR2 DS øH  
CLC LSRNUM,=F'2' LSR Pool count  
BNE CLSR3  
AP TOTLSR2,=P'1'  
B CNTOPEN  

CLSR3 DS øH  
CLC LSRNUM,=F'3' LSR Pool count  
BNE CLSR5  
AP TOTLSR3,=P'1'  
B CNTOPEN  

CLSR5 DS øH  
CLC LSRNUM,=F'5' LSR Pool count  
BNE CLSR6  
AP TOTLSR5,=P'1'  
B CNTOPEN  

CLSR6 DS øH  
CLC LSRNUM,=F'6' LSR Pool count  
BNE CLSR7  
AP TOTLSR6,=P'1'  
B CNTOPEN  

CLSR7 DS øH  
CLC LSRNUM,=F'7' LSR Pool count  
BNE CNTOPEN Shouldn't happen  
AP TOTLSR7,=P'1'  
CNTOPEN DS øH
AP   HLOCNTR,=P'1'
CP   FILECNT,=P'14'
BE   SETFWD
AP   FILPOS,=P'1'
CP   FILPOS,HLQPOS
BNH   FILLOOP
MVI  Ø(R8),X'4Ø'
MVC  1(8,R8),WDDN
CLI   MODEFLG,C'R'
BE   UPCNT
MVI  9(R8),X'4Ø'
MVC  Õ(44,R8),WDSN
MVC  54(8,R8),=C' Closed '
MVC  62(8,R8),=C'
CLC   OPENSTAT,DFHVALUE(OPEN)
BNE   UPCNT
MVC  54(8,R8),=C' Open '
L   R7,INQKLEN
CVD   R7,DWORK
UNPK  62(3,R8),DWORK+5(3)
OC   64(1,R8),=X'FØ'
CLC   62(3,R8),=C'ØØØ'
BNE   NAKEY
MVC  62(3,R8),=C'N/A'
NAKEY  DS   ØH
L   R7,INQRLEN
CVD   R7,DWORK
UNPK  66(4,R8),DWORK+4(4)
OC   69(1,R8),=X'FØ'
UPCNT  DS   ØH
LA   R8,7Ø(R8)
AP   FILECNT,=P'1'
B   FILLOOP
HLQCNT1  DS   ØH
LA   R7,HLQTAB
HLQLOOP  DS   ØH
CLC   WDSN(3),Ø(R7)
BE   HLQMATCH
CLI   Ø(R7),X'4Ø'
BNE   NOTFREE
MVC   Ø(3,R7),WDSN
B   HLQMATCH
NOTFREE  DS   ØH
CLI   Ø(R7),X'ØØ'
BE   HLQMATCH
LA   R7,6(R7)
B   HLQLOOP
HLQMATCH  DS   ØH
AP   3(3,R7),=P'1'
B   FILLOOP
FILEND  DS   ØH
EXEC CICS INQUIRE FILE END
CLI TSTAT,X'ØØ' Summary Level ?
BE TOPMODE Yes go display
CLI TSTAT,X'2Ø' Detail Display ?
BE TOPMODE Yes go display
CLI MODEFLG,C'R' Remote scan ?
BNE NORMSCN2 No check lsr?

LA R7,RSYSTAB
AH R7,OFFSET Get rsysid
MVC RSYSSEL,Ø(R7) And Store it
CLI RSYSSEL,X'4Ø' Empty area ?
BE BEGIN1 Ignore
B NORMSCN1 Carry on

NORMSCN2 DS ØH
CLI MODEFLG,C'L' LSR scan ?
BNE NORMSCN No normal
XC RSYSSEL,RSYSSEL Clear RSYSSEL
MVC RSYSSEL+2(2),OFFSET And Store offset
B NORMSCN1 Carry on

NORMSCN DS ØH
LA R7,HLOQTAB Detail scan
AH R7,OFFSET Get selected HLQ
MVC HLOQSEL,Ø(R7) And Store it
CLI HLOQSEL,X'4Ø' Empty area ?
BE BEGIN1 Ignore

NORMSCN1 DS ØH
LA R8,FILETAB To store file info
ZAP FILECNT,=P'Ø' and a counter
ZAP FILPOS,=P'Ø' and a positioner
ZAP HLOQCNTR,=P'Ø' HLQ counter
ZAP HLOQCTO,=P'Ø' HLQ open counter
MVI TSTAT,X'2Ø' Set Flag and rescan
B BEGIN2

TOPMODE DS ØH
EXEC CICS ASSIGN APPLID(APPL)
MVC REGIO,APPL Move Applid to Map
MVC PARTIO,=C'DEVL'
CLC APPL+3(1),=C'C' PROD is C
BNE GTIME

GTIME DS ØH
EXEC CICS ASKTIME ABSTIME(ATIME)
EXEC CICS FORMATTIME ABSTIME(ATIME) DDMMYYYY(DATEO) *
TIME(TIMEO) DATESEP TIMESEP
MVC OUTCTR,EDPAT
ED OUTCTR,TOTREMT
MVC RTOTOUTO,OUTCTR+1 Remote Total
MVC OUTCTR,EDPAT
ED OUTCTR,TOTLOCL
MVC LTOTOUTO,OUTCTR+1 Local Total
MVC OUTCTR,EDPAT
ED   OUTCTR,TOTOPEN
MVC  OTOTOUTO,OUTCTR+1  Open Total
LA   R7,LSRHEDO
LA   R8,LSRDETO
MVC  OUTCTR,EDPAT
ED   OUTCTR,TOTLSR1
MVC  Ø(5,R7),=C'LSR1 '
MVC  Ø(4,R8),OUTCTR+2  LSR Totals
LA   R7,5(R7)
LA   R8,5(R8)
TRYLSR2  DS  ØH
MVC  OUTCTR,EDPAT
ED   OUTCTR,TOTLSR2
MVC  Ø(5,R7),=C'LSR2 '
MVC  Ø(4,R8),OUTCTR+2  LSR Totals
LA   R7,5(R7)
LA   R8,5(R8)
TRYLSR3  DS  ØH
MVC  OUTCTR,EDPAT
ED   OUTCTR,TOTLSR3
MVC  Ø(5,R7),=C'LSR3 '
MVC  Ø(4,R8),OUTCTR+2  LSR Totals
LA   R7,5(R7)
LA   R8,5(R8)
TRYLSR4  DS  ØH
MVC  OUTCTR,EDPAT
ED   OUTCTR,TOTLSR4
MVC  Ø(5,R7),=C'LSR4 '
MVC  Ø(4,R8),OUTCTR+2  LSR Totals
LA   R7,5(R7)
LA   R8,5(R8)
TRYLSR5  DS  ØH
MVC  OUTCTR,EDPAT
ED   OUTCTR,TOTLSR5
MVC  Ø(5,R7),=C'LSR5 '
MVC  Ø(4,R8),OUTCTR+2  LSR Totals
LA   R7,5(R7)
LA   R8,5(R8)
TRYLSR6  DS  ØH
MVC  OUTCTR,EDPAT
ED   OUTCTR,TOTLSR6
MVC  Ø(5,R7),=C'LSR6 '
MVC  Ø(4,R8),OUTCTR+2  LSR Totals
LA   R7,5(R7)
LA   R8,5(R8)
TRYLSR7  DS  ØH
MVC  OUTCTR,EDPAT
ED   OUTCTR,TOTLSR7
MVC  Ø(5,R7),=C'LSR7 '
MVC  Ø(4,R8),OUTCTR+2  LSR Totals
LA   R7,5(R7)
LA R8,5(R8)

TRYLSR
DS ØH
MVC OUTCTR,EDPAT
ED OUTCTR,TOTLSR
MVC Ø(5,R7),=C'LSR '
MVC Ø(4,R8),OUTCTR+2 LSR Totals
LA R7,5(R7)
LA R8,5(R8)

TRYNSR
DS ØH
MVC OUTCTR,EDPAT
ED OUTCTR,TOTNSR
MVC Ø(5,R7),=C' NSR '
MVC Ø(4,R8),OUTCTR+2 NSR Totals
ENDNSR
DS ØH
LA R7,RSYSTAB
LA R8,RDET1O
LA R9,RDET2O

SYOLOOP
DS ØH
CLI Ø(R7),X'4Ø' Last entry ?
BE HLQOUT
MVC Ø(10,R8),=C'Rsys Total'
MVC Ø(4,R9),Ø(R7)
MVC OUTCTR,EDPAT
ED OUTCTR,4(R7)
MVC 5(5,R9),OUTCTR+1 Move total
CLI Ø(R7),X'ØØ' Bucket entry ?
BE SYSBUCK
LA R7,7(R7) Get next entry
LA R8,11(R8)
LA R9,11(R9)
B SYOLOOP Any more ?

SYSBUCK
DS ØH
MVC Ø(10,R8),=C' Other' Move heading

HLQOUT
DS ØH
CLI TSTAT,X'2Ø' Detail Display ?
BE HLQDET Yes go display
MVC HOUTO,HM1 Move footer
LA R7,HLQTAB
ZAP COLCNT,=P'1' Count HLQ
LA R8,LHED1O
LA R9,LDET1O

HLOOPO
DS ØH
CLI Ø(R7),X'4Ø' Last entry ?
BE SNDMAPE
MVC Ø(9,R8),=C'HLQ Total' Move heading
MVC Ø(3,R9),Ø(R7)
MVC OUTCTR,EDPAT
ED OUTCTR,3(R7)
MVC 4(5,R9),OUTCTR+1 Move total
CLI Ø(R7),X'ØØ' Bucket entry ?
BE HLQBUCK
LA R7,6(R7) Get next entry
LA R8,10(R8) Up display
LA R9,10(R9) pointers
AP COLCNT,=P'1' Next Column
CP COLCNT,=P'8' Seven per row
BL HLLOOP
LA R8,92(R8) Up display
LA R9,92(R9) pointers
ZAP COLCNT,=P'1' Next Line
B HLLOOP Any more ?

Editor's note: this article will be concluded in the next issue.

M J Masters (UK) © Xephon 2000

Sybase has begun shipping Version 12.0 of its MainframeConnect product, geared to moving and accessing information from mainframe sources and LAN datastores.

The new version supports access to both DRDA/MVS and international character sets and enables access to foreign datastores. Support for SQL Server has been added as a source for data replication, as well as enhanced support for access to DB2, Informix, and Oracle data stores.

MainframeConnect provides connectivity between client/server databases and mainframe data, as well as access to DB2/MVS data and on-line production applications in CICS, IMS/TM, and MVS mainframe environments. Production applications in CICS, IMS/TM and MVS environments can also act as clients to LAN-based data and applications.

For further information contact:
Sybase, 6475 Christie Ave, Emeryville, CA 94608, USA.
Tel: (510) 922 3500.

* * *

NEON Systems has unveiled a beta release of its Shadow AutoHTML for CICS/TS, promising direct access from Web and ODBC environments to CICS transactions, promising to Web-enable CICS transactions in minutes, without, it’s claimed, any programming. It includes a High Performance Option, providing pre-built data to deliver large volumes of transactions per second.

It has diagnostics and monitoring capabilities for monitoring the volume of data in the pipeline as well as the number of data transactions queued and waiting to be processed.

A CICS failover feature re-routes users to an available server when the initial server is not accessible. The mapping component allows users to make global changes.

For further information contact:
New Era Of Networks, 7400 E Orchard Rd, Suite 230, Englewood, CO 80111, USA.
Tel: (303) 694 3933.

* * *

WRQ has announced its new suite of integration tools. It includes the Universal Integration Engine, which insulates developers and integrators from interoperability requirements among applications. Also included are visual tools for implementation connectivity options including 25 standard and custom databases and connectors including CICS, XML, CORBA/IIOP, COM/DCOM, Microsoft Repository, MQSeries and Java. It also provides non-SAP information to SAP clients.

New in Apptrieve 3.5, a host integration tool for mining and reusing data in legacy applications on IBM, Unix, and OpenVMS hosts, is multi-processor support, plus new support for HP-UX as well as Solaris.

For further information contact:
WRQ, 1500 Dexter Avenue North, Seattle, WA 98109, USA.
Tel: (206) 217 7500.