In this issue

3  Stopping people bringing up CICS regions and leaving them
9  A simple interface to DFHCSDUP to manage a CICS CSD resource
26 Dynamic Transaction Routing
39 Interfacing CICS to SMTP – part 1
48 CICS news
Editor
Trevor Eddolls

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

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

© Xephon plc 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.
Stopping people bringing up CICS regions and leaving them

Here’s a little program I use to stop people bringing up CICS regions and leaving them there all day without using them. You add it to the PLT so that it runs at start-up, then restarts itself every hour (you can set the interval yourself). It then checks the EIBTASKN, compares it to what it was an hour ago, and, if the minimum number of tasks have not been run, it shuts down the region. I wrote it for CICS 4.1 but it seems to work for CICS TS 1.3 as well.

*ASM XOPTS(PROLOG,NOEPLOG,SP)
  TITLE 'BWATCHDG - TECHNICAL SERVICES TRXN ACTIVITY WATCH DOG'

**********************************************************************
* BWATCHDG:                                                          *
*       - TECHNICAL SERVICES CICS PLTP (PHASE 2) PROGRAM             *
*       - INVOKED DURING THE THIRD STAGE OF CICS INITIALIZATION.     *
*       - THIS PROGRAM THEN RE-INVOKES ITSELF AT 60 MINUTE INTERVALS *
*       - HIGH LEVEL ASSEMBLER                                       *
*       - COMMAND LEVEL CICS/ESA V4.1.Ø                             *
*       - CICS TRANSLATOR OPTION "SP" IS REQUIRED                    *
*       - EXECKEY(CICS)                                             *
*       - TASKDATAKEY(CICS)                                         *
*       - 31 BIT ADDRESSING                                         *
***********************************************************************

TITLE 'BWATCHDG - REGISTER EQUATES'
R0   EQU   0
R1   EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15

TITLE 'BWATCHDG - CICS PROLOG CODE / ADDRESSING MODE'

BWATCHDG CSECT
BWATCHDG AMODE 31
BWATCHDG RMODE ANY

TITLE 'BWATCHDG - MAINLINE CODE'
**********************************************************************
* BRANCH PAST PROGRAM NAME "EYECATCHER".                          *
**********************************************************************
MAIN     DS    ØH
B     MAIN1ØØ
DC    CL2'<<'
DC    CL1'
DC    CL39'BRISTOL AND WEST BUILDING SOCIETY (TSG)'
DC    CL1'
DC    CL8'&SYSDATE'
DC    CL1'
DC    CL5'&SYSTIME'
DC    CL1'
DC    CL45'BWATCHDG - WATCH DOG FOR TRANSACTION ACTIVITY'
DC    CL1'
DC    CL2'>>'
**********************************************************************
* MAINLINE CODE                                                      *
**********************************************************************
MAIN1ØØ  DS    ØH
*                                      GET TASK NUMBER SAVED BY THE
*                                      PREVIOUS INVOCATION OF THIS
*                                      PROGRAM.
*  EXEC CICS ASSIGN APPLID(APPLID)
   LA    R5,APP_TAB
CHK_APP DS    ØH
   CLC   Ø(8,R5),APPLID
   BE    SKIPPROC
   CLC   Ø(8,R5),=C'ZZZZZZZZZ'
   BE    CONTINUE
LA R5,8(R5)
B CHK_APP
CONTINUE DS ØH

WTO 'BWATCHDG - AUTO SHUTDOWN IS ACTIVE IN THIS REGION'
EXEC CICS ASKTIME ABSTIME(ABSTIME)
EXEC CICS FORMATTIME ABSTIME(ABSTIME)
   DATE(DISP_DAT) DATESEP('/')
   TIME(DISP_TIM) TIMESEP(':')
MVC DISP_TXT,MSG1
EXEC CICS WRITEQ TD
   QUEUE('CSML')
   FROM(DISP_MSG)
   LENGTH(L'DISP_MSG)
*
EXEC CICS CANCEL TRANSID(TRNID)
   REQID(APPLID)
   RESP(RESP)
*
EXEC CICS INQUIRE SYSTEM
   CICSSTATUS(CICSSTAT)
   RESP(RESP)
CLC CICSSTAT,DFHVALUE(STARTUP)
BE MAIN8ØØ
*
EXEC CICS READQ TS
   QUEUE('WATCHDOG')
   ITEM(1)
   INTO(TSQ_REC)
   RESP(RESP).
*
CLC RESP,DFHRESP(NORMAL)
BNE MAIN8ØØ
*
UNPK TASK_#(4),TSQ_REC
MVZ TASK_#+3(1),=X'FØ'
MVC MSG4+22(4),TASK_#
EXEC CICS ASKTIME ABSTIME(ABSTIME)
EXEC CICS FORMATTIME ABSTIME(ABSTIME)
   DATE(DISP_DAT) DATESEP('/')
   TIME(DISP_TIM) TIMESEP(':')
MVC DISP_TXT,MSG4
EXEC CICS WRITEQ TD
   QUEUE('CSML')
   FROM(DISP_MSG)
   LENGTH(L'DISP_MSG')
*
ZAP CURR_#,EIBTASKN
ZAP PREV_#,TSQ_REC
ZAP COMP_#,CURR_#
SP COMP_#,PREV_#
CP COMP_=P'3'
BNH SHUTDOWN
MAIN2ØØ ZAP EIBT#,EIBTASKN
* EXEC CICS DELETEQ TS
  QUEUE('WATCHDOG')
  RESP(RESP).
*
B MAIN82Ø
*
MAIN8ØØ DS ØH
* EXEC CICS DELETEQ TS
  QUEUE('WATCHDOG')
  RESP(RESP).
* ZAP EIBT#,=P'1'
MAIN82Ø DS ØH
  UNPK TASK_#(4),EIBT_
  MVZ TASK_#+3(1),=X'F0'
  MVC MSG5+22(4),TASK_
  EXEC CICS ASKTIME ABSTIME(ABSTIME)
  EXEC CICS FORMATTIME ABSTIME(ABSTIME)
    DATE(DISP_DAT) DATESEP('/')
    TIME(DISP_TIM) TIMESEP(':')
  MVC DISP_TXT,MSG5
  EXEC CICS WRITEQ TD
    QUEUE('CSML')
    FROM(DISP_MSG)
    LENGTH(L'DISP_MSG')
* EXEC CICS WRITEQ TS
  QUEUE('WATCHDOG')
  FROM(EIBT_#)
  LENGTH(L'EIBT_#)
  RESP(RESP).
* CLC RESP,DFHRESP(NORMAL)
  BNE RETURN
*
MAIN9ØØ DS ØH
* CLC APPLID,=C'TESTCIC2'
  BE SHORT
* EXEC CICS START
  AFTER MINUTES(60)
  TRANSID(TRNID)
  REQID(APPLID)
  RESP(RESP).
EXEC CICS START
    AFTER MINUTES(60)
    TRANSID(TRNID)
    REQID(APPLID)
    RESP_RESP.
B RETURN
* SKIPPROC DS ØH
    WTO 'BWATCHDG - AUTO SHUTDOWN NOT ACTIVE IN THIS REGION'
* RETURN DS ØH
* EXEC CICS ASKTIME ABSTIME(ABSTIME)
EXEC CICS FORMATTIME ABSTIME(ABSTIME)
    DATE(DISP_DAT) DATESEP('/')
    TIME(DISP_TIM) TIMESEP(':')
MVC DISP_TXT,MSG2
EXEC CICS WRITEQ TD
    QUEUE('CSML')
    FROM(DISP_MSG)
    LENGTH(L'DISP_MSG')
* EXEC CICS RETURN
* SHUTDOWN DS ØH
* ZAP EIBT_#,EIBTASKN
UNPK TASK_#(4),EIBT_
MVZ TASK_#+3(1),=X'F0'
MVC MSG5+22(4),TASK_
EXEC CICS ASKTIME ABSTIME(ABSTIME)
EXEC CICS FORMATTIME ABSTIME(ABSTIME)
    DATE(DISP_DAT) DATESEP('/')
    TIME(DISP_TIM) TIMESEP(':')
MVC DISP_TXT,MSG5
EXEC CICS WRITEQ TD
    QUEUE('CSML')
    FROM(DISP_MSG)
    LENGTH(L'DISP_MSG')
MVC MSG3+11(8),APPLID
EXEC CICS ASKTIME ABSTIME(ABSTIME)
EXEC CICS FORMATTIME ABSTIME(ABSTIME)
    DATE(DISP_DAT) DATESEP('/')
    TIME(DISP_TIM) TIMESEP(':')
MVC DISP_TXT,MSG3
EXEC CICS WRITEQ TD
    QUEUE('CSML')
FROM(DISP_MSG) X
LENGTH(L'_DISP_MSG)

*
*
---

MVC COMMAREA(L'Request).REQUEST
*
EXEC CICS XCTL PROGRAM('DFHEMTP') X
COMMAREA(REQUEST) LENGTH(MAX_LEN)
*
EXEC CICS RETURN
*
*
B MAIN2ØØ
*
TITLE 'BWATCHDG - LITERAL POOL'

LTORG

MSG1 DC CL8Ø'BWATCHØ1 - PROGRAM START'
MSG2 DC CL8Ø'BWATCHØ2 - PROGRAM END - CICS CONTINUES'
MSG3 DC CL8Ø'BWATCHØ3 - XXXXXXXX CICS NOT USED - SHUTDOWN'
MSG4 DC CL8Ø'BWATCHØ4 - EIBTASKN = XXXX (READ FROM QUEUE)'
MSG5 DC CL8Ø'BWATCHØ5 - EIBTASKN = XXXX (READ FROM EIB)'
REQUEST DC C'CEMT P SHUT'
MAX_LEN DC H'11'
TRNID DC CL4'BWAT'
APP_TAB DC CL8'A1CICTD1'
* DC CL8'A1CICTC6'
* DC CL8'A1CICTCA'
* DC CL8'A1CICTK2'
DC CL8'ZZZZZZZZ'
TITLE 'BWATCHDG - DFHEISTG DSECT'

DFHEISTG DSECT

DISP_MSG DS ØCL1ØØ
DISP_DAT DS CL8
DISP_FIL DS CL1
DISP_TIM DS CL8
DISP_FI1 DS CL1
DISP_TXT DS CL8Ø
ABSTIME DS PL8
TSQ_REC DS PL4
CURR_# DS PL4
PREV_# DS PL4
COMP_# DS PL4
TASK_# DS CL4
EIBT_# DS PL4
EIB_LEN DS H
EIBT_D DS D
EIBT_F DS F
RESP DS F
RESP2 DS F
CICSSTAT DS F
A simple interface to DFHCSDUP to manage a CICS CSD resource

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

There is a CSD back-up utility program to produce a file of DFHCSDUP DEFINE control statements. The program is DFH0CBDC and it is written in VS COBOL II. It produces a sequential file containing all the resource definitions in a CSD file.

You can use the output from the utility:

- For later editing and commenting to document CSD resources.
- For distribution, in part or as a whole, to other CICS installations.
- To recreate or add resource definitions to any CSD using DFHCSDUP.
- To build or rebuild CSD definitions.
- As a sample back-up copy of the CSD VSAM file.

The program DFH0CBDC must be run against an EXTRACT command using the following formats:

```
EXTRACT GROUP(group name) OBJECTS USERPROGRAM(DFH0CBDC)
```

or:

```
EXTRACT LIST(list name) OBJECTS USERPROGRAM(DFH0CBDC)
```

In the file produced by DFH0CBDC, any DEFINE statements that
relate to CICS-supplied resources are commented out (there is an asterisk in column 1).

This is important if you use the file as input to define resources to a CSD because the CICS-supplied definitions are already present in the CSD, having been produced automatically when it was initialized.

The DEFINE statements for user CICS resources aren’t preceded by an asterisk in column 1.

When an extract is carried out for a CICS list, we obtain a sequential file that contains all the definitions of the CICS list. This can make the management of single groups more complex.

For a simpler way to manage single groups, once the execution of the DFH0CBDC user program to extract the CICS list is ended, we can process the output sequential file by subdividing the CICS list into groups.

In order to provide this functionality, I have developed the CICSCSD CLIST. The CLIST subdivides the CICS list into CICS groups. It has been developed using the REXX language.

It reads the sequential file produced from the DFH0CBDC user program and produces a partitioned file.

With the execution of CICSCSD CLIST, we obtain, therefore, a partitioned file that contains a member for every group in the CICS list, and a member (NEWLIST) with the control statements to define a new CICS list with DFHCSDUP.

This makes the management of individual CICS groups much simpler. It makes it easier to use the CICS resource definitions in the CSD file.

This utility was developed and tested in the following environment: OS/390 1.3 and OS/390 2.7, CICS/ESA 4.1.0, and CICS TS 1.3.

**COMPILE AND LINK-EDITING THE EXTRACT USER PROGRAM**

You must compile and link-edit the DFH0CBDC user programs as batch programs, not as CICS applications.
When you link-edit the programs, you must specify the following link-edit control statements:

- An ENTRY statement that defines the entry name as DFHEXTRA. DFHEXTRA is the entry name in the CICS supplied stub, DFHEXCI.

- An INCLUDE statement for a CICS-supplied stub that must be included in your user program. Include DFHEXCI in any COBOL language user program that you write for use with the DFHCSDUP EXTRACT command. DFHEXCI is the interface stub between DFHCSDUP and the COBOL user program.

- Specify the COBOL routines on the INCLUDE statements.

- A CHANGE statement to change the dummy CSECT name in the CICS-supplied stub from EXITEP to the name of your user program. The CICS supplied stub, DFHEXCI, is generated with a link to the user program using a dummy CSECT name (EXITEP). Use the link-edit CHANGE statement to change the CSECT name from EXITEP to the name specified on the PROGRAM-ID statement in the user program. The CSD user program DFH0CBDC uses the program-id BDEFCSD.

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

SAMPLE JOB TO COMPILE AND LINK-EDIT THE USER PROGRAM

```plaintext
//ZZNS611M JOB (ZZNSØ0000),
//   CLASS=S,
//   MSGCLASS=X,
//   MSGLEVEL=(1,1),
//   NOTIFY=&SYSUID
//*
//**---------------------------------------------------------------*/
//**--------   Step to compile COBOL II user program     --------*/
//**---------------------------------------------------------------*/
//*
//COB2    EXEC PGM=IGYCRCTL,
//   PARM='APOST,LIB,OFFSET,MAP,NOSEQ,BUFSIZE(30K),NORENT,NORES'
//*
//STEPLIB DD DISP=SHR,DSN=COBII.COB2COMP
//SYSLIB DD DISP=SHR,DSN=ZNS611.LIB.SOURCE(DFH0CBDC)
//SYSLIB DD DISP=SHR,DSN=ZNS611.LIB.SOURCE
```
PROGRAM DFH0CBDC

******************************************************************
* MODULE NAME = DFH0CBDC  *
* Sample program to produce a CSD listing from EXTRACT          *
******************************************************************

ENTRY    DFH0EXTRA
CHANGE   EXITEP(BDEFCSD)
INCLUDE  CICSLIB(DFH0EXCI)
INCLUDE  SYSLIB(ILBOSRV)
INCLUDE  SYSLIB(ILBOCMM)
INCLUDE  SYSLIB(ILBOBEG)
INCLUDE  OBJLIB(DFH0CBDC)
NAME     DFH0CBDC(R)
/*
/ *
* Descriptions.

* * *

*----------------------------------------------------------------*  

* LANGUAGE: COBOL II

* * *

* BDEFCSD takes input from the batch EXTRACT command and
* produces a listing of part or all of the CSD that can be used
* to build or rebuild CSD definitions, or just as a summary
* listing of the CSD. The listing can be used as a means of
* passing CSD definitions from one CSD to another in a readable,
* commentable form.
* The output varies according to the EXTRACT command being
* processed as follows:
* * *

* EXTRACT GROUP( .. ):  
* For every resource in the specified group, a partial DEFINE
* command is generated, eg:
*    DEFINE resource(XXXXXXX) GROUP(YYYYYYYY)
* Note that this command is not acceptable input to the batch
* utility. However, it does provide an editable list of the
* resources in each group.
* * *

* EXTRACT GROUP( .. ) OBJECTS:
* For every resource in the specified group, a DEFINE command is
* generated, eg:
*    DEFINE PROGRAM(XXXXXXXX) GROUP(YYYYYYYY)
*    DESCRIPTION( ... )
*    LANGUAGE(COBOL) RELOAD(NO) RESIDENT( ...  
* * This is acceptable input to the batch utility.
* * *

* EXTRACT LIST:
* For every occurrence of a group in the specified list an ADD
* command is generated, eg:
*    ADD GROUP(XXXXXXXX) LIST(YYYYYYYY)
* * *

* EXTRACT LIST OBJECTS:
* As above, plus the equivalent of EXTRACT GROUP( .. ) OBJECTS
* for every group in the specified list.
* * *

* Notes:
* ADD and DEFINE commands and DESCRIPTION keywords begin
* in column 1. All other keywords are listed starting in
* column 8. If commented out, these values are increased
* by 1.
* * *

* An ADD command featuring a list beginning 'DFH' is
* prefixed by a DFHCSDUP comment character, '*'.
* *
A DEFINE command featuring a group beginning 'DFH' is prefixed by a DFHCSDUP comment character, '*'.

Any resource keyword containing a null value is not output because it is not valid input to DEFINE.

Any resource keyword and its value (other than DESCRIPTION) that will not fit into columns 8-71 is output over multiple lines in a format acceptable as input to DFHCSDUP.

Inputs: Data from DFHCULIS
Outputs: Data file (see DATA-OUT) or the following return code:
Ø2: Invalid call type supplied to program

OUTPUT FILE
DEFINE and ADD commands will be written to this file.
Note that NO error processing is performed for this file, eg open/close errors.

IDENTIFICATION DIVISION.
PROGRAM-ID. BDEFCSO.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT DATA-OUT
  ASSIGN TO SYS002-DA-33B0-S-CBDOUT
  ORGANIZATION IS SEQUENTIAL
  ACCESS IS SEQUENTIAL.
EJECT.
DATA DIVISION.
FILE SECTION.
FD DATA-OUT
  RECORD CONTAINS 80 CHARACTERS
  BLOCK CONTAINS 0 RECORDS
  RECORDING MODE IS F
  LABEL RECORDS ARE OMITTED.
Ø1 WRITE-BUFFER PIC X(80).
EJECT.
WORKING-STORAGE SECTION.
* Call types, as defined by DFHCSDUP.
77 INITIAL-CALL          PIC 99 VALUE IS Ø.
77 LIST-START-CALL      PIC 99 VALUE IS 2.
77 GROUP-START-CALL     PIC 99 VALUE IS 4.
77 OBJECT-START-CALL    PIC 99 VALUE IS 6.
77 KEYWORD-CALL         PIC 99 VALUE IS 8.
77 OBJECT-END-CALL       PIC 99 VALUE IS 10.
77 GROUP-END-CALL        PIC 99 VALUE IS 12.
77 LIST-END-CALL         PIC 99 VALUE IS 14.
77 FINAL-CALL           PIC 99 VALUE IS 16.

* Current list and group being processed.
*
Ø1 CURRENT-LIST-ID.
    Ø2 LIST-PREFIX       PIC X(3).
    Ø2 LIST-SUFFIX       PIC X(5).
Ø1 CURRENT-GROUP-ID.
    Ø2 GROUP-PREFIX      PIC X(3).
    Ø2 GROUP-SUFFIX      PIC X(5).

* LONG KEYWORD VALUE HANDLING
*
Ø1 LINE-LENGTH           PIC S9(4) COMP.
Ø1 LENGTH-LINE-1         PIC S9(4) VALUE IS 49.
Ø1 CHAR-COUNT            PIC S9(4) COMP.
Ø1 LAST-CHAR             PIC X.
Ø1 TEMP-BUFFER           PIC X(71).
Ø1 TEMP-BUFFER-PTR       PIC S9(4) COMP.
Ø1 CREATE-LINE-END       PIC X(1).

Ø1 INDENTED-LINE-LEN     PIC S9(4) VALUE IS 62.

77 ASTERISK              PIC X VALUE IS '*'.
77 FLUSH-PREFIX           PIC X(3).
* Return code, if an invalid call is made to this program
77 INVALID-CALL-TYPE     PIC S9(4) COMP VALUE IS 2.

* For buffering data whilst it is being formatted.
*
77 BUFFER                PIC X(73).
77 BUFFER-PTR            PIC S9(4) COMP.
77 NUMBER-OF-SPACES      PIC S9(4) COMP.
77 KEYWORD-TYPE-LENGTH   PIC S9(4) COMP.
77 KEYWORD-LENGTH        PIC S9(4) COMP.
77 BUFFER-SPACE          PIC S9(4) COMP.
77 MAX-TYPE-LENGTH       PIC S9(4) COMP VALUE IS 12.
77 BUFFER-INDENT         PIC S9(4) COMP VALUE IS 8.
77 PARENTHESIS-LENGTH    PIC S9(4) COMP VALUE IS 2.
77 KEYWORD-SCAN-PTR      PIC S9(4) COMP.
77 CHARACTER-COUNT       PIC S9(4) COMP.
77 CURRENT-LINE-LENGTH   PIC S9(4) COMP.
* Define the linkage between this program and the CSD off-line *
* utility program. The addressability to the values addressed by *
* the parameter list passed from DFHCS DUP is established      *
* automatically by the COBOL compiler so all we need to define *
* here are the actual formats of the values themselves.         *

LINKAGE SECTION.
Ø1 EXIT-FUNCTION-CODE             PIC 99 COMP.
Ø1 EXIT-WORK-AREA-PTR             POINTER.
Ø1 EXIT-BACK-TRANS-CMD-PTR        POINTER.
Ø1 EXIT-LIST-NAME                 PIC X(8).
Ø1 EXIT-GROUP-NAME                PIC X(8).
Ø1 EXIT-OBJECT-TYPE               PIC X(12).
Ø1 EXIT-OBJECT-NAME               PIC X(8).
Ø1 EXIT-KEYWORD-TYPE              PIC X(12).
Ø1 EXIT-KEYWORD-LENGTH            PIC 999 COMP.
Ø1 EXIT-KEYWORD-VALUE.
Ø3 EXIT-KEYWORD-CHAR           PIC X OCCURS 1 TO 183
                                    DEPENDING ON EXIT-KEYWORD-LENGTH.
EJECT.

PROCEDURE DIVISION USING EXIT-FUNCTION-CODE
    EXIT-WORK-AREA-PTR
    EXIT-BACK-TRANS-CMD-PTR
    EXIT-LIST-NAME
    EXIT-GROUP-NAME
    EXIT-OBJECT-TYPE
    EXIT-OBJECT-NAME
    EXIT-KEYWORD-TYPE
    EXIT-KEYWORD-LENGTH
    EXIT-KEYWORD-VALUE.

* Perform appropriate action, according to EXIT-FUNCTION-CODE *

    EVALUATE EXIT-FUNCTION-CODE
    * Open output file and initialize BUFFER to SPACES.
    WHEN INITIAL-CALL PERFORM INITIALISE
* Store current list name.
    WHEN LIST-START-CALL
        MOVE EXIT-LIST-NAME TO CURRENT-LIST-ID
* Store current group name and output an ADD command if there
* is a current list.
    WHEN GROUP-START-CALL
        PERFORM
MOVE EXIT-GROUP-NAME TO CURRENT-GROUP-ID
IF CURRENT-LIST-ID NOT EQUAL SPACES
  THEN PERFORM
    PERFORM ADD-GROUP-TO-LIST
    MOVE LIST-PREFIX TO FLUSH-PREFIX
    PERFORM FLUSH-BUFFER
    END-PERFORM
  END-IF
END-PERFORM

* Output a partial define command.
WHEN OBJECT-START-CALL
  PERFORM
    PERFORM PROCESS-OBJECT
    MOVE GROUP-PREFIX TO FLUSH-PREFIX
    PERFORM FLUSH-BUFFER
    END-PERFORM
  WHEN KEYWORD-CALL
  PERFORM
    IF EXIT-KEYWORD-LENGTH NOT EQUAL ZERO
      THEN PERFORM
        IF EXIT-KEYWORD-TYPE = 'DESCRIPTION'
          THEN PERFORM
            IF BUFFER-PTR > BUFFER-INDENT
              THEN PERFORM FLUSH-BUFFER
            END-IF
            PERFORM PROCESS-KEYWORD
            PERFORM FLUSH-BUFFER
          END-IF
        ELSE PERFORM
          IF BUFFER-PTR = 1
            THEN MOVE BUFFER-INDENT TO BUFFER-PTR
          END-IF
          PERFORM PROCESS-KEYWORD
        END-IF
      END-IF
    END-PERFORM
  END-IF
END-PERFORM

* Flush the buffer at the end of a resource.
WHEN OBJECT-END-CALL PERFORM FLUSH-BUFFER

* Clear current group name at the end of the group.
WHEN GROUP-END-CALL MOVE SPACES TO CURRENT-GROUP-ID

* Clear current list name at the end of the list.
WHEN LIST-END-CALL MOVE SPACES TO CURRENT-LIST-ID

* Close output file.
WHEN FINAL-CALL PERFORM CLEAN-UP
* Signal invalid call type to DFHCULIS.
  WHEN OTHER MOVE INVALID-CALL-TYPE TO RETURN-CODE
  END-EVALUATE
*
* Must use GOBACK so that COBOL data areas are preserved and usable
* on the next invocation of the program (eg CURRENT-LIST-ID and
* CURRENT-GROUP-ID).
*
  IF EXIT-FUNCTION-CODE = FINAL-CALL THEN
    STOP RUN
  ELSE
    GOBACK.
*
ADD-GROUP-TO-LIST.
*
* Place command on output file of the form:
*          ADD GROUP( ... ) LIST( ... )
*
* If there is no current list then this routine should not be
* called.
*
  STRING 'ADD GROUP(' DELIMITED BY SIZE
  CURRENT-GROUP-ID DELIMITED BY SPACE
  ')' LIST(' DELIMITED BY SIZE
  CURRENT-LIST-ID DELIMITED BY SPACE
  ')' DELIMITED BY SIZE
  INTO BUFFER
  END-STRING.
*
ADD-KEYWORD-TO-BUFFER.
*
* Routine to write a keyword to the output buffer.
* Note that parentheses are placed around the keyword value and
* a space is appended to the string for padding.
*
  STRING EXIT-KEYWORD-TYPE DELIMITED BY SPACE
    '(' EXIT-KEYWORD-VALUE ') ' DELIMITED BY SIZE
    INTO BUFFER
    WITH POINTER BUFFER-PTR
  END-STRING.
*
*
SPLIT-KEYWORD.
*
* This procedure splits a keyword value(s) over
* multiple lines. As much of the value as possible is put
* out on the first line, and a continuation character (*)
* placed in column 72. Subsequent lines contain 71
* Keyword characters plus a continuation character, apart
* from the final line which contains the remainder of the
* Keyword value and a right parenthesis.
*
* Put keyword name at the start of a new line
  STRING EXIT-KEYWORD-TYPE DELIMITED BY SPACE
       '(' DELIMITED BY SIZE
       INTO BUFFER
       WITH POINTER BUFFER-PTR
  END-STRING.
*
* Initialize variables required for data formatting.
*
  MOVE SPACES TO TEMP-BUFFER
  MOVE 0 TO KEYWORD-SCAN-PTR
  MOVE 1 TO TEMP-BUFFER-PTR
  MOVE 0 TO CHARACTER-COUNT
*
* Now scan keyword value(s) and fill output buffer. When
* full, flush buffer and continue processing for keyword
* creating output records until all keyword value(s)
* processed.
*
  PERFORM PERFORM-SPLIT
       UNTIL KEYWORD-SCAN-PTR =
           EXIT-KEYWORD-LENGTH.
*
* Write a closing ')' to the output buffer.
*
  STRING TEMP-BUFFER DELIMITED BY SPACE
       ')' ' DELIMITED BY SIZE
       INTO BUFFER
       WITH POINTER BUFFER-PTR
  END-STRING.
*
PERFORM-SPLIT.
*
* This routine sequentially transfers characters from
* EXIT-KEYWORD-VALUE to TEMP-BUFFER. If TEMP-BUFFER is
* filled, then its contents are written to BUFFER,
* a continuation character concatenated in column 72. The
* contents of BUFFER are then written to the output file
* via a call to FLUSH-BUFFER.
*
  ADD 1 TO KEYWORD-SCAN-PTR.
  ADD 1 TO CHARACTER-COUNT.
*
* Calculate line length so far.
*
  ADD CHARACTER-COUNT
  BUFFER-PTR
GIVING CURRENT-LINE-LENGTH.

* Find the next delimited string

* STRING EXIT-KEYWORD-CHAR(KEYWORD-SCAN-PTR)
  DELIMITED BY SIZE
  INTO TEMP-BUFFER
  WITH POINTER TEMP-BUFFER-PTR
END-STRING.

* Check for full output line, ie 71 characters.

* IF CURRENT-LINE-LENGTH > 71 THEN
  STRING TEMP-BUFFER DELIMITED BY SPACE
  ASTERISK DELIMITED BY SIZE
  INTO BUFFER
  WITH POINTER BUFFER-PTR
END-STRING
  PERFORM FLUSH-BUFFER
  MOVE SPACES TO TEMP-BUFFER
  MOVE 1 TO BUFFER-PTR
  MOVE 1 TO TEMP-BUFFER-PTR
  MOVE Ø TO CHARACTER-COUNT
END-IF.

* PROCESS-OBJECT.

* Routine to write an object type to the output file

* STRING 'DEFINE ' DELIMITED BY SIZE
  EXIT-OBJECT-TYPE DELIMITED BY SPACE
  '(' DELIMITED BY SIZE
  EXIT-OBJECT-NAME DELIMITED BY SPACE
  ') GROUP(' DELIMITED BY SIZE
  CURRENT-GROUP-ID DELIMITED BY SPACE
  ') Delimited by SIZE
  INTO BUFFER
  WITH POINTER BUFFER-PTR
END-STRING.

* PROCESS-KEYWORD.

* Routine to process a keyword and its value.
* If the keyword and its value will fit in the remaining buffer
* space then write them. Otherwise flush the buffer and place
* the keyword and its value in an empty buffer. If it still won't
* fit, then issue an error message for that keyword.

* Calculate the length of the keyword and its value (including
* parentheses) and then check to see if it will fit in the buffer.
* If it won't fit, flush the buffer and retry. If it still won't
* then this is an error.
*
    MOVE ZERO TO NUMBER-OF-SPACES
    INSPECT EXIT-KEYWORD-TYPE
        TALLYING NUMBER-OF-SPACES
        FOR ALL SPACE
    SUBTRACT NUMBER-OF-SPACES FROM MAX-TYPE-LENGTH
        GIVING KEYWORD-TYPE-LENGTH
    ADD KEYWORD-TYPE-LENGTH EXIT-KEYWORD-LENGTH
        PARENTHESIS-LENGTH
        GIVING KEYWORD-LENGTH
    SUBTRACT BUFFER-PTR FROM 72 GIVING BUFFER-SPACE
    IF BUFFER-SPACE > KEYWORD-LENGTH - 1
        THEN PERFORM ADD-KEYWORD-TO-BUFFER
    ELSE PERFORM
        IF BUFFER-PTR > BUFFER-INDENT
            THEN
                PERFORM FLUSH-BUFFER
        END-IF
        IF EXIT-KEYWORD-TYPE NOT EQUAL 'DESCRIPTION'
            THEN MOVE BUFFER-INDENT TO BUFFER-PTR
        END-IF
    SUBTRACT BUFFER-PTR FROM 72 GIVING BUFFER-SPACE
    IF BUFFER-SPACE > KEYWORD-LENGTH - 1
        THEN PERFORM ADD-KEYWORD-TO-BUFFER
    ELSE PERFORM SPLIT-KEYWORD
    END-IF
END-PERFORM
END-IF.
*
    FLUSH-BUFFER.
*
    * Routine to write the buffer to the output file.
    * If the group (for DEFINE) or the list name (for ADD) begin
    * 'DFH' then prefix the output with DFHCSUDP comment
    * character, '*' .
    *
        MOVE SPACES TO WRITE-BUFFER
        IF FLUSH-PREFIX = 'DFH'
            THEN STRING ASTERISK BUFFER DELIMITED BY SIZE
                INTO WRITE-BUFFER
            END-STRING
        ELSE MOVE BUFFER TO WRITE-BUFFER
        END-IF
WRITE WRITE-BUFFER.
MOVE SPACES TO BUFFER.
MOVE 1 TO BUFFER-PTR.
*
INITIALISE.
* * Routine called at initial call to set up working environment. *
* OPEN OUTPUT DATA-OUT.
  MOVE SPACES TO BUFFER.
  MOVE 1 TO BUFFER-PTR.
  MOVE SPACES TO CURRENT-LIST-ID.
  MOVE SPACES TO CURRENT-GROUP-ID.
*
CLEAN-UP.
*
* Routine called at final call to clean up
*
  CLOSE DATA-OUT.

CICSCSD EXEC

/* REXX */
/* CICS CSD utility
  C-List CICSCSD
   Called by job batch or TSO user command.
   It executes the subdivision of CICS list in CICS groups.
   The relative output is written into a partitioned file. */

Trace ?o
PARSE ARG cxfili cxfilo
endfin = 'NO'
/****************************/
/* Alloc input file          */
/****************************/
ADDRESS TSO
  dd=OUTTRAP(dd.)
"ALLOC DA('"cxfili"') F(CXFILI) SHR REUSE"
  dd=OUTTRAP('OFF')
  codret = rc
  if codret ~= Ø then do
    typfunc = 'Allocfili'
    Signal CX_Error_func
    say mess
    Return
  End
/****************************/
/* Read input file           */
/****************************/
"NEWSTACK"
  dd=OUTTRAP(dd.)
"EXECIO Ø DISKR CXFILI (OPEN"
"EXECIO 1 DISKR CXFILI"
  dd=OUTTRAP('OFF')
codret = rc
if codret == Ø then do
  typfunc = 'Readfili'
  Signal CX_Error_func
  say mess
  Return
End

Call CX_read_func

/**************************************************************/
/* Alloc output file: Routine to write PDS file.              */
/**************************************************************/
If define = 'ADD GROUP' then
  do
    "ALLOC DA('"cxfilo"(NEWLIST)') F(CXFILOA) SHR REUSE"
    "EXECIO Ø DISKW CXFILOA (OPEN"
    Call CX_write_func
    "EXECIO 1 DISKW CXFILOA "
    "EXECIO Ø DISKW CXFILOA (FINIS"
    "ALLOC DA('"||cxfilo||"("||GRUPPO||")') F(CXFILO) SHR REUSE"
    "EXECIO Ø DISKW CXFILO (OPEN"
    "EXECIO 1 DISKR CXFILI"
    CALL CX_read_func
  END
Do While endfin = 'NO'
  Do While endfin = 'NO' & define <> 'ADD GROUP'
    Call CX_write_func
    "EXECIO 1 DISKW CXFILO "
    "EXECIO 1 DISKR CXFILI"
    Call CX_read_func
  End
  "EXECIO Ø DISKW CXFILO (FINIS"
if define = 'ADD GROUP' then
  do
    "ALLOC DA('"||cxfilo||"(NEWLIST)') F(CXFILOA) SHR"
    "EXECIO Ø DISKR CXFILOA (OPEN"
    "EXECIO * DISKR CXFILOA (STEM NEWLIST."n
    "EXECIO Ø DISKR CXFILOA (FINIS"
    "ALLOC DA('"||cxfilo||"("||GRUPPO||")') F(CXFILO) SHR REUSE"
    "EXECIO Ø DISKW CXFILOA (OPEN"
    "EXECIO * DISKW CXFILOA (STEM NEWLIST."n
    Call CX_write_func
    "EXECIO 1 DISKW CXFILOA"
    "EXECIO Ø DISKW CXFILOA (FINIS"
    "ALLOC DA('"||cxfilo||"("||GRUPPO||")') F(CXFILO) SHR REUSE"
    "EXECIO Ø DISKW CXFILO (OPEN"
    "EXECIO 1 DISKR CXFILI"
    Call CX_read_func
  End
END
"EXECIO Ø DISKR CXFILI (FINIS"
"FREE F(CXFILI CXFILO CXFILOA)"

Exit

CX_Error_func:
FLICTED---------------------------------------------------------*/
/* Errors routine */
FLICTED---------------------------------------------------------*/

Select
when typfunc = 'Allocfili' then
mess = 'Allocation INPUT file in error.'
when typfunc = 'Readfili' then
mess = 'Read INPUT file in error.'
Otherwise nop
End

Return

CX_read_func:
PULL cxreci.1
define = SUBSTR(cxreci.1,1,9)
ngroup = SUBSTR(cxreci.1,11,8)
Parse Value ngroup With GRUPPO '}
if rc = 2 then
   endfin = 'YES'

Return

CX_WRITE_FUNC:
cxreco.1 = cxreci.1
PUSH cxreco.1
Return

SAMPLE JOB TO EXECUTE THE DFH0CBDC PROGRAM AND CICSCSD CLIST

//ZZNS611M JOB (ZZNS0000),
 //   CLASS=S,REGION=0M,
 //   MSGCLASS=X,
 //   MSGLEVEL=(1,1),
 //   NOTIFY=&SYSUID
 //*
//*JOBPARM BYTES=999999,LINES=9999
//*
//**************************
//* DELETE INPUT/OUTPUT FILE FOR CSD UTILITY
//**************************
//DELETE EXEC  PGM=IDCAMS,REGION=1M
//SYSPRINT DD  SYSOUT=*
//SYSSIN DD  *
DELETE ZZNS611.CICS.DFHCSD.PDS NONVSAM
DELETE ZZNS611.CICS.DFHCSD.SEQ NONVSAM
/*
//*/
EXEC PGM=IEFBR14
   DD DSN=ZZNS611.CICS.DFHCSD.SEQ,
   DD VOL=SER=CIXHØ2,UNIT=339Ø,
   DD DSN=ZZNS611.CICS.DFHCSD.PDS,
   DD VOL=SER=CIXHØ2,UNIT=339Ø,
   DD DSN=ZZNS611.CICS.DFHCSD.SEQ,DISP=SHR
EXEC PGM=DFHCSDUP,REGION=4Ø96K,
   DSN=ZZNS611.LOAD,DISP=SHR
EXEC PGM=IKJEFTØ1,DYNAMNBR=3Ø,REGION=4Ø96K
EXEC CICSCSD ZZNS611.CICS.DFHCSD.SEQ ZZNS611.CICS.DFHCSD.PDS

Espedito Morvillo
Systems Programmer (Italy)
Dynamic Transaction Routing

Here is some code you may find useful for CICS Dynamic Transaction Routing. The code is installed as a user-replaceable module and executes in the TOR whenever a transaction defined as DYNAMIC(YES) is executed or an undefined transaction code is entered.

You must define a PPT for the program and specify ‘DTRPGM=’ on the CICS start-up parameters.

We implemented this code to address a CPU constraint on one of our AORs. We now have a generic AOR, which we can replicate as many times as we need to fit our capacity requirements.

SYSTEM REQUIREMENTS
Our CICS system is Release 4.1. Code has not been tested on any other release. There are no special operating system requirements.

NAMING CONVENTION
A simple naming of sysids is used – any connection to the TOR with the same 3-character prefix in its sysid is considered to be an AOR eligible for transaction routing. AORs are searched for and found dynamically. This search is done every five minutes, enabling new AORs to be used shortly after they are started.

CICS AFFINITIES
Before you can route your transaction to more than one region, you have to understand any affinities that exist. The most common one is temporary storage for pseudoconversational programming.

Our application has this type of constraint. Although you can now function ship temporary storage requests, we chose not to do this. Instead, whenever a particular terminal does an application log-on, an AOR is selected. This AOR is used for that terminal until the next application log-on.
We still produced excellent load balancing with this technique.

If you cannot easily tell the start of an application affinity, you could simply route all transactions from a particular terminal to one AOR, deleting the affinity at CICS log-off time using a node error program.

Another affinity problem is CICS/VSAM. We function-shipped our CICS/VSAM to make this work. This caused a considerable CPU overhead, which we offset slightly by using shared data tables for some files. In our case, the overall CPU increase was a price worth paying for relief of the single TCB constraint of our AOR.

ROUTING ALGORITHM
The algorithm is a simple round-robin. Once an AOR is selected it is used until the next application-level log-on.

ROUTING ERRORS
The routing program routes only to AORs known to be acquired. The actual status is checked once every five minutes.

If a routing error occurs and no affinity exists, another AOR is tried automatically.

If an affinity already exists, an error message is sent. Pressing Enter forces an application relog-on (breaking the affinity) and another AOR is selected.

LCIDYP00

LCIDYP00 AMODE 31
***********************************************************************
*  LCIDYP00 CICS Dynamic Transaction routing exit for BANK
*  Select an AOR to service a dynamically-routed transaction.
*  AORs are selected round-robin at proiv log-on time (PRCL).
*  After log-on, an affinity exists and all transactions for that termid
*  are routed to the same AOR unless a routing error occurs.
*  AOR status is maintained in shared storage. Up to 10 (MAXAOR) AORs
*  Can be part of an AOR group. An AOR group is a set of AORs with the
* same 3 character SYSID prefix.

***********************************************************************
LCIDYPØØ RMODEL ANY
RØ   EQU 0
R1   EQU 1
R2   EQU 2
R3   EQU 3
R4   EQU 4
R5   EQU 5
R6   EQU 6
R7   EQU 7
R8   EQU 8
R9   EQU 9
R1Ø  EQU 1Ø
R11  EQU 11
R12  EQU 12
R13  EQU 13
R14  EQU 14
R15  EQU 15
LCIDYPØØ DFHEIENT CODEREG=(12),DATAREG=(13),EIBREG=(11)

* CHECK_FOR_COMMAREA EQU *
USING DFHDYPDS,5
USING TMP,7
USING AOR,6
ICM 5,15,DFHEICAP
BZ EXIT_POINT
CLI DYRDTRXN,C’Y’ Undefined tran code ?
BE EXIT_POINT Not interested

* MVC DYRRETC,RETURN_CODE8
CLC EIBCALEN.=AL2(DYRCLEN)
BNE EXIT_POINT

* CLI DYPFUNC,C’3’ INVOKED FOR STATIC ROUTE
BE EXIT_POINT

* CLI DYPFUNC,C’2’ INVOKED AFTER TRANSACTION END?
BE TRANTERM

* CLI DYPFUNC,C’4’ INVOKED AFTER AOR ABEND
BE ROUTE_ABEND
CLI DYPFUNC,C’Ø’ INITIAL INVOCATION OF ROUTER?
BE ROUTE_SELECT

* CLI DYPFUNC,C’1’ INVOKED DUE TO ROUTING ERROR?
BE ROUTE_SELECT

* MVC DYRRETC,RETURN_CODE8 INVALID REQUEST.
B EXIT_POINT
* ROUTE_SELECT EQU *
*            Must be invoked for route or route fail
*            so do common processing
*            MVI  PROCFLAG,X'00'
*            MVC  USERTSQ(4),=C'DYPQ'    TSQ with user affinities
MVC  USERTSQ+4(4),EIBTRMID  C'DYPQYYtermid
EXEC  CICS HANDLE CONDITION QIDERR(NO_USER_TSQ)  X
      ITEMERR(NO_USER_TSQ)
CHECK_FOR_USER_TSQ EQU *
EXEC  CICS READQ TS QUEUE(USERTSQ) INTO(USRDATA)  X
      ITEM(ENTRYNUM)  X
      NUMITEMS(MAXENTRIES)
OI  PROCFLAG,PROCUSER+PROCADDR  Sysid and storage address
L  R7,USRADDR  Address shared storage
NO_USER_TSQ EQU *
EXEC  CICS HANDLE CONDITION QIDERR() ITEMERR()
CLC  EIBTRNID,=C'PRCL'  Log-on transaction ?
BNE  NOT_LOGON_TRAN  No
TM  PROCFLAG,PROCUSER  User TSQ exists?
BNO  NOT_LOGON_TRAN  No don't delete it
EXEC  CICS DELETEQ TS QUEUE(USERTSQ)
NI  PROCFLAG,255-PROCUSER  Reset user TSQ flag
NOT_LOGON_TRAN EQU *
TM  PROCFLAG,PROCUSER  User TSQ exists ?
BNO  ROUTE_SELECT3  No go select route
CLI  DYRFUNC,C'1'  INVOKED DUE TO ROUTING ERROR?
BNE  ROUTE_SELECT2  No use existing affinity
****  Routing error occurred and
****  affinity exists.
****
LH  R6,USR_SLOT  Slot we tried to route to
LA  R15,TMPENTS  1st slot entry
SLL  R6,4  * entry length
AR  R6,R15  Entry we tried to route to
AP  AORFCNT,=P'1'  Update stats
NO_USABLE_AOR EQU *
*            Send error message to user
*            Return transid PRCL will break
*            affinity and attempt to select
*            another link
EXEC  CICS SEND  X
      FROM(ROUTE_ERROR)  X
      FLENGTH(=AL4(ROUTE_ERROR_LENGTH))  X
      ERASE
      MVC  DYRRETC,RETURN_CODE4  Terminate no error message
EXEC  CICS RETURN TRANSID('PRCL')
ROUTE_SELECT2 EQU *                    Existing affinity
LH R6,USRSLLOT               Get slot number for user
LA R15,DYRUSER              User area
AR R15,R6                   Index to our slot
MVI Ø(R15),X'FF'            Note attempted to route
SLL R6,4                   *16 = offset to slot
LA R15,TMPENTS              1st entry
AR R6,R15                   our entry
AP AORRcnt,-P'1'            Add 1 to routing count
MVC DYSYSID,USRYSYSID       We must use existing SYSID
B ROUTE_OK

ROUTE_SELECT3 EQU *                    Come here to select an AOR
*
EXEC CICS ASKTIME
   ABSTIME(TIMENOW)
EXEC CICS HANDLE CONDITION QIDERR(CREATE_TABLE) ITEMERR(CREATE_TABLE)
   TM PROCFLAG,PROCADDR     Is shared storage addr known?
   BO READ_AOR_ENTRY2      Yes - bypass TSQ read for addr
EXEC CICS ASSIGN SYSID(USRRTORID)
MVC TSQNAME(3),=-C'AOR'
MVC TSQNAME+3(3),USRRTORID  TSQ with shared storage ptr
MVC TSQNAME+6(2),=-C' '    C'AORxxx '
MVC LISTNAME,TSQNAME
LA R1,L'LISTNAME
STH R1,LISTNAME_LENGTH
*
READ_FIRST_AOR_ENTRY EQU *
MVC ENTRYNUM,-H'1'
*
READ_AOR_ENTRY EQU *
EXEC CICS READQ TS QUEUE(TSQNAME) ITEM(ENTRYNUM)
   ITEM(ENTRYNUM)
   NUMITEMS(MAXENTRIES)
   INTO(USRADDR)
   L R7,USRADDR
READ_AOR_ENTRY2 EQU *
ZAP TIMWORK,TIMENOW      Current time
TM TPMFLAG,TPMAORS       Useable AOR exists ?
BNO READ_AOR_ENTRY3     No - check 3Ø sec
SP TIMWORK,-P'3Ø0000'   - 5 minutes
B READ_AOR_ENTRY4
READ_AOR_ENTRY3 EQU *
SP TIMWORK,-P'3Ø000'    - 3Ø secs
READ_AOR_ENTRY4 EQU *
CP TIMWORK,TPMSCANT     Less than scan time?
BL LOCATE_CURRENT_AOR   Yes don't do scan
ZAP TMPSCANT,TIMENOW    Prevent anyone else doing scan
EXEC CICS FORMATTIME
   ABSTIME(TIMENOW)
   TIMESEP(':')
TIME(TMPSCANF)

* This code is driven once every 5 minutes during normal
* processing, or every 30 seconds if no AORs exist.
* The AOR status is compared to the actual connection.
* Any new AORs discovered are added to the table dynamically.

EXEC CICS INQUIRE CONNECTION CUSTOMER
NI   TMPFLAG,255-TMPAORS   Reset valid AOR flag

SCAN_AOR_LOOP EQU *
  EXEC CICS INQUIRE CONNECTION(CONN_NAME)
X    NETNAME(CONN_NETNAME)
X    CONNSTATUS(CONN_STATUS) RESP(RESPONSE_CODE) NEXT

*  CLC RESPONSE_CODE,DFHRESP(NORMAL)
  BNE SCAN_AOR_LOOP_END

*  CLC CONN_NAME(3),USRTORID   Matches tran SYSID prefix?
  BL SCAN_AOR_LOOP   No
  BH SCAN_AOR_LOOP_END   Processed all

*  Found connection we are interested in
  Now locate the AOR for this SYSID and update its status

  LA   R6,TMPENTS   Address 1st AOR

SCAN_AOR_LOOP2 EQU *
  TM   AORFLAG,AORUSED   Slot in use
  BO   SCAN_AOR_CHECK   Yes check if it is
  MVC   AORSYSID,CONN_NAME   Found a new AOR
  OI   AORFLAG,AORUSED   This slot is used
  OI   PROCFLAG,PROCNEW   New AOR found
  B    SCAN_AOR_FOUND

SCAN_AOR_CHECK EQU *
  CLC   AORSYSID,CONN_NAME   Is this our connection
  BNE   SCAN_AOR_NEXT   No loop to next
  NI   PROCFLAG,255-PROCNEW   Not a new AOR

SCAN_AOR_FOUND EQU *
  BAL   R14,STATUS_CHECK   check/set AOR status
  B    SCAN_AOR_LOOP   Check next connection

SCAN_AOR_NEXT EQU *
  LA   R6,AORLEN(,R6)   Next AOR
  B    SCAN_AOR_LOOP2

*  

SCAN_AOR_LOOP_END EQU *
  EXEC CICS INQUIRE CONNECTION END
  ZAP   TIMEWORK,TIMENOW   Current time
  SP   TIMEWORK,=P'3600000'   - 60 minutes

CP   TIMEWORK,TMPSTATT       Less than stat time?
BL   LOCATE_CURRENT_AOR      Yes don't do stats
ZAP  TIMEWORK,TMPSTATT       Save last stats time
ZAP  TMPSTATT,TIMENOW         Prevent anyone else doing stat
EXEC CICS FORMATTIME                                           X
   ABSTIME(TIMENOW)                                      X
   TIMESEP(':')                                             X
   TIME(TMPSTATF)
*
* This code is driven once every 60 minutes
* The AOR stats are written to SYSLOG
*
MVC  MSGWORK,MSG01            Move constants
EXEC CICS FORMATTIME                                     X
   ABSTIME(TIMENOW)                                    X
   TIMESEP(':')                                      X
   TIME(TIMEF)
   R1,MSGWORK                                             
   MSG01TIM-MSG01(L'MSG01TIM,R1),TIMEF
EXEC CICS WRITE OPERATOR                                 X
   TEXT(MSGWORK)
LA   R6,TMPENTS                                    1st AOR entry
STATS_LOOP EQU *
TM   AORFLAG,AORUSED            AOR in use?
BNO  LOCATE_CURRENT_AOR       No - all AORs processed
MVC  MSGWORK,MSG02            Move constants
   R1,MSGWORK                                            
   MSG02AOR-MSG02(L'MSG02AOR,R1),AORSYSID Move name
ED   MSG02RC-MSG02(L'MSG02RC,R1),AORRCNT Move route count
ED   MSG02RF-MSG02(L'MSG02RF,R1),AORFCNT Move fails count
MVC  MSG02STA-MSG02(L'MSG02STA,R1),='ACQ'    Assume acquired
TM   AORFLAG,AORACQ            Aquired ?
BO   STATS_LOOP_MSG           Yes
   MSG02STA-MSG02(L'MSG02STA,R1),='REL'  must be REL
STATS_LOOP_MSG EQU *
   EXEC CICS WRITE OPERATOR                             X
   TEXT(MSGWORK)
   ZAP  AORFCNT,='P'0'            Reset
   ZAP  AORRCNT,='P'0'            Reset
   R6,AORLEN(,R6)                                Next entry
B    STATS_LOOP               Process
*
* This is the start of the AOR
*
LOCATE_CURRENT_AOR EQU *
CLI  DYRFUNC,='C1'            Invoked due to routing error?
BNE  LOCATE_CURRENT_AOR2     No
LH   R6,USRSLOT               Get slot number we used
SLL  R6,4                     * ent len
LA   R15,TMPENTS              1st slot entry
AR   R6,R15                   Entry we tried to route
AP AORFCNT,=P'1'          Add 1 to fail count
LOCATE_CURRENT_AOR2 EQU *
LA R15,TMPENTS            1st slot entry
LH R6,TMPNEXT             Next AOR slot
STH R6,SAVENEXT          Save for loop check
SLL R6,4                *16 (SLOT LEN)
AR R6,R15               Point to next slot
*
TRY_THIS_AOR EQU *
TM AORFLAG,AORUSED+AORACQ AOR OK ?
BNO TRY_NEXT_AOR
LA R15,DYRUSER            User area
LH R1,TMPNEXT             Current entry
AR R15,R1                Index
CLI Ø(R15),X'FF'        Already tried to route this AOR?
BE TRY_NEXT_AOR        Yes try another
MVC DYSYSID,AORSYSID     Use this AOR
USE_THIS_AOR EQU *
MVC USRSYSID,AORSYSID   Remember SYSID to use this user
MVC USRSLOT,TMPNEXT     Remember slot number
LH R1,TMPNEXT
LA R15,DYRUSER        User area
AR R15,R1            Index to slot number
MVI Ø(R15),X'FF'    Note attempted to route to AOR
AP AORRCNT,=P'1'       Add 1 to stats
LA R6,AORLEN(R6)      Next slot
TM AORFLAG,AORUSED   Slot in use?
BO REWRITE_ENTRY     Yes can use this one next
MVC TMPNEXT,=H'-1'   End of used slots - go back
*
REWRITE_ENTRY EQU *
LH R1,TMPNEXT
LA R1,1(R1)          Next slot
STH R1,TMPNEXT      Save for next user
*
Create user TSQ to route all subsequent Trans till next log-on
* (PRCL) is issued
EXEC CICS WRITEQ TS QUEUE(USERTSQ) X
ITEM(ENTRYNUM)        X
FROM(USRDATA)         X
MAIN
ROUTE_OK EQU *
MVI DYROPTER,C'N'
MVC DYRETC,RETURN_CODEØ
B EXIT_POINT
*
TRY_NEXT_AOR EQU *
LA R6,AORLEN(R6)      Point to next AOR
TM AORFLAG,AORUSED   At end of AORs?
BO TRY_NEXT_AOR2     No - try next
MVC TMPNEXT,=H'-1'  Go back to AOR Ø
LA R6,TMPENTS        Point to AOR Ø
TRY_NEXT_AOR2 EQU *
LH R1,TMPNEXT
LA R1,1(R1) Next AOR
STH R1,TMPNEXT
CLC TMPNEXT,SAVENEXT Loop check
BNE TRY_THIS_AOR Not looped
B NO_USABLE_AOR
*
TRANTERM EQU *
MVI DYROPTER,C'N'
MVC DYRRETC,RETURN_CODE8
B EXIT_POINT
*
ROUTE_ABEND EQU *
MVI DYROPTER,C'N'
MVC DYRRETC,RETURN_CODEØ
B EXIT_POINT
*
CREATE_TABLE EQU *
*
Create AOR status table.
* Because many tasks might try to do this, ENQ is issued and
* retest is done after successful enq.
*
EXEC CICS ENQ RESOURCE(LISTNAME) LENGTH(LISTNAME_LENGTH)
EXEC CICS HANDLE CONDITION X
QIDERR(CREATE_TABLE_TSQ)
*
EXEC CICS READQ TS QUEUE(TSQNAME) INTO(USRADDR) X
ITEM(ENTRYNUM)
B GET_LIST_OF_AORS_RETURN TS CREATED DURING ENQUEUE
CREATE_TABLE_TSQ EQU *
*
EXEC CICS GETMAIN X
SET(R7) X
INITIMG(X'ØØ') X
FLENGTH(AL4(TMPLEN)) X
SHARED
ST R7,USRADDR
MVC MSGWORK,MSGØ4 Init message
LA R2,MSGWORK
UNPK HEXWORK,USRADDR(5)
TR HEXWORK(8),HEXTAB
MVC MSGØ4STO-MSGØ4(L'MSGØ4STO,R2),HEXWORK
EXEC CICS WRITE OPERATOR X
TEXT(MSGWORK)
MVC TMPNEXT,=H'Ø' First AOR entry
MVC TMPSCANT,TIMENOW Set time of last scan
MVC TMPSTATT,TIMENOW Set time of last stats init
EXEC CICS FORMATTIME X
ABSTIME(TIMENOW) X
TIMESEP(':') X
TIME(TMPSCANF)
MVC TMPSTATF,TMPSCANF
LA R6,TMPENTS 1st entry
MVC ENTRYNUM=H'1'
OI PROCFLAG,PROCNEW Note all AORs are new
GET_LIST_OF_AORS EQU *
EXEC CICS INQUIRE CONNECTION START
NI TMPFLAG,255-TMPAORS Reset valid AOR flag
GET_AOR_LOOP EQU *
EXEC CICS INQUIRE CONNECTION(CONN_NAME) X
NETNAME(CONN_NETNAME) X
CONNSTATUS(CONN_STATUS) RESP(RESPONSE_CODE) NEXT
* 
CLC RESPONSE_CODE,DFHRESP(NORMAL)
BNE GET_LIST_OF_AORS_END *
CLC CONN_NAME(3),USRTORID Matches tran SYSID prefix?
BL GET_AOR_LOOP Less loop till we get there
BH GET_LIST_OF_AORS_END All our entries processed
OI AORFLAG,AORUSED This slot is used
MVC AORSYSID,CONN_NAME AOR SYSID
ZAP AORFCNT,=P'Ø' Route fail count
ZAP AORRCNT,=P'Ø' Route count *
BAL R14,STATUS_CHECK Check/set AOR Status
LA R6,AORLEN(,R6) Next entry
B GET_AOR_LOOP *
GET_LIST_OF_AORS_END EQU *
EXEC CICS INQUIRE CONNECTION END
WRITE_ENTRIES EQU *
EXEC CICS WRITEQ TS QUEUE(TSQNAME) X
ITEM(ENTRYNUM) X
FROM(USRADDR) X
MAIN *
B GET_LIST_OF_AORS_RETURN *
GET_LIST_OF_AORS_RETURN EQU *
EXEC CICS DEQ RESOURCE(LISTNAME) LENGTH(LISTNAME_LENGTH)
EXEC CICS HANDLE CONDITION X
QIDERR()
B READ_FIRST_AOR_ENTRY *
STATUS_CHECK EQU *
ST R14,STATUS_R14
* Compare AOR status to actual
* Update status if required
* Issue WTO if status has changed, or if 1st time
* This code is driven each AORSCAN, and also on first
* time building of shared storage.
* A message is issued if the status changes unless PROCNEW
* is set, in which case the message is unconditional.
* 
DS ØH
MVC MSGWORK,MSGØ3 Move message constants
LA R1,MSGWORK
MVC MSGØ3AOR-MSGØ3(L'MSGØ3AOR,R1),AORSYSID
CLC CONN_STATUS,DFHVALUE(ACQUIRED)
BNE STATUS_CHECK_REL No
MVC MSGØ3STA-MSGØ3(L'MSGØ3STA,R1),=C'ACQ'
OI TMPFLAG,TPAORS Note usable AOR exists
TM AORFLAG,AORACQ We think acquired ?
BO STATUS_CHECK_MSG Do we need to issue MSG?
OI AORFLAG,AORACQ AOR is acquired
B STATUS_CHECK_MSG2 We definitely need to issue MSG

STATUS_CHECK_REL EQU *
MVC MSGØ3STA-MSGØ3(L'MSGØ3STA,R1),=C'REL'
TM AORFLAG,AORACQ We think released ?
BZ STATUS_CHECK_MSG Do we need to issue MSG?
NI AORFLAG,255-AORACQ AOR is released
B STATUS_CHECK_MSG2 We definitely need to issue MSG

STATUS_CHECK_MSG EQU *
TM PROCFLAG,PROCNEW Creating new entry ?
BNO STATUS_CHECK_EXIT No - no need to issue MSG

STATUS_CHECK_MSG2 EQU *
EXEC CICS WRITE OPERATOR TEXT(MSGWORK)

STATUS_CHECK_EXIT EQU *
L R14,STATUS_R14
BR R14
EJECT

ROUTE_ERROR EQU *
DC CL8Ø'LCIDYP1Ø Your request could not be processed due to
a system problem.'
DC CL8Ø'Press ENTER to relog-on to the Caller system.'
DC CL8Ø'If this problem persists, please contact the Helpd
esk.'

ROUTE_ERROR_LENGTH EQU *-ROUTE_ERROR

* 
MSGØ1 DC CL8Ø' '
ORG MSGØ1
DC 'LCIDYPØ1 - AOR statistics follow (reset at

MESSAGETIM DC CL8
DS CL8
DC C').' ORG

* 
MSGØ2 DC CL8Ø' '
ORG MSGØ2

© 2000. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.
DC C'LCIDYP02 - AOR : '
MSG02AOR DS CL4
DC C' Status : '
MSG02STA DS CL3
DC C' Routed count : '
MSG02RC DC X'40202020202120'
DC C' failures : '
MSG02RF DC X'40202020202120'
DC C'.'

ORG

MSG03 DC CL80' '
ORG MSG03
DC C'LCIDYP03 - AOR : '
MSG03AOR DS CL4
DC C' Status is '
MSG03STA DS CL3
DC C'.'

ORG

MSG04 DC CL80' '
ORG MSG04
DC C'LCIDYP04 - Initializing shared storage at '
MSG04STO DS CL8
DC C'.'

ORG

RETURN_CODE0 DC F'0'
RETURN_CODE4 DC F'4'
RETURN_CODE8 DC F'8'
HEXTAB DC 256X'00'
ORG HEXTAB+X'F0'
DC C'0123456789ABCDEF'
ORG
LTORG ,

EXIT_POINT DS 0H
EXEC CICS RETURN

*  DFHEISTG
TIMENOW DS PL8
TIMWORK DS PL8
RESPONSE_CODE DS F
CONN_STATUS DS F
STATUS_R14 DS F
SAVENEXT DS H
LISTNAME_LENGTH DS AL2
LISTNAME DS CL8
TSQNAME DS CL8
USERTSQ DS CL8
HEXWORK DS CL9
Interfacing CICS to SMTP – part 1

This package is intended to demonstrate one technique for interfacing CICS with the SMTP mail capability of TCP/IP for MVS. This package is only intended to send outbound mail and has no provision for reading mail responses, although the same technique could be used to do so.

MAILMAP.CPY

Ø1 CONFIRMI.
Ø2 FILLER PICTURE X(12).
Ø2 FILLER PICTURE X.
Ø1 CONFIRM O REDEFINES CONFIRMI.
Ø2 FILLER PICTURE X(12).
Ø2 FILLER PICTURE X.
Ø1 ERRORI.
Ø2 FILLER PICTURE X(12).
Ø2 FILLER PICTURE X.
Ø1 ERRORO REDEFINES ERRORI.
Ø2 FILLER PICTURE X(12).
Ø2 FILLER PICTURE X.
Ø1 MAILMAPI.
Ø2 FILLER PICTURE X(12).
Ø2 MAIL-DATEL COMP PICTURE S9(4).
Ø2 MAIL-DATEF PICTURE X.
Ø2 MAIL-DATEI PICTURE X(Ø1Ø).
Ø2 MAIL-APPLIDL COMP PICTURE S9(4).
Ø2 MAIL-APPLIDF PICTURE X.
Ø2 MAIL-APPLIDI PICTURE X(ØØ8).
Ø2 MAIL-TIMEL COMP PICTURE S9(4).
Ø2 MAIL-TIMEF PICTURE X.
Ø2 MAIL-TIMEI PICTURE X(ØØ8).
Ø2 MAIL-USERIDL COMP PICTURE S9(4).
Ø2 MAIL-USERIDF PICTURE X.
Ø2 MAIL-USERIDI PICTURE X(ØØ8).
Ø2 MAIL-TOL COMP PICTURE S9(4).
Ø2 MAIL-TOF PICTURE X.
Ø2 MAIL-TOI PICTURE X(Ø4Ø).
Ø2 MAIL-REPLYL COMP PICTURE S9(4).
Ø2 MAIL-REPLYF PICTURE X.
Ø2 MAIL-REPLYI PICTURE X(Ø4Ø).
Ø2 MAIL-FROML COMP PICTURE S9(4).
Ø2 MAIL-FROMF PICTURE X.
Ø2 MAIL-FROMI PICTURE X(Ø2Ø).
Ø2 MAIL-SUBJECTL COMP PICTURE S9(4).
Ø2 MAIL-TIMEA PICTURE X.
Ø2 MAIL-TIMEO PICTURE X(ØØ8).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-USERIDA PICTURE X.
Ø2 MAIL-USERIDO PICTURE X(ØØ8).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-TOA PICTURE X.
Ø2 MAIL-TOO PICTURE X(Ø4Ø).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-REPLYA PICTURE X.
Ø2 MAIL-REPLYO PICTURE X(Ø4Ø).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-FROMA PICTURE X.
Ø2 MAIL-FROMO PICTURE X(Ø2Ø).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-SUBJECTA PICTURE X.
Ø2 MAIL-SUBJECTO PICTURE X(Ø69).
Ø2 BMSGØØ2-MAILMAP-MAILMAP OCCURS 15 TIMES.
  Ø3 FILLER PICTURE X(2).
  Ø3 MAIL-BODYA PICTURE X.
  Ø3 MAIL-BODYO PICTURE X(Ø79).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F1A PICTURE X.
Ø2 MAIL-F1O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F2A PICTURE X.
Ø2 MAIL-F2O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F3A PICTURE X.
Ø2 MAIL-F3O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F4A PICTURE X.
Ø2 MAIL-F4O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F5A PICTURE X.
Ø2 MAIL-F5O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F6A PICTURE X.
Ø2 MAIL-F6O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F7A PICTURE X.
Ø2 MAIL-F7O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F8A PICTURE X.
Ø2 MAIL-F8O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F9A PICTURE X.
Ø2 MAIL-F9O PICTURE X(ØØ5).
Ø2 FILLER PICTURE X(2).
Ø2 MAIL-F10A PICTURE X.
Ø2 MAIL-F10O PICTURE X(ØØ6).
MAILTEST.CBL

IDENTIFICATION DIVISION.

* *****************************************************
* IDENTIFICATION DIVISION                           *
* *****************************************************

PROGRAM-ID.       MAILTEST.

AUTHOR.           DONALD GRINSELL.

INSTALLATION.     STATE OF MONTANA.


DATE-COMPILED.    

REMARKS.          CICS MAIL TRANSACTION

THIS PROGRAM IS USED TO TEST THE SENDMAIL PGM.

******************************************************************
EJECT
ENVIRONMENT DIVISION.

* *****************************************************
* ENVIRONMENT DIVISION                               *
* *****************************************************

CONFIGURATION SECTION.

SOURCE-COMPUTER.   IBM-370.

OBJECT-COMPUTER.   IBM-370.

******************************************************************
EJECT
DATA DIVISION.

******************************************************************
WORKING-STORAGE SECTION.
SKIP1
******************************************************************
*       WORKING STORAGE SECTION       *
******************************************************************
Ø1  MISC-DATA.
Ø5  RESPONSE-CODE           PIC S9(8) COMP.
Ø5  TERMINAL-MSG-LEN        PIC S9(4) COMP.
Ø5  TERMINAL-MSG            PIC X(80) VALUE SPACES.
Ø5  WS-MAPSET-NAME          PIC X(8)   VALUE 'MAILMAP'.
Ø5  WS-MAP-CONFIRM          PIC X(8)   VALUE 'CONFIRM'.
Ø5  WS-MAP-ERROR            PIC X(8)   VALUE 'ERROR'.
Ø5  WS-ABSTIME              PIC S9(16).
Ø5  WS-ERROR-FLAG           PIC X      VALUE LOW-VALUES.
Ø5  WS-INDEX                PIC 9(2).
******************************************************************
*                 C I C S    C O N S T A N T S                   *
******************************************************************
SKIP3
COPY DFHAID.
EJECT
COPY DFHBMSCA.
EJECT
******************************************************************
*                 C I C S    M A P S                             *
******************************************************************
SKIP3
COPY MAILMAP.
SKIP3
******************************************************************
*                 S E N D M A I L   A R E A                       *
******************************************************************
SKIP3
Ø1  SEND-MAIL-PARMS.
Ø5  SEND-MAIL-HEAD1.
  1Ø SEND-RCPT              PIC X(40).
  1Ø SEND-REPLY1            PIC X(40).
Ø5  SEND-RESPONSE REDEFINES SEND-MAIL-HEAD1 PIC X.
  88 SEND-ERROR             VALUE HIGH-VALUES.
Ø5  SEND-MAIL-HEAD2.
  1Ø FILLER                 PIC X(4) VALUE 'TO:'.
  1Ø SEND-TO                PIC X(40).
  1Ø FILLER                 PIC X(36) VALUE SPACES.
Ø5  SEND-MAIL-HEAD3.
  1Ø FILLER                 PIC X(6) VALUE 'FROM:'.
  1Ø FILLER                 PIC X VALUE '"'.
  1Ø SEND-FROM              PIC X(20).
  1Ø FILLER                 PIC X VALUE '"'.
  1Ø FILLER                 PIC X VALUE '<'.
  1Ø FILLER                 PIC X VALUE SPACES.
  1Ø SEND-REPLY2            PIC X(40) VALUE SPACES.
EXEC CICS RECEIVE MAP(WS-MAPSET-NAME)
    MAPSET(WS-MAPSET-NAME)
    INTO(MAILMAPI)
    ASIS
    RESP(RESPONSE-CODE)
END-EXEC.
IF EIBAID = DFHCLEAR OR
   EIBAID = DFHPF3 OR
   EIBAID = DFHPF15 GO TO 9999-EXIT-CICS.
IF EIBAID = DFHPF12 OR
   EIBAID = DFHPF24
   MOVE LOW-VALUES TO MAILMAPO
   PERFORM 6000-BUILD-HEADER
   PERFORM 7000-SEND-ERASE
   GO TO 9000-RETURN-CICS.
IF RESPONSE-CODE NOT = DFHRESP(NORMAL)
   MOVE LOW-VALUES TO MAILMAPO
   PERFORM 6000-BUILD-HEADER
   PERFORM 7000-SEND-ERASE
   GO TO 9000-RETURN-CICS.
* HERE WE NEED TO CHECK THE LENGTH OF THE MAP INPUT FIELDS *
* AT A MINIMUM MAIL-TOL, MAIL-FROMML AND MAIL-SUBJECML MUST *
* BE GREATER THAN ZERO. IF THEY ARE WE CAN SEND THE MESSAGE. *
* IF THEY ARE NOT WE MUST POSITION THE CURSOR BY MOVING -1 TO *
* THE APPROPRIATE L FIELD AND PERFORM SEND-OVER ROUTINE. *
* OTHERWISE WE CAN GO AHEAD AND MOVE THE MAP DATA TO THE *
* COMMAREA AND LINK TO SENDMAIL.                          *

******************************************************************

IF MAIL-TOL = ZERO OR MAIL-TOI = SPACES THEN
  MOVE HIGH-VALUES TO WS-ERROR-FLAG
  MOVE -1 TO MAIL-TOL.

IF MAIL-REPLYL = ZERO OR MAIL-REPLYI = SPACES THEN
  MOVE HIGH-VALUES TO WS-ERROR-FLAG
  MOVE -1 TO MAIL-REPLYL.

IF MAIL-FROML = ZERO OR MAIL-FROMI = SPACES THEN
  MOVE HIGH-VALUES TO WS-ERROR-FLAG
  MOVE -1 TO MAIL-FROML.

IF MAIL-SUBJECTL = ZERO OR MAIL-SUBJECTI = SPACES THEN
  MOVE HIGH-VALUES TO WS-ERROR-FLAG
  MOVE -1 TO MAIL-SUBJECTL.

IF WS-ERROR-FLAG = HIGH-VALUES THEN
  PERFORM 6000-BUILD-HEADER
  PERFORM 7500-SEND-OVER
  GO TO 9000-RETURN-CICS.

IF EIBAID = DFHPF9 OR EIBAID = DFHPF21 THEN
  PERFORM 4000-SEND-MAIL
  IF NOT SEND-ERROR THEN
    PERFORM 7700-SEND-CONFIRM
    GO TO 9000-RETURN-CICS
  ELSE
    PERFORM 8000-DUMP-TASK
    PERFORM 7800-SEND-ERROR
    GO TO 9000-RETURN-CICS
  END-EXEC.

ELSE
  PERFORM 6000-BUILD-HEADER
  MOVE -1 TO MAIL-BODYL(1)
  PERFORM 7500-SEND-OVER
  GO TO 9000-RETURN-CICS.

4000-SEND-MAIL.
  MOVE MAIL-TOI TO SEND-RCPT SEND-TO.
  MOVE MAIL-REPLYI TO SEND-REPLY1 SEND-REPLY2.
  MOVE MAIL-FROMI TO SEND-FROM.
  MOVE MAIL-SUBJECTI TO SEND-SUBJECT.
  PERFORM 4000-MOVE-BODY VARYING WS-INDEX FROM 1 BY 1
    UNTIL WS-INDEX IS GREATER THAN 15.
  EXEC CICS LINK PROGRAM('SENDMAIL')
    COMMAREA(SEND-MAIL-PARMS)
  END-EXEC.

4000-MOVE-BODY.
  MOVE MAIL-BODYI(WS-INDEX) TO SEND-BODY(WS-INDEX).

6000-BUILD-HEADER.
  EXEC CICS ASSIGN
    APPLID(MAIL-APPLIDO)
    USERID(MAIL-USERIDO)
    USERNAME(MAIL-FROMO)
  END-EXEC.
EXEC CICS ASKTIME
   ABSTIME (WS-ABSTIME)
END-EXEC.
EXEC CICS FORMATTIME
   ABSTIME (WS-ABSTIME)
   MMDDYYYY (MAIL-DATEO)
   DATESEP
   TIME (MAIL-TIMEO)
   TIMESEP
END-EXEC.

7000-SEND-ERASE.
EXEC CICS SEND
   MAP (WS-MAPSET-NAME)
   MAPSET (WS-MAPSET-NAME)
   FROM (MAILMAPO)
   ERASE
END-EXEC.

7500-SEND-OVER.
EXEC CICS SEND
   MAP (WS-MAPSET-NAME)
   MAPSET (WS-MAPSET-NAME)
   FROM (MAILMAPO)
   CURSOR
END-EXEC.

7700-SEND-CONFIRM.
EXEC CICS SEND
   MAP (WS-MAP-CONFIRM)
   MAPSET (WS-MAPSET-NAME)
   MAPONLY
END-EXEC.

7800-SEND-ERROR.
EXEC CICS SEND
   MAP (WS-MAP-ERROR)
   MAPSET (WS-MAPSET-NAME)
   MAPONLY
END-EXEC.

8000-DUMP-TASK.
EXEC CICS DUMP DUMPCODE('ERR1')
END-EXEC.

9000-RETURN-CICS.
EXEC CICS
   RETURN TRANSID(EIBTRNID)
END-EXEC.

9999-EXIT-CICS.
MOVE +80 TO TERMINAL-MSG-LEN.
MOVE ' TRANSACTION TERMINATED / PRESS CLEAR TO CONTINUE' TO TERMINAL-MSG.
EXEC CICS
   SEND FROM(TERMINAL-MSG) LENGTH(TERMINAL-MSG-LEN)
   ERASE LAST
END-EXEC.
EXEC CICS
SENDMAIL.ASM

TITLE 'SENDMAIL TCPSMTP MAIL INTERFACE'
***********************************************************************
* SENDMAIL THIS PROGRAM IS USED TO SEND E-MAIL FROM CICS *
* USING THE CICS SPOOL INTERFACE. *
* COMMAREA FORMAT: *
*   40 BYTES RECIPIENT EMAIL ADDRESS *
*   40 BYTES SENDER EMAIL ADDRESS *
*   80 BYTES TO: *
*   80 BYTES FROM: *
*   80 BYTES SUBJECT: *
*   80 BYTES MESSAGE TEXT *
* .... *
* 80 BYTES END OF MESSAGE LINES *
* NOTE THAT THE COMMAREA MUST BE A MULTIPLE OF 80 CHARACTERS *
* AND BE A MINIMUM OF 5*80 OR 400 BYTES LONG AND A MAXIMUM OF *
* 32K (CICS COMMAREA LIMIT). *
* RETURN CODES: *
* FF ERROR *
* EXAMPLE: *
* CA       DS 0D *
* CARCPT   DC CL40'DGRINSELL@STATE.MT.US' *
* CAFROM   DC CL40'CZ0055@HLNCTR STATE.MT.US' *
* ORG      CA *
* CARESP   DC XLI'00' *
* ORG      *
* CAMSG1   DC CL68'TO: DONALD GRINSELL' *
* CAMSG2   DC CL80'FROM: "CZ0055 ON CICS"<DGRINSELL@STATE.MT.US> *
* CAMSG3   DC CL80'SUBJECT: MAIL FROM CICS' *
* CAMSG4   DC CL80' ' *
* CAMSG5   DC CL80'THIS IS MY MESSAGE' *
* CALEN    DC Y(*-CA) *
* EXEC CICS LINK PROGRAM('SENDMAIL') *
*   COMMAREA(CA) LENGTH(CALEN) *
* CLI CARESP,X'FF' Q-ERROR RETURN CODE *
* BE SENDFAIL BIF YES, ELSE CONTINUE *
***********************************************************************
EJECT
***********************************************************************

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

Donald A Grinsell
CICS Systems Programmer
State of Montana (USA) © Donald A Grinsell 2000
IBM has tweaked its XML Toolkit for OS/390 with the XML Parser for OS/390, Java Edition and the XML Parser for OS/390, C++ Edition, both based on cross-platform open-source code and compliant with industry standards.

The toolkit, we’re told, reduces development time by providing tested functions, promotes code re-use, and supports ASCII, EBCDIC, and Unicode data.

It is positioned to complement WebSphere Application Server for OS/390, Developer Kit for OS/390, and MQSeries.

The OS/390 XML Parser Java Edition can work with CICS TS 1.1 by exploiting the transaction system’s ability to run a Java Virtual Machine within itself.

Interoperability with CICS TS 1.3 is supported via the CICS Transaction Gateway 3.1 which transmits application requests to CICS.

For further information contact your local IBM representative.

* * *

Dynasty Technologies has announced availability of its native support for System/390 systems via Enterprise SpA, Dynasty’s development partner for System/390 platforms, which has ported its existing Dynasty foreign exchange application to a native CICS and DB2 Universal Database.

This adds to the ability to deploy application servers on NT, Windows 2000, Sun Solaris, HP, RS6000, NonStop Himalaya, and AS/400 servers.

Enterprise SpA is an Italian banking software house, which ported its PREMIA foreign exchange application.

For further information contact:
Dynasty Technologies, 101 Redwood Shores Parkway, #200 Redwood shores, CA 94065, USA.
Tel: (650) 631 5889.

* * *

IBM has announced its Developer Kit for OS/390, Java 2 Technology Edition, JDK 1.3 environment for System/390s. Besides Java 2 APIs, additional functions include Java Record I/O (JRIO) and security API facilities. It runs on System/390s running OS/390 Version 2.8 (or later) with floating point instructions.

JVM implementations are intended to allow applications written in Java for OS/390 subsystems such as CICS TS, IMS, and DB2 stored procedures. IBM says it plans to add these new technologies to the existing base JVM, while maintaining the Java compatibility of existing applications in Java deployed on OS/390.

For further information contact your local IBM representative.
URL: http://www.alphaworks.ibm.com/aw.nsf/textOverviews/8638C67170C0ED5D882568DD0077C8FF.