180 CICS update

November 2000

In this issue

CICS Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38342
From USA: 01144 1635 38342
E-mail: trevore@xephon.com

North American office

Xephon
PO Box 350100
Westminster, CO 80035-0100
USA
Telephone: 303 410 9344

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £175.00 in the UK; $270.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £16.00 ($23.50) each including postage.

CICS Update on-line

Code from *CICS Update* can be downloaded from our Web site at http://www.xephon.com/cicsupdate.html; you will need the user-id shown on your address label.

Editor

Trevor Eddolls

Disclaimer

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

Contributions

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

Printed in England.

Cold start next time – revisited

CICS Update, Issue 167, October 1999, described a 'system' to set CICS to cold start next time it loads. That 'system' had a small loophole. The system worked by creating a SHUTOK dataset after the CICS step completed, then checking for this dataset before REPROing the initialization record into the DFHGCD to force a cold start.

It was possible for the CICS step to complete with RC=0, but a warm keypoint had not been taken. The 'system' would create a SHUTOK dataset in this situation and therefore allow the DFHGCD to be re-initialized and force a cold start when an emergency restart was required.

The following small program closes that loophole. The program queries the DFHGCD for the warm keypoint indicator. Using this program removes the need for the SHUTOK dataset, so the SHUTOK step can be removed and the SHUTCHK step can be replaced with the following:

```
//*
//*====================================================================*
//*          CHECK CICS WARM KEYPOINT                                  *
//*====================================================================*
//*
//SHUTCHK  EXEC PGM=LTWARMCK
//DFHGCD   DD  DSN=&REGION..DFHGCD,
//             DISP=SHR
//SYSPRINT DD  SYSOUT=*
```

The program is written in C and works against a CICS Transaction Server for OS/390 Version 1.3. Global Catalog dataset. It is very simple and could be converted to any language. The program locates the DFHGCD key '....DFHRMDM DFHRMDM_ANCHOR ' (.... = X'00000011' and there are two spaces after ANCHOR). It then checks offset 20 in the record – X'03' indicates a warm keypoint was taken, X'04' indicates an emergency restart is required. The program then sets the return code to 0 if a warm keypoint was taken and 4 if not.

```
/*********************************************************************
 * LTWARMCK : Check CICS DFHGCD for Warm keypoint indicator          *
 * v1.0.0.  :                                                        *
```

```
 ************************************************************************
 * Hardware and Software Requirements                                   *
 * -------------------------------------------------------------------- *
 * Language: C (LE - use of flocate)                                    *
 * CPU: All that run MVS/ESA and C                                      *
 * Operating System: MVS/ESA                                            *
 * CICS Version/Release: CICS Transaction Server for OS/390 v1.3.       *
 ************************************************************************
 * Function                                                             *
 * -------------------------------------------------------------------- *
 *  This program reads the CICS DFHGCD file and checks the Warm         *
 * keypoint record: '....DFHRMDM DFHRMDM_ANCHOR ' (.... = x'00000011')*
 * The Warm keypoint indicator is then at offset 19,2 -                 *
 *                                                                      *
 * x'0003' - Warm keypoint taken                                        *
 * x'0004' - Emergency restart required                                 *
 ************************************************************************/
/*----------------------------------------------------------------------*
 * #includes                                                            *
 *----------------------------------------------------------------------*/
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
/*----------------------------------------------------------------------*
 * #defines and typedefs                                                *
 *----------------------------------------------------------------------*/
#define INDD "DD:DFHGCD"
#define DFHGCD_KEYLEN 28
#define RC_OK 0
#define RC_WARN 4
#define RC_FAILURE 12
#define KEYPOINT_WARM 0x3
#define KEYPOINT_EMER 0x4
/*----------------------------------------------------------------------*
 * global variables                                                     *
 *----------------------------------------------------------------------*/
/*----------------------------------------------------------------------*
 * function prototypes                                                  *
 *----------------------------------------------------------------------*/
void error_exit(int);
/*----------------------------------------------------------------------*
 * main()                                                               *
 *----------------------------------------------------------------------*/
int main()                              /* ==> Entry point              */
{                                       /*                              */
/*----------------------------------------------------------------------*
 * local variables                                                      *
 *----------------------------------------------------------------------*/
                                        /*------------------------------*/
```

```
FILE *dfhgcd;                          /* Input file (DFHGCD)         */
                                       /*-----------------------------*/
struct dfhgcd_rec                      /* DFHGCD record layout        */
{                                      /*                             */
  unsigned int key_hex;                /* Record key hex              */
  char key_char??(24??);               /* Record key char             */
  char filler??(19??);                 /* ** Filler **                */
  char keypoint;                       /* Keypoint indicator          */
};                                     /*                             */
                                       /*-----------------------------*/
_Packed struct dfhgcd_rec              /* DFHGCD anchor record        */
 dfhgcd_rec_anch =                     /*                             */
{                                      /*                             */
  0x11,                                /* Record key hex              */
  "DFHRMDM DFHRMDM_ANCHOR  ",          /* Record key char             */
  "                    ",              /* ** Filler **                */
  0                                    /* Keypoint indicator          */
};                                     /*                             */
                                       /*-----------------------------*/
_Packed struct dfhgcd_rec              /* Pointer at anchor record    */
 *dfhgcd_rec_anch_ptr =                /*                             */
 &dfhgcd_rec_anch;                     /*                             */
                                       /*-----------------------------*/
int rc_warm;                           /* Return code                 */
                                       /*-----------------------------*/
/*--------------------------------------------------------------------*
 * code                                                               *
 *--------------------------------------------------------------------*/
                                       /*                             */
rc_warm = RC_WARN;                     /* Set return code             */
if(dfhgcd = fopen(INDD, "rb, type=record"))
{                                      /* Open DFHGCD                 */
  if(!flocate(dfhgcd,                  /* Locate Anchor record        */
      &dfhgcd_rec_anch.key_hex,        /*                             */
      DFHGCD_KEYLEN,                   /*                             */
      __KEY_EQ))                       /*                             */
  {                                    /*                             */
    fread(dfhgcd_rec_anch_ptr,         /* Read Anchor record          */
      1,                               /*                             */
      sizeof(struct dfhgcd_rec),       /*                             */
      dfhgcd);                         /*                             */
    if(dfhgcd_rec_anch.keypoint ==     /* Check Keypoint flag         */
     KEYPOINT_WARM)                    /* ...if WARM then...          */
    {                                  /*                             */
      rc_warm = RC_OK;                 /* ...set return code          */
    }                                  /*                             */
  }                                    /*                             */
}                                      /*                             */
else                                   /* Unable to open file         */
{                                      /*                             */
```

```
  rc_warm = RC_FAILURE;                 /* Set return code             */
}                                       /*                             */
return rc_warm;                         /* <== Exit point              */
}                                       /*                             */
/*-------------------------------------------------------------------------*
 * function: error_exit                                                    *
 *-------------------------------------------------------------------------*/
void error_exit(int sig_num)            /* ==> Function Entry          */
{                                       /*                             */
/*-------------------------------------------------------------------------*
 * local variables                                                         *
 *-------------------------------------------------------------------------*/
/*-------------------------------------------------------------------------*
 * code                                                                    *
 *-------------------------------------------------------------------------*/
exit(EXIT_FAILURE);                     /* <== Exit point              */
}                                       /* <== Function Exit           */
/*-------------------------------------------------------------------------*
 * End of Program                                                          *
 *-------------------------------------------------------------------------*/
```

J Lemmon
CICS Consultant (UK)

CICS log stream guidance

BACKGROUND

CICS log stream definition is a complicated issue, and guidance on certain aspects of it is much appreciated by customers migrating to CICS Transaction Server from CICS/ESA 4.1.0 or earlier releases. This article gives hints and tips on some of the more common sources of confusion in this area, and offers guidance on where to obtain further information.

AUTODELETE AND RETPD

OS/390 Version 1 Release 3 introduced new optional parameters specifiable when defining a log stream to the MVS System Logger – RETPD and AUTODELETE. RETPD defines a retention period (in days) for log data before it is physically deletable from a log stream.

AUTODELETE (YES or NO) specifies whether the MVS System Logger will perform an automatic physical deletion of log data, or whether an explicit IXGDELET macro call is required to perform the delete. The default values are AUTODELETE(NO) and RETPD(0).

Always specify AUTODELETE(NO) and RETPD(0) for the CICS system log streams DFHLOG and DFHSHUNT. This is important for two good reasons – data integrity and system performance. CICS manages its own system log data deletion, by 'trimming the tail' of the DFHLOG and DFHSHUNT system log streams during keypoint operations when appropriate to do so. This ensures CICS deletes only log data when there is no further need to retain the information for recovery purposes. If AUTODELETE(YES) is specified, the MVS System Logger could physically delete system log data that was still required to guarantee data integrity for a task abend or system restart. Also, if a non-zero RETPD value were specified for the system log streams, the MVS System Logger would not physically delete log data until it has reached this age. Hence redundant data marked for deletion by CICS tail-trimming would not be physically deleted by the MVS System Logger at the next opportunity (a subsequent offload process occurring), resulting in unnecessary I/O activity to retain it.

HIGHOFFLOAD AND LOWOFFLOAD VALUES

HIGHOFFLOAD and LOWOFFLOAD are specified (as percentage values) when a log stream is defined. They represent the boundary limits, relating to the space used by the log stream, that control when MVS System Logger 'offload processing' is to occur. Offload processing is used to optimize the use of log stream primary storage (that is, space within a structure on the Coupling Facility, or within staging datasets for DASD-only logging). One trigger for the start of offload processing is when the primary storage utilization for a log stream reaches its HIGHOFFLOAD value. When this happens, the MVS System Logger will physically delete log data that has already been logically deleted by 'tail-trimming', which was carried out automatically by CICS keypoint processing. This operation will free up space within the log stream's primary storage. Having completed this work, the percentage of remaining log data may now be less than the LOWOFFLOAD value. If it is not, the MVS System Logger will

move log data from the oldest part of the log stream out to secondary storage (offload datasets) until the log stream's primary storage space usage is reduced to the LOWOFFLOAD value.

The values specified for HIGHOFFLOAD and LOWOFFLOAD can have a considerable impact on the performance of a log stream. If HIGHOFFLOAD is set too high, there may be insufficient space remaining for the log stream to use while an offload process is occurring. This can result in log stream entry-full or structure-full conditions. Conversely, if LOWOFFLOAD is set too low, physical deletion of log data during an offload process may not result in the LOWOFFLOAD space boundary being reached, and hence I/O operations to offload datasets would be required.

For the CICS system log streams (DFHLOG and DFHSHUNT), realistic empirical values for HIGHOFFLOAD are 80-85% (for both system and user log streams). For LOWOFFLOAD, a value in the range 40-60% should be used for system log streams, and 0% for user log streams. The percentage values can be fine-tuned following analysis of logging activity (eg from MVS System Logger SMF 88 records).

User journals are not read by CICS applications (they are processed in batch), and there is no advantage in retaining their log data in primary storage once it has been written. A 0% value for LOWOFFLOAD allows offload processing to free up all space within user log streams.

FURTHER SOURCES OF INFORMATION

Run the DFHLSCU batch utility against CICS/ESA 4.1.0 journal datasets to obtain an initial set of values for specification on the equivalent log streams for CICS Transaction Server to use. In addition, the CICS Transaction Server Installation Guide provides details of the manual calculations required when defining log streams. Having defined and used them, refer to the MVS System Logger SMF 88 records – these detail log stream usage and activity, and provide a useful tool for analysing and tuning log stream definitions. On-line information about log streams can be obtained via the MVS D LOGGER command. The CICS joblog and CSMT transient data

destination contains pertinent messages relating to log stream usage and management, such as DFHLG0743 messages when a keypoint has been able to trim the DFHLOG or DFHSHUNT log streams. Also, the IXCMIAPU utility and DFH0STAT COBOL sample program output may be used to review log stream attributes and associated CICS statistics, respectively.

Readers wishing to discuss this article further may e-mail me at andy_wright@uk.ibm.com. CICS is a registered trademark of International Business Machines Corporation.

Andy Wright
CICS Change Team Programmer
IBM (UK)

Wiring CICS regions for desktop sound, e-mail, and alpha paging via TCP/IP – part 2

This month we continue the examples of how a mainframe CICS region could play specific sounds from a desktop computer sound card, send e-mails to an SMTP server on the intranet or the Internet, or send an alpha page to a paging application hooked to an SMTP server.

```
PROCEDURE DIVISION.
MAINLINE.
    PERFORM HANDLE-ABEND
    PERFORM HOUSE-KEEPING
    PERFORM SEND-ALL-EPAGES
    GO TO CLEAR-EM-OUT.
HANDLE-ABEND.
    EXEC CICS HANDLE ABEND LABEL(ABEND-ERR) END-EXEC
    CONTINUE.
HOUSE-KEEPING.
    EXEC CICS ASKTIME
        ABSTIME(ABSTIME)
        END-EXEC
    EXEC CICS FORMATTIME
        ABSTIME(ABSTIME)
        YYMMDD(ERR-DATE) DATESEP
        TIME(ERR-TIME) TIMESEP
        END-EXEC
```

```
    EXEC CICS ASSIGN SYSID(SYSTEM-ID) END-EXEC
    MOVE SYSTEM-ID TO OPR-REGION-ID
    MOVE 'TCPIPØØD' TO ERR-PROGRAM-ID
                       OPR-PROGRAM-ID
    MOVE EIBTRNID TO ERR-TRANSACTION
    MOVE SPACES TO BUFFER-IN BUFFER-OUT
    CONTINUE.
SEND-ALL-EPAGES.
    MOVE SPACES TO HOLD-EML-RECORD
    MOVE 'B' TO EML-RECTYPE
    MOVE COMM-PAGE-NUM TO EML-COMPONENT
    MOVE 'ØØ' TO EML-RECORD
    EXEC CICS STARTBR FILE('TCPIPEML')
        RIDFLD(EML-KEY)
        RESP(FILE-RESPONSE)
        END-EXEC
    IF FILE-RESPONSE NOT = DFHRESP(NORMAL)
        MOVE 'STARTBR ' TO ERR-CALL-TYPE
        PERFORM EVALUATE-STARTBROWSE-RESPONSE
        GO TO CLEAR-EM-OUT
    END-IF
    PERFORM UNTIL EML-RECTYPE NOT = 'B'
        OR EML-COMPONENT NOT = COMM-PAGE-NUM
        OR END-FILE
        EXEC CICS READNEXT INTO(HOLD-EML-RECORD)
           LENGTH(EML-REC-LENGTH)
           KEYLENGTH(EML-KEY-LENGTH)
           FILE('TCPIPEML')
           RIDFLD(EML-KEY)
           RESP(FILE-RESPONSE)
           END-EXEC
        EVALUATE FILE-RESPONSE
            WHEN DFHRESP(NORMAL)
                IF EML-COMPONENT = COMM-PAGE-NUM
                    IF EML-FLAG = 'Y'
                        PERFORM MOVE-EML-PAGER-DATA
                        PERFORM GET-SOCKET
                        PERFORM CONNECT-SOCKET
                        PERFORM RECEIVE-SOCKET
                        PERFORM SEND-MESSAGE
                        PERFORM CLOSE-SOCKET
                    END-IF
                END-IF
            WHEN DFHRESP(ENDFILE)
                MOVE 'END' TO END-FILE-SW
            WHEN OTHER
                MOVE 'READNEXT' TO ERR-CALL-TYPE
                PERFORM EVALUATE-READNEXT-RESPONSE
                GO TO CLEAR-EM-OUT
        END-EVALUATE
```

```
    END-PERFORM
    EXEC CICS ENDBR FILE('TCPIPEML')
        RESP(FILE-RESPONSE)
        END-EXEC
    IF FILE-RESPONSE NOT = DFHRESP(NORMAL)
        MOVE 'ENDBR   ' TO ERR-CALL-TYPE
        PERFORM EVALUATE-ENDBROWSE-RESPONSE
        GO TO CLEAR-EM-OUT
    END-IF
    CONTINUE.
MOVE-EML-PAGER-DATA.
    MOVE SPACES TO PAGE-TO
                   PAGE-SUBJECT
    PERFORM VARYING PAGER-SS FROM 1 BY 1
        UNTIL EML-PAGER-NUM-ENTRY(PAGER-SS) = ' '
        OR PAGER-SS > 8
            MOVE EML-PAGER-NUM-ENTRY(PAGER-SS)
                TO PAGE-TO-ENTRY(PAGER-SS)
                   PAGE-SUBJECT-ENTRY(PAGER-SS)
    END-PERFORM
    CONTINUE.
SEND-MESSAGE.
    IF SMTP-RETURN-CODE = '22Ø'
        MOVE SPACES TO BUFFER-OUT
        MOVE LENGTH OF MAIL-HEADER-LINEØ1 TO NBYTE
        MOVE MAIL-HEADER-LINEØ1 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
    ELSE
        GO TO SMTP-ERROR
    END-IF
    IF SMTP-RETURN-CODE = '25Ø'
        MOVE SPACES TO BUFFER-OUT
        MOVE LENGTH OF MAIL-HEADER-LINEØ2 TO NBYTE
        MOVE MAIL-HEADER-LINEØ2 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
    ELSE
        GO TO SMTP-ERROR
    END-IF
    IF SMTP-RETURN-CODE = '25Ø'
        MOVE SPACES TO BUFFER-OUT
        MOVE LENGTH OF MAIL-HEADER-LINEØ3 TO NBYTE
        MOVE MAIL-HEADER-LINEØ3 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
    ELSE
        GO TO SMTP-ERROR
    END-IF
    IF SMTP-RETURN-CODE = '25Ø'
```

```
           MOVE SPACES TO BUFFER-OUT
           MOVE LENGTH OF MAIL-HEADER-LINEØ4 TO NBYTE
           MOVE MAIL-HEADER-LINEØ4 TO BUFFER-OUT
           PERFORM SEND-SOCKET
           PERFORM RECEIVE-SOCKET
       ELSE
           GO TO SMTP-ERROR
       END-IF
       IF SMTP-RETURN-CODE = '354'
           MOVE SPACES TO BUFFER-OUT
           MOVE LENGTH OF MAIL-BODY-LINEØ1 TO NBYTE
           MOVE MAIL-BODY-LINEØ1 TO BUFFER-OUT
           PERFORM SEND-SOCKET
           MOVE SPACES TO BUFFER-OUT
           MOVE LENGTH OF MAIL-BODY-LINEØ2 TO NBYTE
           MOVE MAIL-BODY-LINEØ2 TO BUFFER-OUT
           PERFORM SEND-SOCKET
           MOVE SPACES TO BUFFER-OUT
           MOVE LENGTH OF MAIL-BODY-LINEØ3 TO NBYTE
           MOVE MAIL-BODY-LINEØ3 TO BUFFER-OUT
           PERFORM SEND-SOCKET
           MOVE SPACES TO BUFFER-OUT
           MOVE LENGTH OF MAIL-BODY-LINEØ4 TO NBYTE
           MOVE COMM-MESSAGE TO MAIL-BODY-LINEØ4
           MOVE MAIL-BODY-LINEØ4 TO BUFFER-OUT
           PERFORM SEND-SOCKET
           MOVE SPACES TO BUFFER-OUT
           MOVE LENGTH OF MAIL-BODY-END TO NBYTE
           MOVE MAIL-BODY-END TO BUFFER-OUT
           PERFORM SEND-SOCKET
           PERFORM RECEIVE-SOCKET
       ELSE
           GO TO SMTP-ERROR
       END-IF
       IF SMTP-RETURN-CODE = '25Ø'
           MOVE SPACES TO BUFFER-OUT
           MOVE LENGTH OF MAIL-MESSAGE-END TO NBYTE
           MOVE MAIL-MESSAGE-END TO BUFFER-OUT
           PERFORM SEND-SOCKET
           PERFORM RECEIVE-SOCKET
       ELSE
           GO TO SMTP-ERROR
       END-IF
       CONTINUE.
   GET-SOCKET.
       MOVE 'SOCKET          ' TO SOC-FUNCTION
       CALL 'EZASOKET' USING SOC-FUNCTION
            AF
            SOCTYPE
            PROTO
```

```
           ERRNO
           RETCODE
      IF RETCODE < Ø
          MOVE 'TCPIP   ' TO ERR-CALL-TYPE
          MOVE 'GET SOCKET' TO ERR-TYPE
          MOVE ERRNO TO ERR-NUMBER
          MOVE RETCODE TO ERR-RETURN-CODE
          MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
          PERFORM WRITEQ-TRANSIENT-DATA
          GO TO CLEAR-EM-OUT
      ELSE
          MOVE RETCODE TO SOCKET-DESC
      END-IF
      CONTINUE.
  CONNECT-SOCKET.
      MOVE COMM-IP-ADDRESS TO IP-ADDRESS
      MOVE 'CONNECT         '   TO SOC-FUNCTION
      CALL 'EZASOKET' USING SOC-FUNCTION
           SOCKET-DESC
           NAME
           ERRNO
           RETCODE
      IF RETCODE = Ø
          CONTINUE
      ELSE
          MOVE 'TCPIP   ' TO ERR-CALL-TYPE
          MOVE 'CONNECT   ' TO ERR-TYPE
          MOVE ERRNO TO ERR-NUMBER
          MOVE RETCODE TO ERR-RETURN-CODE
          MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
          PERFORM WRITEQ-TRANSIENT-DATA
          PERFORM CLOSE-SOCKET
          GO TO CLEAR-EM-OUT
      END-IF
      CONTINUE.
  RECEIVE-SOCKET.
      MOVE SPACES TO BUFFER-IN
      MOVE LENGTH OF BUFFER-IN TO NBYTE
      MOVE 'RECV            ' TO SOC-FUNCTION
      CALL 'EZASOKET' USING SOC-FUNCTION
           SOCKET-DESC
           FLAGS
           NBYTE
           BUFFER-IN
           ERRNO
           RETCODE
      IF RETCODE <  Ø
          MOVE 'TCPIP   ' TO ERR-CALL-TYPE
          MOVE 'RECEIVE   ' TO ERR-TYPE
          MOVE ERRNO TO ERR-NUMBER
```

```
        MOVE RETCODE TO ERR-RETURN-CODE
        MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
        PERFORM WRITEQ-TRANSIENT-DATA
        PERFORM CLOSE-SOCKET
        GO TO CLEAR-EM-OUT
    ELSE
        MOVE RETCODE TO NBYTE
        CALL 'EZACICØ5' USING BUFFER-IN NBYTE
    END-IF
    CONTINUE.
SEND-SOCKET.
    CALL 'EZACICØ4' USING BUFFER-OUT
         NBYTE
    MOVE 'SEND            ' TO SOC-FUNCTION
    CALL 'EZASOKET' USING SOC-FUNCTION
         SOCKET-DESC
         FLAGS
         NBYTE
         BUFFER-OUT
         ERRNO
         RETCODE
    IF RETCODE <  Ø
        MOVE 'TCPIP   ' TO ERR-CALL-TYPE
        MOVE 'SEND      ' TO ERR-TYPE
        MOVE ERRNO TO ERR-NUMBER
        MOVE RETCODE TO ERR-RETURN-CODE
        MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
        PERFORM WRITEQ-TRANSIENT-DATA
        PERFORM CLOSE-SOCKET
        GO TO CLEAR-EM-OUT
    ELSE
        CONTINUE
    END-IF
    CONTINUE.
CLOSE-SOCKET.
    MOVE ZEROES TO RETCODE ERRNO
    MOVE 'CLOSE           ' TO SOC-FUNCTION
    CALL 'EZASOKET' USING SOC-FUNCTION
        SOCKET-DESC
        ERRNO
        RETCODE
    IF RETCODE <  Ø
        MOVE 'TCPIP   ' TO ERR-CALL-TYPE
        MOVE 'CLOSESOKET' TO ERR-TYPE
        MOVE ERRNO TO ERR-NUMBER
        MOVE RETCODE TO ERR-RETURN-CODE
        MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
        PERFORM WRITEQ-TRANSIENT-DATA
        GO TO CLEAR-EM-OUT
    ELSE
```

```
            CONTINUE
        END-IF
        CONTINUE.
    SMTP-ERROR.
        MOVE 'SMTP    ' TO ERR-CALL-TYPE
        MOVE 'MESSAGE   ' TO ERR-TYPE
        MOVE SMTP-RETURN-CODE TO ERR-NUMBER
        MOVE ZEROES TO ERR-RETURN-CODE
        MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
        PERFORM WRITEQ-TRANSIENT-DATA
        MOVE SPACES TO BUFFER-OUT
        MOVE LENGTH OF MAIL-MESSAGE-RSET TO NBYTE
        MOVE MAIL-MESSAGE-RSET TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
        MOVE SPACES TO BUFFER-OUT
        MOVE LENGTH OF MAIL-MESSAGE-END TO NBYTE
        MOVE MAIL-MESSAGE-END TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
        PERFORM CLOSE-SOCKET
        GO TO CLEAR-EM-OUT.
    EVALUATE-STARTBROWSE-RESPONSE.
        EVALUATE FILE-RESPONSE
            WHEN DFHRESP(DISABLED)
                MOVE 'DISABLED  ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(FILENOTFOUND)
                MOVE 'FILENOTFND' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(ILLOGIC)
                MOVE 'ILLOGIC   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(INVREQ  )
                MOVE 'INVREQ    ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(IOERR   )
                MOVE 'IOERR     ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(ISCINVREQ)
                MOVE 'ISCINVREQ ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(NOTAUTH)
                MOVE 'NOTAUTH   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(NOTFND)
                MOVE 'NOTFND    ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(NOTOPEN)
                MOVE 'NOTOPEN   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
```

```
    END-EVALUATE.
    PERFORM WRITEQ-TRANSIENT-DATA
    CONTINUE.
EVALUATE-READNEXT-RESPONSE.
    EVALUATE FILE-RESPONSE
        WHEN DFHRESP(FILENOTFOUND)
            MOVE 'FILENOTFND' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(ILLOGIC)
            MOVE 'ILLOGIC   ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(INVREQ  )
            MOVE 'INVREQ    ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(IOERR   )
            MOVE 'IOERR     ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(ISCINVREQ)
            MOVE 'ISCINVREQ ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(LENGERR)
            MOVE 'LENGERR   ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(NOTAUTH)
            MOVE 'NOTAUTH   ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(NOTFND)
            MOVE 'NOTFND    ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
    END-EVALUATE.
    PERFORM WRITEQ-TRANSIENT-DATA
    CONTINUE.
EVALUATE-ENDBROWSE-RESPONSE.
    EVALUATE FILE-RESPONSE
        WHEN DFHRESP(FILENOTFOUND)
            MOVE 'FILENOTFND' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(ILLOGIC)
            MOVE 'ILLOGIC   ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(INVREQ  )
            MOVE 'INVREQ    ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(IOERR   )
            MOVE 'IOERR     ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(ISCINVREQ)
            MOVE 'ISCINVREQ ' TO ERR-TYPE
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        WHEN DFHRESP(NOTAUTH)
            MOVE 'NOTAUTH   ' TO ERR-TYPE
```

```
            MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
    END-EVALUATE.
    PERFORM WRITEQ-TRANSIENT-DATA
    CONTINUE.
WRITEQ-TRANSIENT-DATA.
    EXEC CICS WRITEQ TD QUEUE('CSMT')
        FROM(ERROR-MESSAGE)
        LENGTH(LENGTH OF ERROR-MESSAGE)
        RESP(TRANSDATA-RESPONSE)
        END-EXEC
    IF TRANSDATA-RESPONSE = DFHRESP(NORMAL)
        CONTINUE
    ELSE
        MOVE 'WRITE TD' TO OPR-CALL-TYPE
        MOVE 'SEVERE ERR' TO OPR-TYPE
        MOVE 'TRANSIENT DATA WRITE FAILED' TO OPR-TEXT
        EXEC CICS WRITE OPERATOR
            TEXT(OPERATOR-MESSAGE)
            TEXTLENGTH(LENGTH OF OPERATOR-MESSAGE)
            RESP(OPERATOR-RESPONSE)
            END-EXEC
    END-IF
    CONTINUE.
ABEND-ERR.
    EXEC CICS ASSIGN ABCODE(ERR-TYPE) END-EXEC
    MOVE 'MAIL TRANSMISSION HAS ABENDED   ' TO ERR-TEXT
    PERFORM WRITEQ-TRANSIENT-DATA
    GO TO CLEAR-EM-OUT.
CLEAR-EM-OUT.
    EXEC CICS RETURN END-EXEC
    STOP RUN.
```

TCPIP00E

This program uses the same basic SOCKETS logic to connect to an SMTP server on the LAN at TCP/IP port 25. This time however, the RCPT TO field is actually a single person on the e-mail system or a list of people on the e-mail system. The short message is copied from COMM-IP-PAGE into the body of the e-mail message and forwarded to the server for placement in the person's in-box. There are some lines of code that are hardcoded to supply the time zone offset value for the SMTP server. This has to change if your location in the world uses daylight-saving time during the summer season. Note that paragraph SEND-MESSAGE processes the responses from the SMTP server and appropriately continues or notifies the message log of the error code.

```
 ID DIVISION.
 PROGRAM-ID. TCPIPØØE.
 DATE-WRITTEN. DEC 1999.
*****************************************************************
* CICS TCP/IP SOCKET INTERFACE PROGRAM--EMAIL TO SMTP PORT 25
*****************************************************************
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77  ABSTIME                     PIC S9(15) COMP-3.
 77  EML-REC-LENGTH              PIC S9(4) COMP VALUE +8Ø.
 77  EML-KEY-LENGTH              PIC S9(4) COMP VALUE +12.
 77  TRANSDATA-RESPONSE          PIC S9(8) COMP.
 77  OPERATOR-RESPONSE           PIC S9(8) COMP.
 77  FILE-RESPONSE               PIC S9(8) COMP.
 77  ABEND-CODE                  PIC X(Ø4).
 77  SYSTEM-ID                   PIC X(Ø4).
 77  ADDRESSEE-SS                PIC 9(Ø2).
 Ø1  END-FILE-SW                  PIC X(3).
   88 END-FILE VALUE 'END'.
 Ø1  MDATE.
     Ø2  MMON                     PIC 9(Ø2).
     Ø2  FILLER                   PIC X(Ø1).
     Ø2  MDAY                     PIC 9(Ø2).
     Ø2  FILLER                   PIC X(Ø1).
     Ø2  MYR                      PIC 9(Ø2).
 Ø1  MTIME.
     Ø2  MHR                      PIC 9(Ø2).
     Ø2  FILLER                   PIC X(Ø1).
     Ø2  MMIN                     PIC 9(Ø2).
     Ø2  FILLER                   PIC X(Ø1).
     Ø2  MSEC                     PIC 9(Ø2).
 Ø1  DAYLIGHT-HOLD.
     Ø2  DAYLIGHT-YR              PIC 9(Ø2).
     Ø2  DAYLIGHT-MON             PIC 9(Ø2).
     Ø2  DAYLIGHT-DAY             PIC 9(Ø2).
     Ø2  DAYLIGHT-HR              PIC 9(Ø2).
     Ø2  DAYLIGHT-MIN             PIC 9(Ø2).
     Ø2  DAYLIGHT-SEC             PIC 9(Ø2).
 Ø1  DAYLIGHT-TIME REDEFINES DAYLIGHT-HOLD PIC 9(12).
 Ø1  FILLER                   PIC X(13) VALUE 'TCPIP PARMS >'.
 Ø1  SOCKET-GROUP.
     Ø3  SOC-FUNCTION             PIC X(16) VALUE SPACES.
     Ø3  ERRNO                    PIC 9(8) COMP VALUE ZEROES.
     Ø3  RETCODE                  PIC S9(8) COMP VALUE ZEROES.
     Ø3  AF                       PIC 9(8) COMP VALUE 2.
     Ø3  SOCTYPE                  PIC 9(8) COMP VALUE 1.
     Ø3  PROTO                    PIC 9(8) COMP VALUE Ø.
     Ø3  NAME.
         Ø5  FAMILY               PIC 9(4) COMP VALUE 2.
```

```
          Ø5  PORT                 PIC 9(4) COMP VALUE 25.
          Ø5  IP-ADDRESS           PIC 9(8) COMP.
          Ø5  RESERVED             PIC X(8) VALUE LOW-VALUES.
      Ø3  FLAGS                    PIC 9(8) COMP VALUE Ø.
      Ø3  SOCKET-DESC              PIC 9(4) COMP.
      Ø3  NBYTE                    PIC 9(8) COMP.
 Ø1   FILLER                   PIC X(1Ø) VALUE 'BUFFER-IN>'.
 Ø1   BUFFER-IN.
      Ø2  SMTP-DATA.
          Ø4  SMTP-RETURN-CODE     PIC 9(Ø3).
          Ø4  SMTP-RETURN-MESSAGE  PIC X(253).
 Ø1   FILLER                   PIC X(11) VALUE 'BUFFER-OUT>'.
 Ø1   BUFFER-OUT.
      Ø2  FILLER                   PIC X(8Ø).
 Ø1   MAIL-MESSAGE-HEADER.
      Ø2  MAIL-HEADER-LINEØ1.
          Ø4  FILLER   PIC X(39)
          VALUE 'HELO MYMAINFRAME                      '.
          Ø4  FILLER   PIC X(39)
          VALUE '                                      '.
          Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
      Ø2  MAIL-HEADER-LINEØ2.
          Ø4  FILLER   PIC X(39)
          VALUE 'MAIL FROM:<CICS-SYSTEMS@MYMAINFRAME>  '.
          Ø4  FILLER   PIC X(39)
          VALUE '                                      '.
          Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
      Ø2  MAIL-HEADER-LINEØ3.
          Ø4  FILLER   PIC X(Ø8) VALUE 'RCPT TO:'.
          Ø4  MAIL-RCPT.
              Ø6  MAIL-RCPT-ENTRY OCCURS 65 TIMES PIC X(Ø1).
          Ø4  FILLER   PIC X(Ø5) VALUE SPACES.
          Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
      Ø2  MAIL-HEADER-LINEØ4.
          Ø4  FILLER   PIC X(39)
          VALUE 'DATA                                  '.
          Ø4  FILLER   PIC X(39)
          VALUE '                                      '.
          Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
 Ø1   MAIL-MESSAGE-BODY.
      Ø2  MAIL-BODY-LINEØ1.
*             'DATE: DD MMM YY HH:MM:SS -Ø5ØØ'.
          Ø4  FILLER   PIC X(Ø6) VALUE 'DATE: '.
          Ø4  MAIL-DAY PIC X(Ø2).
          Ø4  FILLER   PIC X(Ø1) VALUE SPACE.
          Ø4  MAIL-MON PIC X(Ø3).
          Ø4  FILLER   PIC X(Ø1) VALUE SPACE.
          Ø4  MAIL-YR  PIC X(Ø4).
          Ø4  FILLER   PIC X(Ø1) VALUE SPACE.
          Ø4  MAIL-HR  PIC X(Ø2).
```

```
           Ø4  FILLER   PIC X(Ø1) VALUE ':'.
           Ø4  MAIL-MIN PIC X(Ø2).
           Ø4  FILLER   PIC X(Ø1) VALUE ':'.
           Ø4  MAIL-SEC PIC X(Ø2).
           Ø4  FILLER   PIC X(Ø2) VALUE ' -'.
           Ø4  MAIL-OFS PIC X(Ø4) VALUE 'Ø6ØØ'.
           Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
           Ø4  FILLER   PIC X(46).
       Ø2  MAIL-BODY-LINEØ2.
           Ø4  FILLER   PIC X(39)
           VALUE 'FROM: XYZ application                 '.
           Ø4  FILLER   PIC X(39)
           VALUE '                                      '.
           Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
       Ø2  MAIL-BODY-LINEØ3.
           Ø4  FILLER   PIC X(Ø4) VALUE 'TO: '.
           Ø4  MAIL-TO.
               Ø6  MAIL-TO-ENTRY OCCURS 65 TIMES PIC X(Ø1).
           Ø4  FILLER   PIC X(Ø9) VALUE SPACES.
           Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
       Ø2  MAIL-BODY-LINEØ4.
           Ø4  FILLER   PIC X(39)
           VALUE 'SUBJECT:SAMPLE TCPIP EMAIL MESSAGES    '.
           Ø4  FILLER   PIC X(39)
           VALUE '                                      '.
           Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
       Ø2  MAIL-BODY-LINEØ5.
           Ø4  FILLER   PIC X(76) VALUE SPACES.
           Ø4  FILLER   PIC X(Ø4) VALUE X'ØD15ØD15'.
       Ø2  MAIL-BODY-LINEØ6.
           Ø4  FILLER   PIC X(39)
           VALUE 'the following message is for demonstrat'.
           Ø4  FILLER   PIC X(39)
           VALUE 'ion purposes only, use at your own risk.'.
           Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
       Ø2  MAIL-BODY-LINEØ7.
           Ø4  FILLER   PIC X(39)
           VALUE '                                      '.
           Ø4  FILLER   PIC X(39)
           VALUE '                                      '.
           Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
       Ø2  MAIL-BODY-LINEØ8.
           Ø4  MAIL-PAGE PIC X(3Ø).
           Ø4  FILLER    PIC X(48)
           VALUE ', refer to your own staff for more details.     '.
           Ø4  FILLER   PIC X(Ø2) VALUE X'ØD15'.
    Ø1  MAIL-MESSAGE-RSET.
       Ø2  FILLER   PIC X(Ø5)
       VALUE 'RSET '.
       Ø2  FILLER   PIC X(Ø2) VALUE X'ØD15'.
```

```
Ø1  MAIL-BODY-END.
    Ø2  FILLER   PIC X(Ø5) VALUE X'ØD154BØD15'.
    Ø2  FILLER   PIC X(75) VALUE SPACES.
Ø1  MAIL-MESSAGE-END.
    Ø2  FILLER   PIC X(39)
    VALUE 'QUIT                                    '.
    Ø2  FILLER   PIC X(39)
    VALUE '                                        '.
    Ø2  FILLER   PIC X(Ø2) VALUE X'ØD15'.
Ø1  HOLD-EML-RECORD.
    Ø2  EML-KEY.
        Ø4  EML-RECTYPE       PIC X(1).
        Ø4  FILLER            PIC X(1).
        Ø4  EML-COMPONENT     PIC 9(6).
        Ø4  FILLER            PIC X(1).
        Ø4  EML-RECORD        PIC X(3).
    Ø2  FILLER                PIC X(1).
    Ø2  EML-FLAG              PIC X(1).
    Ø2  FILLER                PIC X(1).
    Ø2  EML-DATA              PIC X(65).
    Ø2  EML-ADDRESSEE REDEFINES EML-DATA.
        Ø4  EML-ADDRESSEE-ENTRY OCCURS 65 TIMES PIC X(1).
    Ø2  EML-PAGER-DATA REDEFINES EML-DATA.
        Ø4  EML-PAGER-NUM.
            Ø6  EML-PAGER-NUM-ENTRY OCCURS 8 TIMES PIC X(1).
        Ø4  EML-DESCRIPTION   PIC X(57).
Ø1  ERROR-MESSAGE.
    Ø2  ERR-PROGRAM-ID            PIC X(Ø8).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  ERR-DATE                  PIC X(Ø8).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  ERR-TIME                  PIC X(Ø8).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  ERR-TRANSACTION           PIC X(Ø4).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  ERR-TYPE                  PIC X(1Ø).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  ERR-CALL-TYPE             PIC X(Ø8).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  ERR-TEXT                  PIC X(4Ø).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  ERR-NUMBER                PIC 9(8).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  ERR-RETURN-CODE           PIC 9(8).
Ø1  OPERATOR-MESSAGE.
    Ø2  OPR-REGION-ID             PIC X(Ø4).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  OPR-PROGRAM-ID            PIC X(Ø8).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  OPR-TYPE                  PIC X(1Ø).
```

```
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  OPR-CALL-TYPE             PIC X(Ø8).
    Ø2  FILLER                    PIC X(Ø1) VALUE SPACE.
    Ø2  OPR-TEXT                  PIC X(28).
LINKAGE SECTION.
Ø1  DFHCOMMAREA.
    Ø2  COMM-PAGE-NUM             PIC 9(6).
    Ø2  COMM-PAGE                 PIC X(3Ø).
    Ø2  COMM-IP-ADDRESS           PIC 9(Ø8) COMP.
PROCEDURE DIVISION.
MAINLINE.
    PERFORM HANDLE-ABEND
    PERFORM HOUSE-KEEPING
    PERFORM SEND-ALL-EMAILS
    GO TO CLEAR-EM-OUT.
HANDLE-ABEND.
    EXEC CICS HANDLE ABEND LABEL(ABEND-ERR) END-EXEC
    CONTINUE.
HOUSE-KEEPING.
    EXEC CICS ASKTIME
        ABSTIME(ABSTIME)
        END-EXEC
    EXEC CICS FORMATTIME
        ABSTIME(ABSTIME)
        YYMMDD(ERR-DATE) DATESEP
        TIME(ERR-TIME) TIMESEP
        END-EXEC
    EXEC CICS FORMATTIME
        ABSTIME(ABSTIME)
        mmddYY(MDATE) DATESEP
        TIME(MTIME) TIMESEP
        END-EXEC
    MOVE MHR TO MAIL-HR
    MOVE MMIN TO MAIL-MIN
    MOVE MSEC TO MAIL-SEC
    EVALUATE MMON
        WHEN Ø1 MOVE 'JAN' TO MAIL-MON
        WHEN Ø2 MOVE 'FEB' TO MAIL-MON
        WHEN Ø3 MOVE 'MAR' TO MAIL-MON
        WHEN Ø4 MOVE 'APR' TO MAIL-MON
        WHEN Ø5 MOVE 'MAY' TO MAIL-MON
        WHEN Ø6 MOVE 'JUN' TO MAIL-MON
        WHEN Ø7 MOVE 'JUL' TO MAIL-MON
        WHEN Ø8 MOVE 'AUG' TO MAIL-MON
        WHEN Ø9 MOVE 'SEP' TO MAIL-MON
        WHEN 1Ø MOVE 'OCT' TO MAIL-MON
        WHEN 11 MOVE 'NOV' TO MAIL-MON
        WHEN 12 MOVE 'DEC' TO MAIL-MON
    END-EVALUATE
    MOVE MDAY TO MAIL-DAY
```

```
    MOVE MYR TO MAIL-YR
    MOVE COMM-PAGE TO MAIL-PAGE
    MOVE MYR TO DAYLIGHT-YR
    MOVE MMON TO DAYLIGHT-MON
    MOVE MDAY TO DAYLIGHT-DAY
    MOVE MHR TO DAYLIGHT-HR
    MOVE MMIN TO DAYLIGHT-MIN
    MOVE MSEC TO DAYLIGHT-SEC
    IF DAYLIGHT-YR = 00
        AND DAYLIGHT-TIME > 000402015959
        AND DAYLIGHT-TIME < 001029020000
            MOVE '0500' TO MAIL-OFS
    END-IF
    IF DAYLIGHT-YR = 01
        AND DAYLIGHT-TIME > 000401015959
        AND DAYLIGHT-TIME < 001028020000
            MOVE '0500' TO MAIL-OFS
    END-IF
    IF DAYLIGHT-YR = 02
        AND DAYLIGHT-TIME > 000407015959
        AND DAYLIGHT-TIME < 001027020000
            MOVE '0500' TO MAIL-OFS
    END-IF
    IF DAYLIGHT-YR = 03
        AND DAYLIGHT-TIME > 000406015959
        AND DAYLIGHT-TIME < 001026020000
            MOVE '0500' TO MAIL-OFS
    END-IF
    IF DAYLIGHT-YR = 04
        AND DAYLIGHT-TIME > 000404015959
        AND DAYLIGHT-TIME < 001031020000
            MOVE '0500' TO MAIL-OFS
    END-IF
    EXEC CICS ASSIGN SYSID(SYSTEM-ID) END-EXEC
    MOVE SYSTEM-ID TO OPR-REGION-ID
    MOVE 'TCPIP00E' TO ERR-PROGRAM-ID
                       OPR-PROGRAM-ID
    MOVE EIBTRNID TO ERR-TRANSACTION
    MOVE SPACES TO BUFFER-IN BUFFER-OUT
    CONTINUE.
SEND-ALL-EMAILS.
    MOVE SPACES TO HOLD-EML-RECORD
    MOVE 'A' TO EML-RECTYPE
    MOVE COMM-PAGE-NUM TO EML-COMPONENT
    MOVE '00' TO EML-RECORD
    EXEC CICS STARTBR FILE('TCPIPEML')
        RIDFLD(EML-KEY)
        RESP(FILE-RESPONSE)
        END-EXEC
    IF FILE-RESPONSE NOT = DFHRESP(NORMAL)
```

```
           MOVE 'STARTBR ' TO ERR-CALL-TYPE
           PERFORM EVALUATE-STARTBROWSE-RESPONSE
           GO TO CLEAR-EM-OUT
       END-IF
       PERFORM UNTIL EML-RECTYPE NOT = 'A'
           OR EML-COMPONENT NOT = COMM-PAGE-NUM
           OR END-FILE
           EXEC CICS READNEXT INTO(HOLD-EML-RECORD)
              LENGTH(EML-REC-LENGTH)
              KEYLENGTH(EML-KEY-LENGTH)
              FILE('TCPIPEML')
              RIDFLD(EML-KEY)
              RESP(FILE-RESPONSE)
              END-EXEC
           EVALUATE FILE-RESPONSE
               WHEN ØØ
                   IF EML-COMPONENT = COMM-PAGE-NUM
                       IF EML-FLAG = 'Y'
                           PERFORM MOVE-RCPT-TO-ADDRESSEE-NAME
                           PERFORM MOVE-MAIL-TO-ADDRESSEE-NAME
                           PERFORM GET-SOCKET
                           PERFORM CONNECT-SOCKET
                           PERFORM RECEIVE-SOCKET
                           PERFORM SEND-MESSAGE
                           PERFORM CLOSE-SOCKET
                       END-IF
                   END-IF
               WHEN DFHRESP(ENDFILE)
                   MOVE 'END' TO END-FILE-SW
               WHEN OTHER
                   MOVE 'READNEXT' TO ERR-CALL-TYPE
                   PERFORM EVALUATE-READNEXT-RESPONSE
                   GO TO CLEAR-EM-OUT
           END-EVALUATE
       END-PERFORM
       EXEC CICS ENDBR FILE('TCPIPEML')
           RESP(FILE-RESPONSE)
           END-EXEC
       IF FILE-RESPONSE NOT = DFHRESP(NORMAL)
           MOVE 'ENDBR   ' TO ERR-CALL-TYPE
           PERFORM EVALUATE-ENDBROWSE-RESPONSE
           GO TO CLEAR-EM-OUT
       END-IF
       CONTINUE.
   MOVE-RCPT-TO-ADDRESSEE-NAME.
       MOVE SPACES TO MAIL-RCPT
       PERFORM VARYING ADDRESSEE-SS FROM 1 BY 1
           UNTIL EML-ADDRESSEE-ENTRY(ADDRESSEE-SS) = ' '
           OR ADDRESSEE-SS > 65
               MOVE EML-ADDRESSEE-ENTRY(ADDRESSEE-SS)
```

```
                TO MAIL-RCPT-ENTRY(ADDRESSEE-SS)
    END-PERFORM
    CONTINUE.
MOVE-MAIL-TO-ADDRESSEE-NAME.
    MOVE SPACES TO MAIL-TO
    PERFORM VARYING ADDRESSEE-SS FROM 2 BY 1
        UNTIL EML-ADDRESSEE-ENTRY(ADDRESSEE-SS) = '@'
        OR ADDRESSEE-SS > 65
            MOVE EML-ADDRESSEE-ENTRY(ADDRESSEE-SS)
                TO MAIL-TO-ENTRY(ADDRESSEE-SS)
    END-PERFORM
    INSPECT MAIL-TO REPLACING ALL '_' BY ' '
    CONTINUE.
SEND-MESSAGE.
    IF SMTP-RETURN-CODE = '22Ø'
        MOVE MAIL-HEADER-LINEØ1 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
    ELSE
        GO TO SMTP-ERROR
    END-IF
    IF SMTP-RETURN-CODE = '25Ø'
        MOVE SPACES TO BUFFER-OUT
        MOVE MAIL-HEADER-LINEØ2 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
    ELSE
        GO TO SMTP-ERROR
    END-IF
    IF SMTP-RETURN-CODE = '25Ø'
        MOVE MAIL-HEADER-LINEØ3 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
    ELSE
        GO TO SMTP-ERROR
    END-IF
    IF SMTP-RETURN-CODE = '25Ø'
        MOVE MAIL-HEADER-LINEØ4 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
    ELSE
        GO TO SMTP-ERROR
    END-IF
    IF SMTP-RETURN-CODE = '354'
        MOVE MAIL-BODY-LINEØ1 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        MOVE MAIL-BODY-LINEØ2 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        MOVE MAIL-BODY-LINEØ3 TO BUFFER-OUT
        PERFORM SEND-SOCKET
```

```
        MOVE MAIL-BODY-LINEØ4 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        MOVE MAIL-BODY-LINEØ5 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        MOVE MAIL-BODY-LINEØ6 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        MOVE MAIL-BODY-LINEØ7 TO BUFFER-OUT
        PERFORM SEND-SOCKET
        MOVE MAIL-BODY-LINEØ8 TO BUFFER-OUT
        EXEC CICS ASKTIME END-EXEC
        PERFORM SEND-SOCKET
        MOVE MAIL-BODY-END TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
    ELSE
        GO TO SMTP-ERROR
    END-IF
    CONTINUE.
GET-SOCKET.
    MOVE 'SOCKET          ' TO SOC-FUNCTION
    CALL 'EZASOKET' USING SOC-FUNCTION
         AF
         SOCTYPE
         PROTO
         ERRNO
         RETCODE
    IF RETCODE < Ø
        MOVE 'TCPIP   ' TO ERR-CALL-TYPE
        MOVE 'GET SOCKET' TO ERR-TYPE
        MOVE ERRNO TO ERR-NUMBER
        MOVE RETCODE TO ERR-RETURN-CODE
        MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        PERFORM WRITEQ-TRANSIENT-DATA
    ELSE
        MOVE RETCODE TO SOCKET-DESC
    END-IF
    CONTINUE.
CONNECT-SOCKET.
    MOVE COMM-IP-ADDRESS TO IP-ADDRESS
    MOVE 'CONNECT         '   TO SOC-FUNCTION
    CALL 'EZASOKET' USING SOC-FUNCTION
         SOCKET-DESC
         NAME
         ERRNO
         RETCODE
    IF RETCODE = Ø
        CONTINUE
    ELSE
        MOVE 'TCPIP   ' TO ERR-CALL-TYPE
        MOVE 'CONNECT   ' TO ERR-TYPE
```

```
        MOVE ERRNO TO ERR-NUMBER
        MOVE RETCODE TO ERR-RETURN-CODE
        MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
        PERFORM WRITEQ-TRANSIENT-DATA
        PERFORM CLOSE-SOCKET
        GO TO CLEAR-EM-OUT
    END-IF
    CONTINUE.
RECEIVE-SOCKET.
    MOVE SPACES TO BUFFER-IN
    MOVE LENGTH OF BUFFER-IN TO NBYTE
    MOVE 'RECV            ' TO SOC-FUNCTION
    CALL 'EZASOKET' USING SOC-FUNCTION
         SOCKET-DESC
         FLAGS
         NBYTE
         BUFFER-IN
         ERRNO
         RETCODE
    IF RETCODE <  Ø
        MOVE 'TCPIP   ' TO ERR-CALL-TYPE
        MOVE 'RECEIVE   ' TO ERR-TYPE
        MOVE ERRNO TO ERR-NUMBER
        MOVE RETCODE TO ERR-RETURN-CODE
        MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
        PERFORM WRITEQ-TRANSIENT-DATA
        PERFORM CLOSE-SOCKET
        GO TO CLEAR-EM-OUT
    ELSE
        MOVE RETCODE TO NBYTE
        CALL 'EZACICØ5' USING BUFFER-IN NBYTE
    END-IF
    CONTINUE.
SEND-SOCKET.
    MOVE LENGTH OF BUFFER-OUT TO NBYTE
    CALL 'EZACICØ4' USING BUFFER-OUT
         NBYTE
    MOVE LENGTH OF BUFFER-OUT TO NBYTE
    MOVE 'SEND            ' TO SOC-FUNCTION
    CALL 'EZASOKET' USING SOC-FUNCTION
         SOCKET-DESC
         FLAGS
         NBYTE
         BUFFER-OUT
         ERRNO
         RETCODE
    IF RETCODE <  Ø
        MOVE 'TCPIP   ' TO ERR-CALL-TYPE
        MOVE 'SEND      ' TO ERR-TYPE
        MOVE ERRNO TO ERR-NUMBER
```

```
            MOVE RETCODE TO ERR-RETURN-CODE
            MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
            PERFORM WRITEQ-TRANSIENT-DATA
            PERFORM CLOSE-SOCKET
            GO TO CLEAR-EM-OUT
        ELSE
            CONTINUE
        END-IF
        CONTINUE.
    CLOSE-SOCKET.
        MOVE ZEROES TO RETCODE ERRNO
        MOVE 'CLOSE           ' TO SOC-FUNCTION
        CALL 'EZASOKET' USING SOC-FUNCTION
            SOCKET-DESC
            ERRNO
            RETCODE
        IF RETCODE <  Ø
            MOVE 'TCPIP   ' TO ERR-CALL-TYPE
            MOVE 'CLOSESOKET' TO ERR-TYPE
            MOVE ERRNO TO ERR-NUMBER
            MOVE RETCODE TO ERR-RETURN-CODE
            MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
            PERFORM WRITEQ-TRANSIENT-DATA
            GO TO CLEAR-EM-OUT
        ELSE
            CONTINUE
        END-IF
        CONTINUE.
    SMTP-ERROR.
        MOVE 'SMTP    ' TO ERR-CALL-TYPE
        MOVE 'MESSAGE   ' TO ERR-TYPE
        MOVE SMTP-RETURN-CODE TO ERR-NUMBER
        MOVE ZEROES TO ERR-RETURN-CODE
        MOVE 'TRANSACTION HAS BEEN CANCELLED  ' TO ERR-TEXT
        PERFORM WRITEQ-TRANSIENT-DATA
        MOVE SPACES TO BUFFER-OUT
        MOVE MAIL-MESSAGE-RSET TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
        MOVE SPACES TO BUFFER-OUT
        MOVE MAIL-MESSAGE-END TO BUFFER-OUT
        PERFORM SEND-SOCKET
        PERFORM RECEIVE-SOCKET
        PERFORM CLOSE-SOCKET
        GO TO CLEAR-EM-OUT.
    EVALUATE-STARTBROWSE-RESPONSE.
        EVALUATE FILE-RESPONSE
            WHEN DFHRESP(DISABLED)
                MOVE 'DISABLED  ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
```

```
            WHEN DFHRESP(FILENOTFOUND)
                MOVE 'FILENOTFND' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(ILLOGIC)
                MOVE 'ILLOGIC   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(INVREQ  )
                MOVE 'INVREQ    ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(IOERR   )
                MOVE 'IOERR     ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(ISCINVREQ)
                MOVE 'ISCINVREQ ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(NOTAUTH)
                MOVE 'NOTAUTH   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(NOTFND)
                MOVE 'NOTFND    ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(NOTOPEN)
                MOVE 'NOTOPEN   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        END-EVALUATE.
        PERFORM WRITEQ-TRANSIENT-DATA
        CONTINUE.
    EVALUATE-READNEXT-RESPONSE.
        EVALUATE FILE-RESPONSE
            WHEN DFHRESP(FILENOTFOUND)
                MOVE 'FILENOTFND' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(ILLOGIC)
                MOVE 'ILLOGIC   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(INVREQ  )
                MOVE 'INVREQ    ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(IOERR   )
                MOVE 'IOERR     ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(ISCINVREQ)
                MOVE 'ISCINVREQ ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(LENGERR)
                MOVE 'LENGERR   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(NOTAUTH)
                MOVE 'NOTAUTH   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
```

```
            WHEN DFHRESP(NOTFND)
                MOVE 'NOTFND    ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        END-EVALUATE.
        PERFORM WRITEQ-TRANSIENT-DATA
        CONTINUE.
    EVALUATE-ENDBROWSE-RESPONSE.
        EVALUATE FILE-RESPONSE
            WHEN DFHRESP(FILENOTFOUND)
                MOVE 'FILENOTFND' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(ILLOGIC)
                MOVE 'ILLOGIC   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(INVREQ  )
                MOVE 'INVREQ    ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(IOERR   )
                MOVE 'IOERR     ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(ISCINVREQ)
                MOVE 'ISCINVREQ ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
            WHEN DFHRESP(NOTAUTH)
                MOVE 'NOTAUTH   ' TO ERR-TYPE
                MOVE 'TRANSACTION HAS BEEN RESCHEDULED' TO ERR-TEXT
        END-EVALUATE.
        PERFORM WRITEQ-TRANSIENT-DATA
        CONTINUE.
    WRITEQ-TRANSIENT-DATA.
        EXEC CICS WRITEQ TD QUEUE('CSMT')
            FROM(ERROR-MESSAGE)
            LENGTH(LENGTH OF ERROR-MESSAGE)
            RESP(TRANSDATA-RESPONSE)
            END-EXEC
        IF TRANSDATA-RESPONSE = DFHRESP(NORMAL)
            CONTINUE
        ELSE
            MOVE 'WRITE TD' TO OPR-CALL-TYPE
            MOVE 'SEVERE ERR' TO OPR-TYPE
            MOVE 'TRANSIENT DATA WRITE FAILED' TO OPR-TEXT
            EXEC CICS WRITE OPERATOR
                TEXT(OPERATOR-MESSAGE)
                TEXTLENGTH(LENGTH OF OPERATOR-MESSAGE)
                RESP(OPERATOR-RESPONSE)
                END-EXEC
        END-IF
        CONTINUE.
    ABEND-ERR.
        EXEC CICS ASSIGN ABCODE(ERR-TYPE) END-EXEC
```

```
     MOVE 'MAIL TRANSMISSION HAS ABENDED   ' TO ERR-TEXT
     PERFORM WRITEQ-TRANSIENT-DATA
     GO TO CLEAR-EM-OUT.
 CLEAR-EM-OUT.
     EXEC CICS RETURN END-EXEC
     STOP RUN.
```

DESKTOP TCP/IP LISTENER

The listener is developed in C using Visual Studio. It is started when the PC is booted from the START MENU on a Microsoft NT machine. The program opens a dialog box and runs minimized to prevent interference with other activities. Each message received is logged in the dialogue box and to a file for historical review if necessary. Each message consists of a short numerical value representing the key of an identified error condition (page). The number is concatenated with a literal '.WAV' and used to access a prerecorded sound file that has the same name. It runs continuously from one machine restart to another.

```
/*-------------------------------------------------------
   Plays a sound based on a key from the mainframe os390
  -------------------------------------------------------*/
#include <windows.h>
#include "resource.h"
#include <stdio.h>
#include <iostream.h>
#define WM_SOCKET_NOTIFY (WM_USER + 1)
#define ID_TIMER1         1
#define ID_TIMER2         2
LRESULT CALLBACK WndProc   (HWND, UINT, WPARAM, LPARAM);
BOOL    CALLBACK MainDlg   (HWND, UINT, WPARAM, LPARAM);
void EditPrintf (HWND hwndEdit, TCHAR * szFormat, ...);
HINSTANCE hInst;
HWND      hwndModeless;
HWND      hwnd;
static HWND   hwndButton, hwndEdit;
FILE     *pPageLog;
int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
                    PSTR szCmdLine, int iCmdShow)
{
     static TCHAR szAppName[] = TEXT ("NetTime");
     MSG          msg;
     RECT         rect;
     WNDCLASS     wndclass;
     hInst = hInstance;
     wndclass.style         = 0;
```

```
     wndclass.lpfnWndProc   = WndProc;
     wndclass.cbClsExtra    = 0;
     wndclass.cbWndExtra    = 0;
     wndclass.hInstance     = hInstance;
     wndclass.hIcon         = LoadIcon (hInstance,MAKEINTRESOURCE
(IDI_ICON));
     wndclass.hCursor       = NULL;
     wndclass.hbrBackground = NULL;
     wndclass.lpszMenuName  = NULL;
     wndclass.lpszClassName = szAppName;
     if (!RegisterClass (&wndclass))
     {
          MessageBox (NULL, TEXT ("This program requires Windows NT!"),
                      szAppName, MB_ICONERROR);
          return 0;
     }
     hwnd = CreateWindow (szAppName, TEXT ("Play Mainframe Sounds"),
          WS_OVERLAPPED | WS_CAPTION | WS_SYSMENU |
          WS_BORDER | WS_MINIMIZEBOX,
          CW_USEDEFAULT, CW_USEDEFAULT,
          CW_USEDEFAULT, CW_USEDEFAULT,
          NULL, NULL, hInstance, NULL);
          // Create the modeless dialog box to go on top of the window
     hwndModeless = CreateDialog (hInstance, szAppName, hwnd, MainDlg);
          // Size the main parent window to the size of the dialog box.
          //   Show both windows.
     GetWindowRect (hwndModeless, &rect);
     AdjustWindowRect (&rect, WS_CAPTION | WS_BORDER, FALSE);
     SetWindowPos (hwnd, NULL, 0, 0, rect.right - rect.left,
             rect.bottom - rect.top, SWP_NOMOVE);
     ShowWindow (hwndModeless, SW_SHOW);
     ShowWindow (hwnd, iCmdShow);
     UpdateWindow (hwnd);
          // Normal message loop when a modeless dialog box is used.
     while (GetMessage (&msg, NULL, 0, 0))
     {
         if (hwndModeless == 0 || !IsDialogMessage (hwndModeless, &msg))
          {
              TranslateMessage (&msg);
              DispatchMessage (&msg);
          }
     }
     return msg.wParam;
}
LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM
lParam)
{
        switch (message)
     {
     case WM_SETFOCUS:
          SetFocus (hwndModeless);
```

```
                return Ø;

      case WM_DESTROY:
           PostQuitMessage (Ø);
           return Ø;
        }
      return DefWindowProc (hwnd, message, wParam, lParam);
}
void EditPrintf (HWND hwndEdit, TCHAR * szFormat, ...)
{
      TCHAR   szBuffer [1Ø24];
      va_list pArgList;

      va_start (pArgList, szFormat);
      wvsprintf (szBuffer, szFormat, pArgList);
      va_end (pArgList);

      SendMessage (hwndEdit, EM_SETSEL, (WPARAM) -1, (LPARAM) -1);
      SendMessage (hwndEdit, EM_REPLACESEL, FALSE, (LPARAM) szBuffer);
      SendMessage (hwndEdit, EM_SCROLLCARET, Ø, Ø);
}
BOOL CALLBACK MainDlg (HWND hwnd, UINT message, WPARAM wParam, LPARAM
lParam)
{
        int      index;
        int      iSize;
        int      iError;
        int static bConnectStatus = FALSE;
        char     szSoundName[] = {"d:\\popalarm\\xxxxxx.wav"};
        char     szSoundDflt[] = {"999ØØØ.wav"};
        char     szPlaySoundInit[] = {"Play Sound initialize "};
        char     szPlaySoundError[] = {"Play Sound accept err "};
        char     szPlaySoundClose[] = {"Play Sound terminate  "};
        char     szPlaySoundConnected[] = {"Play Sound connected  "};
        char     szBufferOut[5] = {84,67,8Ø,9Ø,Ø};
        char     szBufferIn[7] = {"      "};
        char     szDate [32];
        char     szTime [32];
        struct   stPageRecord
        {
              char Literal1[22];
              char PageNum[6];
              char Literal2[1];
              char Date[1Ø];
              char Literal3[1];
              char Time[11];
              char Eol[1];
              char Eor[1];
        };
        struct stPageRecord RecordOut =
        {
```

```
            "Received page number  ",
            "      ",
            " ",
            "00,00,0000",
            " ",
            "00:00:00   ",
            "\n",
            "\0"
       };
       const  char * pRecordOut = RecordOut.Literal1;
static SOCKET sock;
static SOCKET sockaccepted;
static char   szIPAddr[32] = { "10.127.1.122" };
static int    iPort = 3000;
static TCHAR  szOKLabel[32];
static struct sockaddr_in sl;
       WORD          wEvent, wError;
     WSADATA       WSAData;
     switch (message)
     {
     case WM_INITDIALOG:
         hwndButton = GetDlgItem (hwnd, IDOK);
         hwndEdit = GetDlgItem (hwnd, IDC_TEXTOUT);

       if ((pPageLog = fopen("d:\\popalarm\\pagelog.txt","a+")) == NULL)
            EditPrintf (hwndEdit, TEXT ("Page Log OPEN failed! \r \n"));
             GetDateFormat (0,0,0,0,szDate,32);
             GetTimeFormat (0,0,0,0,szTime,32);
             for(index = 0;index < 10;index++)
                  RecordOut.Date[index] = szDate[index];
             for(index = 0;index < 11;index++)
                  RecordOut.Time[index] = szTime[index];
             if (RecordOut.Time[10] == '\0')
                    RecordOut.Time[10] = ' ';
             for(index = 0;index < 22;index++)
                  RecordOut.Literal1[index] = szPlaySoundInit[index];
             if (fputs(pRecordOut,pPageLog) != 0)
            EditPrintf (hwndEdit, TEXT ("Page Log WRITE failed! \r\n"));
                       // Call "WSAStartup" and display description text
         if (iError = WSAStartup (MAKEWORD(2,0), &WSAData))
         {
          EditPrintf (hwndEdit, TEXT ("Startup error #%i.\r\n"),iError);
             return TRUE;
         }
         EditPrintf (hwndEdit, TEXT ("Started up
%hs\r\n"),WSAData.szDescription);
                 // Call "socket"
         sock = socket (AF_INET, SOCK_STREAM, IPPROTO_TCP);
         if (sock == INVALID_SOCKET)
         {
             EditPrintf (hwndEdit,TEXT ("Socket creation error
```

```
#%i.\r\n"),WSAGetLastError ());
            WSACleanup ();
            return TRUE;
        }
        EditPrintf (hwndEdit, TEXT ("Socket %i created.\r\n"), sock);
                    // Call "WSAAsyncSelect"
        if (SOCKET_ERROR == WSAAsyncSelect (sock, hwnd,
WM_SOCKET_NOTIFY,FD_CONNECT | FD_READ | FD_ACCEPT))
            {
            EditPrintf (hwndEdit,TEXT ("WSAAsyncSelect error
#%i.\r\n"),WSAGetLastError ());
            closesocket (sock);
            WSACleanup ();
            return TRUE;
        }
            sl.sin_family           = AF_INET;
        sl.sin_port             = htons (3000);
        sl.sin_addr.S_un.S_addr = inet_addr (szIPAddr);
        if (SOCKET_ERROR == bind(sock, (SOCKADDR *) &sl, sizeof (sl)))
        {
                  iError = WSAGetLastError ();
          EditPrintf (hwndEdit, TEXT ("Listen error #%i.\r\n"),iError);
            closesocket (sock);
            WSACleanup ();
            return TRUE;
        }
            EditPrintf (hwndEdit, TEXT ("Socket bound to: %hs port %i
\r\n"), szIPAddr, iPort);
                    // Call "Listen"
                    // The result of the "listen" call will be reported
                    // through the WM_SOCKET_NOTIFY message.
        if (SOCKET_ERROR == listen(sock, SOMAXCONN))
            {
            iError = WSAGetLastError ();
          EditPrintf (hwndEdit, TEXT ("Listen error #%i.\r\n"),iError);
            closesocket (sock);
            WSACleanup ();
            return TRUE;
        }
            GetDateFormat (0,0,0,0,szDate,32);
            GetTimeFormat (0,0,0,0,szTime,32);
            EditPrintf (hwndEdit, TEXT ("Now listening for connections
%hs %hs \r\n"), szDate, szTime);
            return TRUE;
       case WM_COMMAND:
          switch (LOWORD (wParam))
          {
          case IDC_CLOSE:
               if (sock)
               {
                           closesocket (sock);
```

```
                    sock = Ø;
                    WSACleanup ();
                          GetDateFormat (Ø,Ø,Ø,Ø,szDate,32);
                          GetTimeFormat (Ø,Ø,Ø,Ø,szTime,32);
                          for(index = Ø;index < 1Ø;index++)
                                 RecordOut.Date[index] = szDate[index];
                              for(index = Ø;index < 11;index++)
                                 RecordOut.Time[index] = szTime[index];
                              if (RecordOut.Time[1Ø] == '\Ø')
                               RecordOut.Time[1Ø] = ' ';
                              for(index = Ø;index < 22;index++)
                    RecordOut.Literal1[index] = szPlaySoundClose[index];
                          if (fputs(pRecordOut,pPageLog) != Ø)
              EditPrintf (hwndEdit, TEXT ("Page Log WRITE failed! \r\n"));
                          if (fclose (pPageLog) != Ø)
              EditPrintf (hwndEdit, TEXT ("Page Log CLOSE failed! \r\n"));
                    }
                        DestroyWindow (GetParent (hwnd));
                    return TRUE;
               }
               return FALSE;
          case WM_SOCKET_NOTIFY:
               wEvent = WSAGETSELECTEVENT (lParam);   // ie, LOWORD
               wError = WSAGETSELECTERROR (lParam);   // ie, HIWORD
                    // Process two events specified in WSAAsyncSelect
               switch (wEvent)
               {
                    // This event occurs as a result of the "Listen" call
               case FD_ACCEPT:
                          sockaccepted = accept(sock,Ø,Ø);
                           if (sockaccepted == INVALID_SOCKET)
                           {
                          iError = WSAGetLastError ();
                                GetDateFormat (Ø,Ø,Ø,Ø,szDate,32);
                                GetTimeFormat (Ø,Ø,Ø,Ø,szTime,32);
                                  EditPrintf (hwndEdit, TEXT ("Accept error
#%i %hs %hs \r\n"),iError, szDate, szTime);

                                for(index = Ø;index < 1Ø;index++)
                                        RecordOut.Date[index] = szDate[index];
                                    for(index = Ø;index < 11;index++)
                                        RecordOut.Time[index] = szTime[index];
                                    if (RecordOut.Time[1Ø] == '\Ø')
                                      RecordOut.Time[1Ø] = ' ';
                                    for(index = Ø;index < 22;index++)
                          RecordOut.Literal1[index] = szPlaySoundError[index];
                                if (fputs(pRecordOut,pPageLog) != Ø)
              EditPrintf (hwndEdit, TEXT ("Page Log WRITE failed! \r\n"));
                          return TRUE;
                    }
                           //GetDateFormat (Ø,Ø,Ø,Ø,szDate,32);
```

```
                    //GetTimeFormat (0,0,0,0,szTime,32);
                    //EditPrintf (hwndEdit, TEXT ("Accepted connection
%hs %hs \r\n"), szDate, szTime);
                    return TRUE;
          case FD_READ:
               if (wError)
               {
              EditPrintf (hwndEdit, TEXT ("FD_READ error #%i."),wError);
                    SendMessage (hwnd, WM_COMMAND, IDCANCEL, 0);
                    return TRUE;
               }
               iSize = recv (sockaccepted, (char *) szBufferIn, 6, 0);
                    for(index = 0;index < 6;index++)
                    {
                         szSoundName[index+12] = szBufferIn[index];
                    }
                    if (szSoundName[12] != szSoundDflt[0] ||
                         szSoundName[13] != szSoundDflt[1] ||
                         szSoundName[14] != szSoundDflt[2] ||
                         szSoundName[15] != szSoundDflt[3] ||
                         szSoundName[16] != szSoundDflt[4] ||
                         szSoundName[17] != szSoundDflt[5])
                    {
                              if (TRUE !=
PlaySound(szSoundName,0,(SND_SYNC | SND_FILENAME)))
                                   EditPrintf(hwndEdit, TEXT ("Sound file
%hs not played.. \r\n"),szSoundName);
                         GetDateFormat (0,0,0,0,szDate,32);
                        GetTimeFormat (0,0,0,0,szTime,32);
                         EditPrintf (hwndEdit, TEXT ("Received from
mainframe %hs, %hs, %hs \r\n"), szBufferIn,szDate,szTime);
                         for(index = 0;index < 6;index++)
                         RecordOut.PageNum[index] = szBufferIn[index];
                  for(index = 0;index < 10;index++)
                              RecordOut.Date[index] = szDate[index];
                         for(index = 0;index < 11;index++)
                              RecordOut.Time[index] = szTime[index];
                         if (RecordOut.Time[10] == '\0')
                              RecordOut.Time[10] = ' ';
                       if (fputs(pRecordOut,pPageLog) != 0)
            EditPrintf (hwndEdit, TEXT ("Page Log WRITE failed! \r\n"));
                    }
               return TRUE;
          }
          return FALSE;
     }
     return FALSE;
}
```


CICS/TS 1.3 Web enhancements

The past year has seen many enhancements to the base CICS Web Support (CWS) API that arrived with CICS/TS 1.3. This article will review the changes and provide information on the current APAR list required to bring your CICS system up to speed.

APPLICATION CHANGES

The majority of these enhancements have been directly related to the Document and Web APIs, introducing new functionality that will extend the capabilities of Web-enabled CICS applications.

In some cases these API enhancements simply make it easier for application programmers to code CICS Web programs.

SYSTEM CHANGES

System programmers also have changes to contend with.

Secure sockets support is now enabled and recent enhancements to TCPIPSERVICES make it extremely easy to implement security without reliance on the sample analyser programs originally available with the CICS Web Interface.

APAR PQ28513

APAR PQ28513 provides CICS Web applications with an improved method of handling incoming HTTP requests containing forms.

Prior to this APAR, an application programmer would have to issue an EXEC CICS WEB RECEIVE command to gain access to the user data returned in the body of HTML form.

The programmer would first have to check whether any data was received. (Unfortunately this command would return DFHRESP(NORMAL) even though there was no input data.)

Next, the programmer would have to parse the incoming data to pick

up the various name/value pairs returned from the form. (Look at the April/May 2000 editions of *CICS Update* for an example of the code used in the document newcopy application, which used this technique).

Format:

```
EXEC CICS WEB READ
           FORMFIELD (data-area)
           NAMELENGTH(data-value)
           VALUE(data-area)
           VALUELENGTH(data-area)
           CLNTCODEPAGE(name)
           HOSTCODEPAGE(name)

EXEC CICS WEB
           STARTBROWSE FORMFIELD(data-area)
           NAMELENGTH(data-value)
           CLNTCODEPAGE(name)
           HOSTCODEPAGE(name)

EXEC CICS WEB
           READNEXT
           FORMFIELD (data-area)
           NAMELENGTH(data-area)
           VALUE(data-area)
           VALUELENGTH(data-area)

EXEC CICS WEB
           ENDBROWSE FORMFIELD
```

The new API initially allowed a CWS application to selectively read or browse the entire set of name/value pairs in the HTML forms data of an inbound HTTP request without issuing a prior EXEC CICS WEB RECEIVE command.

PQ35708 further enhances the STARTBROWSE FORMFIELD API by providing the ability to specify a start point for browsing HTML forms data.

If used, browsing of the HTML forms data will now start at the first formfield matching the name passed by the application.

Namelength must be specified if formfield(data-area) is specified on the startbrowse.

The Web READ and Web STARTBROWSE commands also support the following options for codepage translation –

CLNTCODEPAGE(name) and HOSTCODEPAGE(name).

If not specified, CICS performs ASCII / EBCDIC translation based on the charset parameter on the Content-Type header of the incoming HTTP request.

If the Content-Type header is not present, CICS uses client codepage 819 (ISO-8859–1) and host codepage 037.

APAR PQ28513 ADDITIONAL CHANGES

The following keywords are added to the EXEC CICS WEB EXTRACT command – Querystring(data-area) and querystrlen(data-value):

```
EXEC CICS  WEB EXTRACT
  HTTPMETHOD(data-area)    METHODLENGTH(data-value)
  HTTPVERSION(data-area)   VERSIONLEN(data-value)
  PATH(data-area)          PATHLENGTH(data-value)
  QUERYSTRING(data-area)   QUERYSTRLEN(data-value)
  REQUESTTYPE(cvda)
```

This provides easy access to any form data passed using the GET method.

Unfortunately you will still be required to parse the data yourself.

The following keywords are added to EXEC CICS DOCUMENT CREATE and EXEC CICS DOCUMENT SET commands – DELIMITER(DATA-VALUE) and UNESCAPED.

DELIMITER is to allow applications to use a delimiter other than ampersand to be included in symbol lists referenced by the SYMBOLLIST option of these commands.

Currently a SYMBOLLIST has to be of the form:

```
SYMBOL1=VALUE1&SYMBOL2=VALUE2&SYMBOL3=VALUE3
```

If the value fields were allowed to contain unsafe URL characters, for example an ampersand, the parsing logic would blow up.

Prior to PQ28513 you were forced to convert all unsafe URL characters to an encoded equivalent. For example, if the user data contains an ampersand, it would have to replaced with the symbol text ‘%26’.

This can create a major headache for application writers.

The delimiter enhancement allows application programmers to specify their own delimiter. The application must guarantee that the delimiter specified does not appear anywhere in symbol text.

EXAMPLE OF URL ENCODED DATA

Figure 1 was entered on a CICS Web page.

Figure 1: Example data

The incoming data can now be processed by either the FORMFIELD API or the original WEB RECEIVE command.

Data returned by EXEC CICS READ/READNEXT FORMFIELD is always in its unescaped form:

```
EXEC CICS WEB READNEXT
 FORMFIELD ('function')
 NAMELENGTH (8)
 VALUE ('       ')
 VALUELENGTH (7)
 NOHANDLE
EXEC CICS WEB READNEXT
 FORMFIELD ('dept')
 NAMELENGTH (4)
 VALUE ('1')
 VALUELENGTH (1)
 NOHANDLE
EXEC CICS WEB READNEXT
 FORMFIELD ('emp')
```

```
 NAMELENGTH (3)
 VALUE ('1&1')
 VALUELENGTH (3)
 NOHANDLE
```

Data returned by EXEC CICS WEB RECEIVE is always in its encoded form:

```
function=++++++++&dept=1&emp=1%261
```

When generating data to be sent to a Web browser the DOCUMENT CREATE and SET enhancements make it easier to process ‘unsafe’ URL characters.

Examples of the new SYMBOLLIST with DELIMITER(‘#’) are:

```
SYMBOL1=BLOGGS & CO#SYMBOL2=VALUE2#SYMBOL3=VALUE3
```

```
EXEC CICS DOCUMENT CREATE
 DOCTOKEN ('.......o        ')
 TEMPLATE ('TCWSPHDR
 SYMBOLLIST ('deptv= #empv=     #namev=
 LISTLENGTH (221)
 DELIMITER ('#')
 DOCSIZE (7215)

 EXEC CICS DOCUMENT SET
  DOCTOKEN ('.......q        ')
  SYMBOLLIST ('deptv=1#empv=00004#namev=BLOGGS & CO
  DELIMITER ('#')
  LENGTH (221)
```

If the UNESCAPED option is used on the DOCUMENT CREATE and SET commands, it prevents CICS from automatically unescaping symbol values contained in the SYMBOLLIST.

This may be useful when testing your application, because the encoded characters will be displayed on the browser.

APAR PQ35709

```
EXEC CICS WEB READ
           FORMFIELD (data-area)
           NAMELENGTH(data-value)
           SET(ptr-ref)
           VALUELENGTH(data-area)
```

The EXEC CICS WEB READ FORMFIELD command has had a

SET option added to it.

SET when used will point to an area containing the value for the specified formfield.

This option is primarily intended to let CICS support file uploads from multipart/form-data.

HTTP request header:

```
Content-Type: multipart/form-data; boundary=--------------------------
7d031937b07d8
```

Form data:

```
-----------------------------7d031937b07d8
Content-Disposition: form-data; name="input_data";
filename="F:\TEMP\sample.txt"
Content-Type: text/plain

 sample text file to upload
-----------------------------7d031937b07d8
Content-Disposition: form-data; name="submit"

Upload file
-----------------------------7d031937b07d8
```

The HTTP request header Content-Type informs you of the boundary marker used in your document

The above example shows the boundary marker used to denote the beginning and end of the uploaded file.

CURRENT LIMITATIONS

Please note that the formfield API is not currently supported by the indirect GWAPI interface as used by the IBM Websphere Web server. Only direct connections using a TCPIPSERVICE are currently supported.

APAR PQ36169

The last part of this article focuses on changes to TCPIPSERVICES and Web/CICS security.

Prior to APAR PQ36169, the security implementation for direct Web/CICS transactions was dependent on the systems programmer implementing some flavour of the user-replaceable analyser program.

For non-SSL connections, two sample sets of analysers were made available with CICS, one using an HTML sign-on dialog, the other using Basic Authentication headers. (Further sample security analysers are available on the CICS Supportpac Web site.)

If you're using SSL connections, client authentication was made almost impossible because of the lack of support for automatic registration of client certificates.

With this APAR we now have a much simpler way to manage SSL and non-SSL connections using the new TCPIPSERVICE parameter AUTHENTICATE:

```
AUTHENTICATE(NO|BASIC|CERTIFICATE|AUTOREGISTER|AUTOMATIC)
```

where:

- NO – no authentication of the client is required. However, if a registered certificate is provided by the client, it is used. This is the default.
- BASIC – HTTP Basic Authentication of the client is attempted. You do not need to use the sample analysers to implement HTTP basic authentication.
- CERTIFICATE with SSL(CLIENTAUTH) – a valid X.509 client certificate is required from the client, and it must map to a valid trusted user-id in the external security manager's database. If such a certificate is not received, the connection is rejected with an HTTP 403 response.
- AUTOREGISTER with SSL(CLIENTAUTH) – this allows the client to register a certificate automatically. If the client presents a certificate that is not registered, an HTTP Basic Authentication dialogue is entered in which the client must enter the user-id for which the certificate is to be registered.
- AUTOMATIC – this combines the AUTOREGISTER and BASIC functions.

It attempts to authenticate the client as best it can:

```
OBJECT CHARACTERISTICS                                  CICS RELEASE = Ø53Ø
 CEDA  View TCpipservice( DUQ36429 )
   TCpipservice   : DUQ36429
   Group          : DACTEST
   Description    : TCPIPSERVICE DEFINITION AFTER UQ36429 APPLIED
   Urm            : DFHWBADX
   Portnumber     : Ø5ØØ7              1-65535
   Certificate    : S39Ø
   STatus         : Open        Open | Closed
   SSl            : Yes         Yes | No | Clientauth
   Authenticate   : AUTOMatic   No | Basic | Certificate | AUTORegister
                                         | AUTOMatic
   TRansaction    : CWXN
   Backlog        : ØØØ1Ø              Ø-32767
   TSqprefix      : DFHWEB
   Ipaddress      :
   SOcketclose    : ØØØØ3Ø              No | Ø-24ØØØØ (HHMMSS)



                                                SYSID=TOR1 APPLID=CIRCTOR1

PF 1 HELP 2 COM 3 END      6 CRSR 7 SBH 8 SFH 9 MSG 1Ø SB 11 SF 12 CNCL
```

CONCLUDING REMARKS

CICS/TS 1.3 gets better and better. The documentation for the new features is (at the time of writing) available only by referring to the documentation that comes with the APARs.

The formfield API makes it very easy to parse user data supplied in HTML forms, which use the ‘post’ method.

The changes to TCPIPSERVICES make implementing SSL in particular much simpler than before.

You can e-mail David at david_clancy@circle-dl2.com.

David Clancy
Circle Computer Group (UK)

December 1997 – November 2000 index

Items below are references to articles that have appeared in *CICS Update* since Issue 145, December 1997. References show the issue number followed by the page number(s). Back-issues of *CICS Update* are available back to issue 145 (December 1997). See page 2 for details.

Compuware has announced its Uniface CICS Connector, which can combine a component-based environment with direct access to legacy systems running under CICS. It uses native communication layers and protocols, without the need for other layers like screen scrapers or middleware.

This results in easier installation, faster performance, and more flexible leveraging of existing system software.

For further information contact:
Compuware, 31440 Northwestern Highway, Farmington Hills, MI 48334-2564, USA.
Tel: (248) 737 7300.
Compuware, 163 Bath Road, Slough, SL1 4AA, UK.
Tel: (01753) 774000.
URL: http://www.compuware.com/products/build.htm.

* * *

IBM has announced the CICS Transaction Server for VSE/ESA. It includes CICS Web Support, REXX for CICS, CICS Universal Client, and a CICS Transaction Gateway function. It also provides Web Support and a 3270 Bridge function.

This new release builds on the restructured CICS code base, allowing access to new CICS functions without having to migrate to OS/390. Users of CICS/VSE Version 2.3 can migrate to the new software to get the same features as the latest OS/390 version.

Among the specifics are CICS Web Support (CWS), which provides direct access to CICS applications from the Web, and the free inclusion of REXX for CICS TS. The software makes use of ESA/390 subsystem storage protection, and provides virtual storage constraint relief and expanded application programming support including a new External CICS Interface (EXCI) and a Front End Programming Interface (FEPI).

For further information contact your local IBM representative.
URL: http://www-4.ibm.com/software/ts/cics.

* * *

Sybase has announced its XJS/390 Enterprise Integrator, which is designed to speed creation of Web interfaces to mainframe processing and data. Developers can implement mainframe-based components in Mscript, a JavaScript-compatible language, or in XML. Small components can be used to extend existing applications to interact with mainframe processes and data.

XJS/390 provides classes to interface XML and Mscript to DB2, VSAM, and MQSeries data queues. It also includes a class to access CICS transactions written in COBOL or other languages, which means the new product can provide Web site developers with access to any data or processing available to a CICS COBOL transaction program.

Supported platforms include CICS/ESA and CICS TS. Development tools and LAN run-time components support NT and Solaris.

For further information contact:
Sybase, 6475 Christie Ave, Emeryville, CA 94608, USA.
Tel: (510) 922 3500.
URL: http://www.sybase.com/products/eaimiddleware/xjs390enterpriseintegrator.