บรรณานุกรม

ภาษาไทย

เดือน สินธุพันธ์ประทุม และ สำนวน หิรัญวงษ์. CU WRITER ศึกษาด้วยตนเอง. พิมพ์ครั้งที่ 2. กรุงเทพ มหานคร: โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2536.

สมบรูณ์ ตั้งใจขจร. CICS/VS . กรุงเทพ มหานคร: โครงการศึกษาต่อเนื่อง ฝ่ายวิชาการ จุฬาลงกรณ์มหาวิทยาลัย.

ไพศาล สงวนหมู่, น.ต.ดร., และ ยืน ภู่วรวรรณ, รศ. การสื่อสารข้อมูล และ ไมโครคอมพิวเตอร์เนตเวอร์ค. กรุงเทพ มหานคร: หจก. เอช-เอนการพิมพ์, 2532.

ภาษาอังกฤษ

Andrew S. Tanenbaum. Computer Network. 2nd ed. New Jersey: Prentice-Hall International Ediition.

IBM. IBM Virtual Storage Extended System Package Hardware and System Support Extensions. Version 2 Release 1. From SC33-6184-04, Bangkok: IBM Co., Ltd. (Thailand), 1987.

IBM. Networking VSE/System Package. Version 2 Release 1. From SC33-6180-1, Bangkok: IBM Co., Ltd. (Thailand), 1985.

IBM. Planning VSE/System Package. Version 2 Release 1. From SC33-6177-1, Bangkok: IBM Co., Ltd. (Thailand), 1985.

IBM. IBM PC 3270 Emulation Program. Version 3.00: System Planner's and User's Reference. Bangkok: IBM Co., Ltd. (Thailand).

IBM. Virtual Storage Extended System Package Using IBM 3270 Display Station and Personal Computer. Version 3 Release 1. From SC33-6308-00, Bangkok: IBM Co., Ltd. (Thailand), 1987.

IBM. Customer Information Control System CICS/DOS/VS. Version 1 Release 7 Resource Definition (MACRO), From SC33-0149-3, Bangkok: IBM Co., Ltd. (Thailand), 1987.

IBM. Customer Information Control System CICS/DOS/VS. Version 1 Release 7 CICS Supplied Transactions, From SC33-0080-4, Bangkok: IBM Co., Ltd. (Thailand), 1987.

IBM. VSE/Enterprise System Architecture Using IBM Workstations. Version 1 Release 1. From SC33-6509-00, Bangkok: IBM Co., Ltd. (Thailand), 1990.

IBM. VSE/Enterprise System Architecture Planning. Version 1 Release 1. From SC33-6503-00, Bangkok: IBM Co., Ltd. (Thailand), 1990.

Frank J. Derfler, Jr. PC Magazine Guide to Connectivity. 2 nd. ed. California: Ziff-Davis Press Emeryville, 1992.

Jame Martin, and Joe Leben. Data Communication Technology. New Jersey: Prentice-Hall International Editions, 1988

Jay Ranade, and Hirday Ranade. VSAM Concept, Programming, and Design. New York: Macmillan Publishing Company, 1986.

Joseph LeBert. CICS for microcomputers. Singapore: McGraw-Hall International Edition., 1989.

Uyless Black. Computer networks protocols, standards, and interfaces. New Jersey: Prentice-Hall International Editions, 1987.

ภาคผนวก

โปรแกรม และ ตาราง ที่ใช้บนพีซี และ เมนเฟรม

```
*************************************************************************
*  program name : enterprg.prg
*  function     : First check userid. If valid, will go to mainframe part.
*                 If not, will return to main.
*  created date : 1 Jan 94
*************************************************************************
procedure enterprg
do begin
do while .t.
        do logon
enddo
do finish


*************************************************************************
Procedure begin
close all
clear all
on  readerror   do nothing
set talk    off
set stat    off
set defa    to c:\thesis\files
set path    to c:\thesis\programs
set delete  on
set color   to w/n,n/w
set esca    off
set date    british
set border to 129,129,128,128,130,131,133,132
```

```
push key clear

on key label f1          do nothing
on key label f2          do nothing
on key label f3          do finish
on key label f4          do nothing
on key label f5          do nothing
on key label f6          do nothing
on key label f7          do nothing
on key label f8          do nothing
on key label f9          do nothing
on key label f10         do nothing
on key label shift-f1    do nothing
on key label shift-f2    do nothing
on key label shift-f3    do nothing
on key label shift-f4    do nothing
on key label shift-f5    do nothing
on key label shift-f6    do nothing
on key label shift-f7    do nothing
on key label shift-f8    do nothing
on key label shift-f9    do nothing
on key label shift-f10   do nothing
on key label ctrl-f1     do nothing
on key label ctrl-f2     do nothing
on key label ctrl-f3     do nothing
on key label ctrl-f4     do nothing
on key label ctrl-f5     do nothing
on key label ctrl-f6     do nothing
```

```
on key label ctrl-f7    do nothing
on key label ctrl-f8    do nothing
on key label ctrl-f9    do nothing
on key label ctrl-f10   do nothing
on key label escape     do nothing

define popup mf_popup from 5,18 to  13,45 title 'MF main menu'
define bar 1 of mf_popup prompt '1. เก็บข้อมูลบน MF'
define bar 2 of mf_popup prompt '2. ดึงข้อมูลจาก MF'
define bar 3 of mf_popup prompt '3. ลบข้อมูลบน MF'
define bar 4 of mf_popup prompt '4. ตรวจสอบข้อมูลบน MF'
define bar 5 of mf_popup prompt '5. เพิ่ม/ลบรหัสประจำตัว'
define bar 6 of mf_popup prompt '6. ปรับปรุงระบบแฟ้มข้อมูล'
define bar 7 of mf_popup prompt '7. เลิกการทำงาน'
on selection popup mf_popup deactivate popup

define popup chk_file from 5,20 to 20,50 prompt field mf_file+;
                                    ' + Dtoc(upd_date)
on selection popup chk_file deactivate popup

define window confirm from 10,1 to 13,78
define window reorg   from 15,1 to 18,78

Public user_id,count,mf_nam

use userid   order  userid  in  1
use listtab  order  listtab in  2
clear
```

```
user_id = SPACE(4)
m->count = 0


*****************************************************************************
Procedure logon

* Check all files were sent or received from host completely since last time*
* if not, ask for confirmation to re-do again **
re_do = 'Y'
do case
      case File("c:\thesis\tran\send_err")
          @ 12,8 say 'มี file ที่รอการส่งค้างอยู่ ท่านต้องการส่งหรือไม่'
          @ 12,58 get re_do PICT '!Y'
          read
          message = 'กรุณากด Key ใดๆ เพื่อส่ง File ไป Mainframe อีกครั้งหนึ่ง'
          if re_do = 'Y'
               @ 14,8 say message
               =inkey(0)
               do finish
               return
          else
               run c:\thesis\programs\del_file.bat
               if file('c:\thesis\files\thesis.mem')
                    rest from c:\thesis\files\thesis addi
                    select 2
                    seek mf_user + mf_nam
                    susp
```

```
            if Allt(dtoc(old_date)) = '/  /'   && new file &&
                delete
            else
                replace upd_date with old_date
            endif
        endif
    endif

case File("c:\thesis\tran\rec_err")
    @ 12,8  say 'มี file ที่รอการรับค้างอยู่ ท่านต้องการรับหรือไม่'
    @ 12,58 get re_do PICT '!Y'
    read
    message = 'กรุณากด Key ใดๆ เพื่อรับ Fileจาก Mainframe อีกครั้งหนึ่ง'
    if re_do = 'Y'
        @ 14,8 say message
        =inkey(0)
        do finish
        return
    else
        run c:\thesis\programs\del_file.bat
    endif

case File("c:\thesis\tran\del_err")
    @ 12,8  say 'มี file ที่รอการลบค้างอยู่ ท่านต้องการลบหรือไม่'
    @ 12,58 get re_do PICT '!Y'
    read
    message = 'กรุณากด Key ใดๆ เพื่อลบ Fileบน Mainframe อีกครั้งหนึ่ง'
    if re_do = 'Y'
```

```
            @ 14,8 say message
            =inkey(0)
            do finish
            return
        else
            run c:\thesis\programs\del_file.bat
            if file('c:\thesis\files\thesis.mem')
                rest from c:\thesis\files\thesis addi
                set delete off
                select 2
                seek mf_user + mf_nam
                if delete()
                        recall
                endif
            endif
        endif

case File("c:\thesis\tran\chk_err")
    @ 12,4  say 'มี file ที่รอการรับเพื่อตรวจสอบค้างอยู่ ท่านต้องการรับหรือไม่'
    @ 12,58 get re_do PICT '!Y'
    read
    message = 'กรุณากด Key ใดๆ เพื่อรับ Fileจาก Mainframe อีกครั้งหนึ่ง'
    if re_do = 'Y'
            @ 14,8 say message
            =inkey(0)
            do finish
            return
    else
```

```
                run c:\thesis\programs\del_file.bat
            endif

endcase
clear
Do while .t.
    @  0,0 to 23,79
    @ 13,20 say 'กรุณาใส่รหัสประจำตัว : '
    @ 13,col() get user_id Pict '@!'
    read
    @ 24,0
    m->count = m->count + 1
    if user_id <> 'ZZZZ'
        seek   user_id
        if not found()
            @ 24,8 say 'รหัสประจำตัวไม่ถูกต้อง กรุณาใส่รหัสอีกครั้ง.'
            if m->count = 3
                exit
            endif
            loop
        endif
    endif
    do mfmenu
    m->count = 0
enddo
do finish
return
```

```
*****************************************************************************
*  This procedure's purpose is to be a dummy procedure.                     *
*****************************************************************************
procedure nothing
return


*****************************************************************************
procedure finish
clear all
close all
release all
set date american
on  readerror
pop key
quit
return
```

```
*****************************************************************************
*  program name : mfmenu.prg
*  function     : mainframe main menu. It is used to call other sub program
*                 which is
*                     1. fileprg.prg
*                     2. updprg.prg
*                     3. delfile.prg
*                     4. chkfile.prg
*                     5. user_id.prg
*                     6. reorg.prg
*****************************************************************************
Procedure mfmenu
do while .t.
        clear
        activate popup mf_popup
        prompt = left(prompt(),1)
        do case
                case prompt = '1'
                        do fileprg
                case prompt = '2'
                        do updprg
                case prompt = '3'
                        do delfile
                case prompt = '4'
                        do chkfile
                case prompt = '5'
                        do user_id
                case prompt = '6'
```

```
                do reorg
        case prompt = '7'
                do finish
    endcase
enddo


*******************************************************************************
Procedure finish
clear
quit
return
```

```
*******************************************************************************
*  program name : fileprg.prg
*  function     : send pc file to mainframe
*******************************************************************************
clear
_pc    = .t.
pc_nam = 'c:\'+SPACE(47)
mf_nam = SPACE(8)
Do while .t.
      @  0,0 clear to 22,79
      @  0,0 to 22,79
      @ 08,10 say 'โปรดระบุชื่อแฟ้มข้อมูลบน PC ที่ท่านต้องการเก็บ'
      @ 10,10 say '==> '
      @ 10,col() get pc_nam    VALID pc_name(pc_nam);
                          when  _pc
      @ 12,10 say 'โปรดระบุชื่อแฟ้มข้อมูลบน MF'
      @ 14,10 say '==> '
      @ 14,col() get mf_nam    Pict '@!';
                          valid mf_chk(mf_nam)
      read
      @ 23,0
      _pc     = .f.
      pc_nam1 = ALLT(pc_nam)

      ** seperate directory and pc file name **
      len_file = LEN(pc_nam1)
      len      = len_file
      do while .t.
```

```
        x  = Substr(pc_nam1,len,1)
        If x <> '\'
                len = len -1
        else
                exit
        endif
enddo
f_nam = Upper(Substr(pc_nam1,len+1))
If LEN(ALLT(f_nam)) = 0
        return
endif
If mf_nam = SPACE(8)
        leng = 1
        do while leng <= LEN(f_nam)
                y  = Substr(f_nam,leng,1)
                If y <> '.'
                        leng = leng + 1
                else
                        exit
                endif
        enddo
        mf_nam = PADR(Upper(Left(f_nam,leng-1)),8,' ')
else
        mf_nam = PADR(ALLT(mf_nam),8,' ')
endif
  if not mf_chk(mf_nam)
          loop
  endif
```

```
sele listtab
seek user_id+mf_nam
old_file = .f.
If found()
        @ 14,14 say mf_nam color n/w
        confirm = ' '
        @ 16,10 say 'ชื่อแฟ้มข้อมูลบน MF มีอยู่แล้ว'
        @ 17,10 say 'ต้องการใช้ชื่อเดิมหรือไม่ (Y/N)? '
        @ 17,col() get confirm  PICT 'Y' color w/n,w/n
        @ 19,10 say 'Y = แฟ้มข้อมูลบน MF จะถูก Write ทับ'
        @ 20,10 say 'N = เปลี่ยนชื่อแฟ้มข้อมูลบน MF'
        read
        @ 16,10 clear to 20,78
        If confirm = 'N'
                @ 23,0
                @ 23,10 say 'กรุณาใส่ชื่อแฟ้มข้อมูลบน MFใหม่.'
                loop
        endif
        old_file = .t.
endif
copy file &pc_nam1 to c:\thesis\tran\&f_nam
if not old_file
        append blank
        replace userid   with user_id
        replace pc_file  with f_nam
        replace upd_date with date()
        replace mf_file  with mf_nam
else
```

```
        replace pc_file  with f_nam
        replace old_date with upd_date
        replace upd_date with date()

endif

mf_user = userid
if file('c:\thesis\files\thesis.mem')
        save to c:\thesis\files\thesis.tmp all like mf*
        dele file c:\thesis\files\thesis.mem
        rena c:\thesis\files\thesis.tmp to c:\thesis\files\thesis.mem
else
        save to c:\thesis\files\thesis.mem all like mf*
endif


send_comm = 'SEND '+'C:\THESIS\TRAN\'+Allt(pc_file)+' '+;
            Allt(mf_file)+;
            ' (ASCII CRLF FILE=TS'+;
            ' REPLACE PROGRAM=CFTRSEN'
set alter    to c:\thesis\tran\sendfile.
set safety  off
set console off
set alter   on
??'send file to MF'
close alter

set alter    to c:\thesis\tran\send_err.
```

```
        set alter   on
        ??'if no error while sending file, this file will be deleted'
        close alter

        set alter   to c:\pc3270\sen.bat
        set alter   on
        set alter   off

        set alter on
        ??'@echo off'
        ?'cls'
        ?'Echo File Sending, Please wait....'
        ?send_comm
        set alter off
        set console on
        set safety  on
        quit
enddo
return

*****************************************************************************
Procedure mf_chk
parameter edit_mf

@ 23,0
if Left(Allt(edit_mf),1) $ '0123456789'
        @ 23,10 say 'ชื่อแฟ้มข้อมูลบน MF ขึ้นต้นด้วยตัวเลขไม่ได้'
        return .f.
```

```
endif

store len(allt(edit_mf)) to leng
begin = 1
do while begin < leng
        x = substr(edit_mf,begin,1)
        if x $ '*:()<>'
            @ 23,10 say 'ชื่อแฟ้มข้อมูลบน MF ไม่ถูกตามข้อกำหนด กรุณาตั้งชื่อใหม่'
            return .f.
        endif
        begin = begin + 1
enddo
return

*********************************************************************************
procedure finish
clear
return to mfmenu


*********************************************************************************
Function  pc_name
parameter edit_name

@ 23,0
If Lower(edit_name) = 'c:\'+SPACE(47)
        @ 23,10 say 'กรุณาใส่ชื่อแฟ้มข้อมูล.'
        return .f.
```

```
else
        If not FILE('&edit_name')
                @ 23,10 say 'ไม่พบแฟ้มข้อมูล กรุณาใส่ชื่อใหม่.'
                return .f.
        Endif
Endif
return
```

```
*******************************************************************************
*  program name : updprg.prg
*  function     : update pc file and send to mainframe
*******************************************************************************
clear
mf_nam = SPACE(8)
sub_dir    = 'c:\'+SPACE(47)
sele listtab
calculate cnt() to num for userid = user_id
if num = 0
      ??chr(7)
      @ 23,0 say 'ไม่มีแฟ้มข้อมูลบน MF สำหรับ user '+user_id
      @ 23,col() say '. กด key ใดๆ เพื่อกลับไป MF Main Menu'
      =inkey(0)
      do finish
endif
Do while .t.
      @  0,0 clear to 23,79
      @  0,0 to 22,79
      @ 10,5 say 'โปรดใส่ชื่อแฟ้มข้อมูลบน MF ที่ท่านต้องการรับลงมาแก้ไข'
      @ 12,5 say '==> '
      @ 12,col() get mf_nam  Pict "@!"  valid r_file()
      read

      Do while .t.
          @ 14,5 say 'โปรดระบุชื่อแฟ้มข้อมูลและ Sub directory ที่ท่านต้องการนำแฟ้มข้อมูลไป
เก็บ'
          @ 16,9 get sub_dir
```

```
        read

@ 23,0
        If sub_dir = SPACE(50)
            @ 23,10 say 'กรุณาระบุชื่อแฟ้มข้อมูลและ Sub directory ใหม่'
            loop
        endif

        sub_dir = Allt(sub_dir)
        If right(sub_dir,1) $ '\:'   && user does not input file name &&
            store sub_dir+mf_nam to sub_dir
        endif

        If File('&sub_dir')
            @ 18,10 say 'แฟ้มข้อมูลเดิมมีอยู่แล้ว ต้องการ write ทับหรือไม่(Y/N)?'
            confirm = ' '
            do while confirm = ' '
                @ 18,col()+2 get confirm Pict 'Y'
                read
            enddo
            If confirm ='N'
                @ 18,10 clear to 18,78
                @ 23,10 say 'กรุณาระบุชื่อแฟ้มข้อมูลและ Sub directory ใหม่'
                loop
            Endif
        Endif
        exit
    enddo
```

```
mf_user = userid

if file('c:\thesis\files\thesis.mem')
        save to c:\thesis\files\thesis.tmp all like mf*
        dele file c:\thesis\files\thesis.mem
        rena c:\thesis\files\thesis.tmp to c:\thesis\files\thesis.mem
else
        save to c:\thesis\files\thesis.mem all like mf*
endif

rec_comm = 'RECEIVE '+'C:\THESIS\TRAN\'+Allt(pc_file)+' '+;
           Allt(mf_file)+;
           ' (ASCII CRLF FILE=TS'+;
           ' REPLACE PROGRAM=CFTRREC'
set alter   to c:\thesis\tran\recfile.
set safety  off
set console off
set alter   on
??'Receive file from MF'
close alter

set alter   to c:\thesis\tran\rec_err.
set alter   on
??'if no error while receiving file from Host, this file will be deleted'
close alter

from_file = 'c:\thesis\tran\'+Allt(pc_file)
```

```
        set alter    to c:\pc3270\rec.bat
        set alter    on

        set alter    off

        set alter on
        ??'@echo off'
        ?'cls'
        ?'Echo File Receiving, Please wait....'
        ?rec_comm
        ?'If exist ',from_file,' GOTO copy_f'
        ?'goto exit'
        ?
        ?':copy_f'
        ?'copy ',from_file,' ',sub_dir,' > nul'
        ?'Echo .'
        ?'Echo .'
        ?'Echo .'
        ?'Echo มีแฟ้มข้อมูลบน PC พร้อมที่จะดำเนินการแก้ไขแล้ว'
        ?'Echo โปรดกด key ใดๆ เพื่อทำงานต่อไป...'
        ?'pause > nul'
        ?
        ?':exit'
        set alter off
        set console on
        set safety  on
        quit
enddo
```

```
rele mf_nam
return

*********************************************************************************
procedure finish
clear
rele mf_nam
return to mfmenu

*********************************************************************************
procedure r_file
@ 24,0
sele    listtab
set     filt to userid = user_id
if EMPTY(mf_nam)
        found = .f.
else
        seek    user_id + mf_nam
        store   found() to found
endif
If not found
        @ 24,10 say 'กด Enter เพื่อเข้าสู่ Window หรือ กด Key อื่นเพื่อแก้ไขข้อมูล'
        If not Inkey(0) = 13
            @ 24,0
            @ 24,10 say 'กรุณาใส่ชื่อแฟ้มข้อมูลที่มีอยู่ในระบบแล้ว'
            return .f.
```

```
        endif
        @ 24,0
        save screen to scr
        @  2,15 say 'โปรดเลือกชื่อแฟ้มข้อมูลบน MF ที่ท่านต้องการรับลงมาแก้ไข'

       @   4,22 say 'ชื่อแฟ้มข้อมูล: วันที่แก้ไขครั้งสุดท้าย'
        activate popup chk_file
        set filt to
        store Allt(Left(prompt(),12)) to mf_nam
        restore screen from scr
        show gets
        @ 24,0
        If len(allt(mf_nam)) = 0
              @ 24,10 say 'กรุณาใส่ชื่อแฟ้มข้อมูลที่มีอยู่ในระบบแล้ว'
              return .f.
        endif
        return
endif
return
```

```
*******************************************************************************
*  program name : delfile.prg
*  function     : delete file from pc and mainframe
*******************************************************************************
set delete on
Public mf_nam
clear
mf_nam = SPACE(8)
sele listtab
calculate cnt() to num for userid = user_id
if num = 0
        ??chr(7)
        @ 23,0 say 'ไม่มีแฟ้มข้อมูลบน MF สำหรับ user '+user_id
        @ 23,col() say '. กด key ใดๆ เพื่อกลับไป MF Main Menu'
        =inkey(0)
        do finish
endif
Do while .t.
        @  0,0 clear to 23,79
        @  0,0 to 22,79
        @  2,15  say 'โปรดใส่ชื่อแฟ้มข้อมูลบน MF ที่ท่านต้องการลบ'
        @  4,15  get mf_nam   Pict "@!"  VALID  f_file()
        read

        delete

        mf_user = userid
        if file('c:\thesis\files\thesis.mem')
```

```
        save to c:\thesis\files\thesis.tmp all like mf*
        dele file c:\thesis\files\thesis.mem
        rena c:\thesis\files\thesis.tmp to c:\thesis\files\thesis.mem
else
        save to c:\thesis\files\thesis.mem all like mf*
endif

del_comm = 'RECEIVE '+'C:\THESIS\TRAN\'+Allt(pc_file)+' '+;
           Allt(mf_file)+;
           ' (ASCII CRLF FILE=TS'+;
           ' REPLACE PROGRAM=CFTRDEL'
set alter   to c:\thesis\tran\recfile.
set safety  off
set console off
set alter   on
??'Delete file on MF'
close alter

set alter   to c:\thesis\tran\del_err.
set alter   on
??'if no error while deleting file, this file will be deleted'
close alter

from_file = 'c:\thesis\tran\'+Allt(pc_file)

set alter   to c:\pc3270\rec.bat
set alter   on
set alter   off
```

```
        set alter on
        ??'@echo off'
        ?'cls'
        ?'Echo File Deleting, Please wait....'
        ?del_comm
        ?
        ?':exit'
        set alter off
        set console on
        set safety  on
        quit
enddo
rele  mf_nam
return

*****************************************************************************
procedure finish
set delete off
clear
rele  mf_nam
return to mfmenu


*****************************************************************************
procedure f_file
@ 24,0
sele    listtab
```

```
set     filt to userid = user_id
if EMPTY(mf_nam)
        found = .f.
else
        seek    user_id + mf_nam
        store   found() to found
endif
If not found
        @ 24,10 say 'กด Enter เพื่อเข้าสู่ Window หรือ กด Key อื่นเพื่อแก้ไขข้อมูล'
        If not Inkey(0) = 13
            @ 24,0
            @ 24,10 say 'กรุณาใส่ชื่อแฟ้มข้อมูลที่มีอยู่ในระบบแล้ว'
            return .f.
        endif
        @ 24,0
          @ 4,15
        save screen to scr
        @  2,15 say 'โปรดเลือกชื่อแฟ้มข้อมูลบน MF ที่ท่านต้องการลบ'
        @  4,22 say 'ชื่อแฟ้มข้อมูล: วันที่แก้ไขครั้งสุดท้าย'
        set deleted on
        activate popup chk_file
        set filt to
        set deleted off
        store Allt(Left(prompt(),12)) to mf_nam
        restore screen from scr
        show gets
        @ 24,0
        If len(allt(mf_nam)) = 0
```

```
            @ 24,10 say 'กรุณาใส่ชื่อแฟ้มข้อมูลที่มีอยู่ในระบบแล้ว'
            return .f.
        endif
        return
endif
return
```

```
*******************************************************************************
*  program name : chkfile.prg
*  function     : check all existing pc file sent to mainframe and can
*                 receive file from mainframe into pc.
*******************************************************************************
set delete on
clear
sub_dir    = 'c:\'+SPACE(47)
sele listtab
calculate cnt() to num for userid = user_id
if num = 0
        ??chr(7)
        @ 23,0 say 'ไม่มีแฟ้มข้อมูลบน MF สำหรับ user '+user_id
        @ 23,col() say '. กด key ใดๆ เพื่อกลับไป MF Main Menu'
        =inkey(0)
        do finish
endif
Do while .t.
        @  0,0 clear to 23,79
        @  0,0 to 22,79
        @  2,15  say 'โปรดเลือกชื่อแฟ้มข้อมูลบน MF ที่ท่านต้องการ'
        @  4,22  say 'ชื่อแฟ้มข้อมูล: วันที่แก้ไขครั้งสุดท้าย'
        set  filt to userid = user_id
        activate popup chk_file
        set filt to
        If len(prompt()) = 0
                do finish
        else
```

```
        mf_nam = Allt(Left(prompt(),8))
    endif
    seek user_id+mf_nam
    @ 6,22 say mf_nam + '          ' + Dtoc(upd_date)
    Do while .t.
        @ 14,5 say 'โปรดระบุชื่อแฟ้มข้อมูลและ Sub directory ที่ท่านต้องการนำแฟ้มข้อมูลไป
เก็บ'

        @ 16,9 get sub_dir
        read
        @ 23,0
        If sub_dir = SPACE(50)
            @ 23,10 say 'กรุณาระบุชื่อแฟ้มข้อมูลและ Sub directory ใหม่'
            loop
        endif

        sub_dir = Allt(sub_dir)
        If right(sub_dir,1) $ '\'
            store sub_dir+mf_nam to sub_dir
        endif

        If File('&sub_dir')
            @ 18,10 say 'แฟ้มข้อมูลเดิมมีอยู่แล้ว ต้องการ write ทับหรือไม่(Y/N)?'
            confirm = ' '
            do while confirm = ' '
                @ 18,col()+2 get confirm Pict 'Y'
                read
            enddo
            If confirm ='N'
```

```
                @ 18,10 clear to 18,78
                @ 23,10 say 'กรุณาระบุชื่อแฟ้มข้อมูลและ Sub directory ใหม่'
                loop
            Endif
        Endif
        exit
enddo

rec_comm = 'RECEIVE '+'C:\THESIS\TRAN\'+Allt(pc_file)+' '+;
           Allt(mf_file)+;
           ' (ASCII CRLF FILE=TS'+;
           ' REPLACE PROGRAM=CFTRREC'
set alter   to c:\thesis\tran\recfile.
set safety  off
set console off
set alter   on
??'Receive file from MF'
close alter

set alter   to c:\thesis\tran\rec_err.
set alter   on
??'if no error while receiving file, this file will be deleted'
close alter

from_file = 'c:\thesis\tran\'+Allt(pc_file)

set alter   to c:\pc3270\rec.bat
set alter   on
```

```
        set alter   off

        set alter on
        ??'@echo off'
        ?'cls'
        ?'Echo File Receiving, Please wait....'
        ?rec_comm
        ?'If exist ',from_file,' GOTO copy_f'
        ?'goto exit'
        ?
        ?':copy_f'
        ?'copy ',from_file,' ',sub_dir,' > nul'
        ?'Echo .'
        ?'Echo .'
        ?'Echo .'
        ?'Echo มีแฟ้มข้อมูลบน PC พร้อมที่จะดำเนินการตรวจสอบแล้ว'
        ?'Echo โปรดกด key ใดๆ เพื่อทำงานต่อไป...'
        ?'pause > nul'
        ?
        ?':exit'
        set alter off
        set console on
        set safety  on
        quit
enddo
return
```

```
**************************************************************************
procedure finish
set delete off
clear
return to mfmenu
```

```
*****************************************************************************
*  program name : user_id.prg
*       function      : to maintenance user id such as add, update, delete
*****************************************************************************
Procedure user_id
do begin
do while .t.
        do preprocess
enddo
do finish


*****************************************************************************
procedure begin
Public upd_id
clear
@ 0,0 to 23,79
if USED('userid')
    select userid
else
    use userid  order  userid in 1
endif
on key label f2 do update
on key label f6 do delete
upd_id = SPACE(4)
return
```

```
**************************************************************************
procedure preprocess
@ 24,0
@ 22,5 say 'F2 : เพิ่มรหัสประจำตัว F3 : ออกจากโปรแกรม     F6 : ลบรหัสประจำตัว'
@ 10,5 say 'ป้อนรหัสประจำตัวที่ต้องการเพิ่มหรือลบ : '
do process
return

**************************************************************************
procedure process
@ 10,col() get upd_id Pict '@!'
read
return

**************************************************************************
Procedure update
if len(allt(upd_id)) # 4
        @ 24,10 say 'กรุณาป้อนรหัสให้ครบ 4 ตำแหน่ง'
        return
endif
select userid
seek upd_id
if found()
        @ 24,10 say 'รหัสเดิมมีอยู่แล้ว ไม่สามารถเพิ่มได้อีก'
        return
endif
append blank
replace userid with upd_id
```

```
@ 24,10 say 'เพิ่มรหัสใหม่เรียบร้อยแล้ว'
return

*****************************************************************************
Procedure delete
if len(allt(upd_id)) # 4
        @ 24,10 say 'กรุณาป้อนรหัสให้ครบ 4 ตำแหน่ง'
        return
endif
select userid
seek upd_id
if .not. found()
        @ 24,10 say 'ไม่พบรหัสนี้ ไม่สามารถลบได้'
        return
endif
delete
pack
@ 24,10 say 'ลบรหัสเก่าเรียบร้อยแล้ว'
return

*****************************************************************************
Procedure finish
on key label f2 do nothing
on key label f6 do nothing
release upd_id
clear
return to mfmenu
```

```
*******************************************************************************
* program name : reorg.prg
* function     : to re-organize database, truncate some records which is
*              marked to be deleted.
*******************************************************************************
clear
activate window confirm
@ 0,0 say 'กด Enter ถ้าต้องการปรับปรุงระบบแฟ้มข้อมูล'
@ 1,0 SAY 'กดคีย์อื่นๆเพื่อยกเลิกคำสั่ง'
IF .NOT. INKEY(0) = 13
        DEACTIVATE WINDOW confirm
        RETURN to mfmenu
ENDIF
If USED('listtab')
        sele listtab
else
        use listtab  order  listtab in  2
endif
activate window reorg
@ 0,0 say 'กำลังทำการปรับปรุงแฟ้มข้อมูล, โปรดรอ'
sele listtab
pack
deactivate window confirm
deactivate window reorg
return
```

```
*********************************************************************************
*  batch file name : mesr.bat
*    funtion          : use in script file PCREC.PCS
*                       in order to show error from LMF
*********************************************************************************
@echo off
cls
if %1 == 0  goto show0
if %1 == 1  goto show1
if %1 == 2  goto show2
if %1 == 3  goto show3
if %1 == 4  goto show4
if %1 == 5  goto show5
if %1 == 6  goto show6
if %1 == 7  goto show7
if %1 == 8  goto show8
if %1 == 9  goto show9
if %1 == 10 goto show10
if %1 == 11 goto show11
if %1 == 12 goto show12
goto exit

:show0
echo INW0025I- รับข้อมูลเรียบร้อยแล้ว
if exist c:\thesis\tran\recfile. del c:\thesis\tran\recfile.
if exist c:\thesis\tran\del_err. call c:\thesis\programs\del_file.bat
if exist c:\thesis\tran\rec_err. del c:\thesis\tran\rec_err.
if exist c:\thesis\tran\chk_err. del c:\thesis\tran\chk_err.
```

```
goto exit0

:show1
rem  INW0027I- NO CICS COMMUNICATION AREA IS GIVEN TO CFTRRB02
echo INW0027I- ไม่มีที่รองรับในการติดต่อสื่อสารข้อมูล
goto exit1

:show2
echo INW0022I- รับข้อมูลขึ้นเรียบร้อยแล้วแต่I/Oมีปัญหา
goto exit1

:show3
rem  INW0027I- ERROR IN THE CICS COMMAND =STARTBR= WHEN ACCESSING
RDBUPD
echo INW0027I- มีข้อผิดพลาดในการอ่านแฟ้มข้อมูล
goto exit1

:show4
rem  INW0027I- ERROR IN THE CICS COMMAND =READNEXT= WHEN ACCESSING
RDBUPD
echo INW0027I- มีข้อผิดพลาดในการอ่าน record ต่อมา
goto exit1

:show5
rem  INW0027I- ERROR IN THE CICS COMMAND =ENDBR= WHEN ACCESSING RDBUPD
echo INW0027I- มีข้อผิดพลาดในการเลิกอ่าน
goto exit1
```

```
:show6
rem  INW0027I- ERROR IN THE CICS COMMAND =READQ TS=
echo INW0027I- มีข้อผิดพลาดในการอ่าน TS Queue
goto exit1

:show7
echo INW0021I-เกิดข้อผิดพลาดในการลบ TS Queue
goto exit1

:show8
rem  INW0027I- ERROR IN THE CICS COMMAND =WRITEQ TS=
echo INW0027I- มีข้อผิดพลาดในการเขียนข้อมูลลง TS Queue

goto exit1

:show9
rem  INW0027I- ERROR OPENING THE CICS FILE - RDBUPD
echo INW0027I- มีข้อผิดพลาดในการใช้แฟ้มข้อมูล
goto exit1

:show10
rem  INW0027I- ERROR IN THE CICS COMMAND =ASKTIME= DETECTED
echo INW0027I- มีข้อผิดพลาดเกี่ยวกับคำสั่ง CICS ในการตรวจสอบเวลา
goto exit1

:show11
rem  INW0027I- ERROR IN THE CICS COMMAND =FORMATTIME= DETECTED
echo INW0027I- มีข้อผิดพลาดเกี่ยวกับการ format ของเวลา
```

```
goto exit1

:show12
rem  INW0027I- CFTRRB02 IS NOT STARTED FROM A RECEIVE COMMAND
echo INW0027I- ส่งผ่านข้อมูลไม่ได้มาจากคำสั่ง receive
goto exit1

:exit1
echo โปรดติดต่อเจ้าหน้าที่ที่ดูแลระบบ. เมื่อเสร็จเรียบร้อยแล้วสามารถรับ
echo File จาก MF ใหม่ โดยเลือกจากเมนูหลัก
goto exit0

:exit0
echo กด Key ใดๆเพื่อทำงานต่อ...
pause > nul
call c:\pc3270\dummy > nul
cls
rem echo กด Key ใดๆเพื่อทำงานต่ออีกครั้ง...
rem pause > nul
```

```
*****************************************************************************
*  batch file name : mess.bat
*    funtion          : use in script file PCSEN.PCS
*                       in order to show error from LMF
*****************************************************************************
@echo off
cls
if %1 == 0000 goto show0000
if %1 == 9000 goto show9000
if %1 == 9001 goto show9001
if %1 == 9002 goto show9002
if %1 == 9003 goto show9003
if %1 == 9004 goto show9004
if %1 == 9005 goto show9005
if %1 == 9006 goto show9006
if %1 == 9007 goto show9007
if %1 == 9008 goto show9008
if %1 == 9009 goto show9009
if %1 == 9010 goto show9010
if %1 == 9011 goto show9011
if %1 == 9012 goto show9012
goto exit

:show0000
echo CFTRSEN-0000
echo ส่งข้อมูลขึ้นเรียบร้อยแล้ว
del c:\thesis\tran\send_err
goto exit0
```

```
:show9000
echo CFTRSEN-9000
echo เกิดข้อผิดพลาดที่ TS Queue
goto exit1

:show9001
echo CFTRSEN-9001
echo ความยาวของ Record ไม่ถูกต้อง
goto exit1

:show9002
echo CFTRSEN-9002
echo ไม่พบ TS Queue บน LMF
goto exit1

:show9003
echo CFTRSEN-9003
echo ไม่มี Header Queue ใน TS Queue
goto exit1

:show9004
echo CFTRSEN-9004
echo ไม่พบแฟ้มข้อมูล ใน LMF
goto exit1

:show9005
echo CFTRSEN-9005
```

```
echo เงื่อนไข Input/Output จากแฟ้มข้อมูลเกิดข้อผิดพลาด
goto exit1

:show9006
echo CFTRSEN-9006
echo เงื่อนไข Input/Output จาก RDBUPD เกิดข้อผิดพลาด
goto exit1

:show9007
echo CFTRSEN-9007
echo ไม่มีชื่อแฟ้มข้อมูลของ Host ที่ทำการส่ง
goto exit1

:show9008
echo CFTRSEN-9008
echo ไม่มีที่รองรับในการติดต่อสื่อสารข้อมูล
goto exit1

:show9009
echo CFTRSEN-9009
echo เวลาประทับไม่ถูกต้อง มีการส่งข้อมูลซ้ำ
goto exit1

:show9010
echo CFTRSEN-9010
echo เกิดข้อผิดพลาดในส่วนของ ASKTIME
goto exit1
```

```
:show9011
echo CFTRSEN-9011
echo เกิดข้อผิดพลาดในส่วนของ Format Time
goto exit1

:show9012
echo CFTRSEN-9012
echo ไม่มีที่รองรับในการติดต่อสื่อสารข้อมูล
goto exit1

:exit1
echo โปรดติดต่อเจ้าหน้าที่ที่ดูแลระบบ. เมื่อเสร็จเรียบร้อยแล้วสามารถส่ง
echo File ขึ้น MF ใหม่ โดยเลือกจากเมนูหลัก
goto exit0

:exit0
echo กด Key ใดๆเพื่อทำงานต่อ...
pause > nul
call c:\pc3270\dummy > nul
rem echo กด Key ใดๆเพื่อทำงานต่อ...
rem pause > nul
```

```
@echo off
set CMGR.QUIET=-q
set CMGR.TRACE=00
:strt
if x%1 == x/v goto qt1
if x%1 == x/V goto qt1
if x%1 == x/t goto trc
if x%1 == x/T goto trc
if x%1 == x goto exit
echo CFG36 Parameter specified is incorrect
GOTO exit1
:qt1
set CMGR.QUIET=
shift
goto strt
:trc
set CMGR.TRACE=%2
shift
shift
goto strt
:exit
C:\PC3270\LOGO
C:\PC3270\TQDOS %CMGR.QUIET%
if ERRORLEVEL 1 GOTO exit1
C:\PC3270\PCSCCCP C:\PC3270\PCSCONV.TXT 0838 %CMGR.QUIET%
C:\PC3270\PCSDOS 2 a=24*80 b=24*80 -gs -ke -f=PCS874.FNT -ef %CMGR.QUIET%
C:\PC3270\PCSSNA B=02500  %CMGR.QUIET%
```

```
C:\PC3270\PCSTKR LU=*,* A=0 n=snalantrace1 G=10005A8ADF88 RS=265 LS=1 I=06100001
XS=0265 DS=04 T=%CMGR.TRACE% %CMGR.QUIET%
C:\PC3270\PCSBKEY -c %CMGR.QUIET% TH500.ENH
C:\PC3270\PCSXLATE %CMGR.QUIET% /838874
C:\PC3270\PCSLDTBL %CMGR.QUIET% /838874
C:\PC3270\PCSBKEY %CMGR.QUIET% C:\PC3270\3270DISP.TDF
C:\PC3270\PCSBKEY %CMGR.QUIET% C:\PC3270\PC3270.KDF
C:\PC3270\PCSTHAI %CMGR.QUIET%
C:\PC3270\PCSHLL k=04  c=838 %CMGR.QUIET%
rem YN 1,1,1,1,1,1,2,1,1,1,2,1,1,2,2,2,3,1
rem PC3270 Version 2.00
C:\PC3270\PCSJUMPH BATCH HOST DOS
c:\pc3270\pcslogon c:\pc3270\pcsen.pcs %CMGR.QUIET%
CLS
:exit1
```

```
@echo off
set CMGR.QUIET=-q
set CMGR.TRACE=00
:strt
if x%1 == x/v goto qt1
if x%1 == x/V goto qt1
if x%1 == x/t goto trc
if x%1 == x/T goto trc
if x%1 == x goto exit
echo CFG36 Parameter specified is incorrect
GOTO exit1
:qt1
set CMGR.QUIET=
shift
goto strt
:trc
set CMGR.TRACE=%2
shift
shift
goto strt
:exit
C:\PC3270\LOGO
C:\PC3270\TQDOS %CMGR.QUIET%
if ERRORLEVEL 1 GOTO exit1
C:\PC3270\PCSCCCP C:\PC3270\PCSCONV.TXT 0838 %CMGR.QUIET%
C:\PC3270\PCSDOS 2 a=24*80 b=24*80 -gs -ke -f=PCS874.FNT -ef %CMGR.QUIET%
C:\PC3270\PCSSNA B=02500  %CMGR.QUIET%
```

```
C:\PC3270\PCSTKR LU=*,* A=0 n=snalant-ace1 G=10005A8ADF88 RS=265 LS=1 I=06100001
XS=0265 DS=04 T=%CMGR.TRACE% %CMGR.QUIET%
C:\PC3270\PCSBKEY -c %CMGR.QUIET% TH500.ENH
C:\PC3270\PCSXLATE %CMGR.QUIET% /838874
C:\PC3270\PCSLDTBL %CMGR.QUIET% /838874
C:\PC3270\PCSBKEY %CMGR.QUIET% C:\PC3270\3270DISP.TDF
C:\PC3270\PCSBKEY %CMGR.QUIET% C:\PC3270\PC3270.KDF
C:\PC3270\PCSTHAI %CMGR.QUIET%
C:\PC3270\PCSHLL k=04  c=838 %CMGR.QUIET%
rem YN 1,1,1,1,1,1,2,1,1,1,2,1,1,2,2,2,3,1
rem PC3270 Version 2.00
C:\PC3270\PCSJUMPH BATCH HOST DOS
c:\pc3270\pcslogon c:\pc3270\pcrec.pcs %CMGR.QUIET%
CLS
:exit1
```

```
*********************************************************************************
*  batch file name : 1.bat                                                      *
*  funtion.        : call CU-writer                                             *
*********************************************************************************
@echo off
prompt $p$g
c:
cd\cw
cw
goto ended

:ended
cd\thesis
main
```

```
*****************************************************************************
*  batch file name : 2.bat                                                  *
*  funtion         : call FoxPro program                                    *
*****************************************************************************
@echo off
cd\
cls
PATH=C:\DOS;C:\thesis;C:\FOX
cd\thesis\programs
foxl -t enterprg
if exist c:\thesis\tran\sendfile goto send
if exist c:\thesis\tran\recfile  goto rec
goto end

:send
cd\pc3270
call pcsen.bat
goto end

:rec
cd\pc3270
call pcrec.bat

:end
cd\thesis
main
```

```
*****************************************************************************
*  batch file name : 3.bat                                                  *
*  funtion         : send PC file to Mainframe via PC3270                   *
*****************************************************************************
@Echo off
If not exist c:\thesis\tran\send_err. goto no_err
If not exist c:\thesis\tran\sendfile. goto no_file
goto re_send

:no_err
Echo +
Echo File ได้ถูกส่งขึ้น MF ไปเรียบร้อยแล้ว
Echo กด Key ใดๆเพื่อกลับไปเมนูหลัก.....
pause > nul
goto end

:no_file
Echo +
Echo ไม่มี File ที่จะส่งขึ้น MF
Echo กด Key ใดๆเพื่อกลับไปเมนูหลัก.....
pause > nul
goto end

:re_send
cd\pc3270
call pcsen.bat

:end
```

```
cd\thesis
main
```

```
*********************************************************************************
*  batch file name : 4.bat                                                      *
*  funtion         : receive file from Mainframe via PC3270                     *
*********************************************************************************
@Echo off
If not exist c:\thesis\tran\rec_err. goto no_err
If not exist c:\thesis\tran\recfile. goto no_file
goto re_rec

:no_err
Echo +
Echo File ถูกรับจาก MF ลงมาเรียบร้อยแล้ว
Echo กด Key ใดๆเพื่อกลับไปเมนูหลัก.....
pause > nul
goto end

:no_file
Echo +
Echo ไม่มี File ที่จะรับจาก MF
Echo กด Key ใดๆเพื่อกลับไปเมนูหลัก.....
pause > nul
goto end

:re_rec
cd\pc3270
call pcrec.bat

:end
```

```
cd\thesis
main
```

```
*******************************************************************************
*  batch file name : main.bat                                                 *
*  funtion         : main menu of PC-MAINFRAME file transfer.                 *
*                    Once user select option, it will execute                 *
*                    the appropriate batch file.                              *
*                    There are 5 options.                                     *
*                        (1)  run CU-writer                                   *
*                        (2)  trigger a transaction to send/receive file      *
*                             between PC and Mainframe                        *
*                        (3)  send file to Mainframe via PC3270               *
*                        (4)  receive file from Mainframe via PC3270          *
*                        (5)  exit to DOS                                     *
*******************************************************************************
@break off
@echo off
break = off
path=c:\dos;c:\thesis;c:\fox;c:\
call c:\thesis\crepro.bat
cd\thesis
cls

IF exist C:\PC3270\*.* GOTO HOST

TYPE menu.scr
GOTO input

:HOST
type menuspt.scr
```

```
goto input

:input
deprompt กดตัวเลขที่ต้องการ... 1 2 3 4 5
if errorlevel == 5 goto menu5
if errorlevel == 4 goto menu4
if errorlevel == 3 goto menu3
if errorlevel == 2 goto menu2
if errorlevel == 1 goto menu1
goto input

:menu1
1.bat
goto exit

:menu2
2.bat
goto exit

:menu3
3.bat
goto exit

:menu4
4.bat
goto exit

:menu5
```

```
goto exit_dos



:exit
cd\thesis
main

:exit_dos
prompt $p$g
cd\
call setpath/
```

```
*****************************************************************************
*  batch file name : del_file.bat
*     function     : delete all file in working area (c:\thesis\tran)
*                    in case of cancel unsuccessful transaction such as
*                    can not connect to HOST because the communication
*                    line is down so user can not send/receive file
*                    to/from mainframe.
*****************************************************************************
@echo off
If not exist c:\thesis\tran\*.*  goto exit
Echo Y| del c:\thesis\tran

:exit
```

```
*****************************************************************************
*
*   Table name  :  LISTTAB.DBF
*
*     function    :  This database keeps PC-MAINFRAME filename, owner (userid) and
*                    last updated date
*   Table name  :  USERID.DBF
*
*   function      :   This database keeps userid that we allowed to use PC-MAINFRAME
*
*                     file transfer.  The userid must be unique
*
*****************************************************************************
*
```

Structure for database : C:\THESIS\FILES\LISTTAB.DBF

Number of data records : 6

Date of last update : 01/28/94

Field	Field Name	Type	Width	Dec	Index
1	USERID	Character	4		
2	PC_FILE	Character	12		
3	UPD_DATE	Date	8		
4	MF_FILE	Character	8		
** Total **			33		

Structure for database : C:\THESIS\FILES\USERID.DBF

Number of data records : 3

Date of last update : 02/04/94

Field	Field Name	Type	Width	Dec	Index
1	USERID	Character	4		Asc
** Total **			5		

```
IDENTIFICATION DIVISION.
PROGRAM-ID.    CFTRSEN.

*****************************************************************
*                                                               *
* CFTRSEN - THIS PROGRAM IS USED IN THE TRANSFER FILE PROCEDURE *
*          TO UPLOAD THE USER'S PC FILE (CU WRITER) TO THE      *
*          HOST VSAM FILE. THE PROGRAM USE HOST FILE NAME,      *
*          USER-ID, TRMID TO BE KEY. BEFORE KEEPING A FILE      *
*          FROM PC, THE PROGRAM WILL CHECK THE KEY IN THE       *
*          UPLOADED FILE AGAINST THE EXISTING KEY IN THE HIST   *
*          FILE. IF THE KEY ARE THE SAME THEN DELETE THE OLD    *
*          DATA IN THE DATA FILE, WRITE NEW DATA IN THE DATA    *
*          FILE AND UPDATE THE HISTORY FILE. IF THE KEY ARE     *
*          NOT THE SAME, THE PROGRAM WILL WRITE NEW DATA IN     *
*          THE DATA AND HISTORY FILE.                           *
*                                                               *
*          A DESCRIPTION OF THE STANDARDS FOR FILE TRANSFER     *
*          CAN BE FOUND IN THE FOLLOWING VSE MANUAL.            *
*          >> VSE/ESA ADMINISTRATION                            *
*          >>        CHAPTER 11.  USING WORKSTATION FILE        *
*          >>        TRANSFER INTERFACES AND FUNCTIONS.         *
*          >>        - FILE TRANSFER TO AND FROM CICS/VSE       *
*          >>          TEMPORARY STORAGE                        *
*                                                               *
*          THIS PROGRAM USES A SET OF CICS COMMANDS, WHICH      *
*          HAVE BEEN MADE SUBJECT FOR SECURITY CHECKING.        *
*          A DESCRIPTION OF THE SECURITY FOR THESE CICS         *
*          COMMANDS CAN BE FOUND IN THE CICS MANUALS.           *
*          >> CICS/VSE SYSTEM PROGRAMMING REFERENCE             *
*          >>        CHAPTER 1. INTRUDUCTION                    *
*
```

```
*                                                               *
* VERSION 1.0.0                                                 *
*                                                               *
*****************************************************************

 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*****************************************************************
*                                                               *
*   DEFINITION OF THE COMMUNICATION AREA                        *
*     (LAYOUT FROM VSE/ESA MANUAL)                              *
*****************************************************************
 01  WS-COMMAREA-G.
       05 FILLER                PIC XX.
       05 FILLER                PIC XX.
       05 COMM-UFILE-X          PIC X(8).
       05 COMM-UQUEUE-X         PIC X(8).
       05 FILLER  REDEFINES COMM-UQUEUE-X.
         07 COMM-CFTR-X         PIC X(4).
         07 COMM-TRMID-X        PIC X(4).
       05 FILLER                PIC X(8).
*
*****************************************************************
*                                                               *
*   DEFINITION OF THE TS QUEUE HEADER RECORD                    *
*     (LAYOUT FROM VSE/ESA MANUAL)                              *
*****************************************************************
 01  TS-HDR-RECORD-G.
       05 TS-HDR-FILE-X         PIC X(8).
       05 FILLER                PIC X.
       05 TS-HDR-TIME-X         PIC X(6).
       05 FILLER                PIC X.
```

```
           05 TS-HDR-DATE-X          PIC X(5).
           05 FILLER                 PIC X.
           05 TS-HDR-LPAR-X          PIC X.
           05 TS-HDR-OPT1-X          PIC X(6).
           05 FILLER                 PIC X.
           05 TS-HDR-OPT2-X          PIC X(6).
           05 FILLER                 PIC X.
           05 TS-HDR-TYPE-X          PIC X(8).
           05 FILLER                 PIC X.
           05 TS-HDR-DCDF-X          PIC X(8).
           05 FILLER                 PIC X.
           05 TS-HDR-USID-X          PIC X(8).
           05 FILLER                 PIC X.
           05 TS-HDR-RESERVD-X       PIC X(28).
           05 FILLER                 PIC X.
           05 TS-HDR-RPAR-X          PIC X.
           05 TS-HDR-COMM-X          PIC X(66).

*****************************************************************
*                                                               *
*    DEFINITION OF THE WORKING FOR HIST RECORD                  *
*****************************************************************
 01  WS-HIST-RECORD-G.
     02 WS-HIST-KEY.
        05 WS-HIST-APPNAME-X          PIC X(8).
        05 WS-HIST-USERID-X           PIC X(4).
        05 WS-HIST-TRMID-X            PIC X(4).
     02 WS-HIST-NUM-B                 PIC S9(4)  COMP.
     02 WS-HIST-APP-X                 PIC X(10).

*****************************************************************
*    DEFINITION OF THE WORKING FOR DATA  RECORD                 *
*****************************************************************
```

```
01  WS-DATA-RECORD-G.
    02 WS-DATA-KEY.
       05 WS-DATA-KEY-B.
          10 WS-DATA-APPNAME-X          PIC X(8).
          10 WS-DATA-USERID-X           PIC X(4).
          10 WS-DATA-TRMID-X            PIC X(4).
       05 WS-DATA-SEQNO-B               PIC S9(4)   COMP.
    02 WS-DATA-DETAIL-X                 PIC X(200).

*****************************************************************
*                                                               *
*   DEFINITIONS FOR CHECKING THE TS QUEUE                       *
*                                                               *
*****************************************************************
 77  CHK-CONTROL-HIST-X            PIC X  VALUE 'W'.
         88 SWITCH-WRITE-HIST             VALUE 'W'.
         88 SWITCH-UPDATE-HIST            VALUE 'U'.
*
*****************************************************************
*                                                               *
*   DEFINITIONS USED FOR INPUT/OUTPUT                           *
*                                                               *
*****************************************************************
 77  IO-QUEUE-EOF-X                PIC X  VALUE '0'.
         88 NOT-EOF-QUEUE                 VALUE '0'.
         88 EOF-QUEUE                     VALUE '1'.
*
 77  IO-QUEUE-X                    PIC X  VALUE '0'.
         88 NO-QUEUE-FOUND                VALUE '0'.
         88 QUEUE-FOUND                   VALUE '1'.

 77  IO-EOF-DATA-X                 PIC X  VALUE '0'.
         88 NOT-EOF-DATA                  VALUE '0'.
```

```
        88 EOF-DATA                  VALUE '1'.

*
 01  IO-ASCIIREC-G.
        05 FILLER                PIC X(200).
*
 77  IO-FIRST-B                  PIC S9(4) COMP VALUE +1.
 77  IO-RECLEN-B                 PIC S9(4) COMP VALUE +220.
 77  IO-RECRBA-B                 PIC S9(8) COMP.
 77  IO-RESPONSE-B               PIC S9(8) COMP.
*
 77  C-HIST-X                   PIC X(6) VALUE 'PF002L'.
 77  C-DATA-X                    PIC X(6) VALUE 'PF001L'.
*
*****************************************************************
*                                                               *
*      PROGRAM MESSAGES                                         *
*                                                               *
*****************************************************************
 77  WS-MSGAREA-X                PIC X(80).

 01  WS-PROGRAM-MSG-G.
        02 MSG-TRANSFER-OK-X       PIC X(45) VALUE
          'CFTRSEND-0000 TRANSFER COMPLETED SUCCESSFULLY'.

        02 MSG-TS-ERROR-X          PIC X(44) VALUE
          'CFTRSEND-9000 UNEXPECTED ERROR FROM TS QUEUE'.

        02 MSG-TS-LENGTHERR-X      PIC X(39) VALUE
          'CFTRSEND-9001 ERROR IN TS RECORD LENGTH'.

        02 MSG-TS-QIDERR-X         PIC X(47) VALUE
          'CFTRSEND-9002 TEMPORARY STORAGE QUEUE NOT FOUND'.
```

```
02 MSG-TS-HDR-ITEMERR-X     PIC X(48) VALUE
  'CFTRSEND-9003 NO QUEUE HEADER RECORD IN TS QUEUE'.

02 MSG-FILENOTFND-X         PIC X(33) VALUE
  'CFTRSEND-9004 CICS FILE NOT FOUND'.


02 MSG-FILE-IOERR-X         PIC X(50) VALUE
  'CFTRSEND-9005 IOERR CONDITION RETURN FROM CICSFILE'.

02 MSG-FILE-ERROR-X         PIC X(47) VALUE
  'CFTRSEND-9006 UNEXPECTED ERROR FROM CICS FILE  '.

02 MSG-TS-NOFILENAME-X      PIC X(39) VALUE
  'CFTRSEND-9007 NO HOST FILE NAME IN SEND'.

02 MSG-NO-COMMAREA-X        PIC X(34) VALUE
  'CFTRSEND-9008 NO COMMAREA RECEIVED'.

02 MSG-ERR-START-X          PIC X(37) VALUE
  'CFTRSEND-9009 START DATA RECORD ERROR'.

02 MSG-ERR-READNEXT-X       PIC X(34) VALUE
  'CFTRSEND-9010 READ NEXT DATA ERROR'.

02 MSG-ERR-END-X            PIC X(31) VALUE
  'CFTRSEND-9011 END POINTER ERROR'.

02 MSG-ERR-DELETE-X         PIC X(33) VALUE
  'CFTRSEND-9012 DELETE DATA RECORD ERROR'.
/

*****************************************************************
```

```
*                                                               *
*      COMM AREA                                                *
*                                                               *
*****************************************************************
 LINKAGE SECTION.
 01  DFHCOMMAREA                 PIC X(28).
*
*****************************************************************
*                                                               *
*    START OF EXECUTION                                         *
*                                                               *
*****************************************************  *********
 PROCEDURE DIVISION.
*
*****************************************************************
*    CHECK IF A COMMAREA WAS PASSED                             *
*    IF NOT, ISSUE A MESSAGE TO THE SCREEN AND TERMINATE        *
*****************************************************************
    IF EIBCALEN = 0
      MOVE MSG-NO-COMMAREA-X    TO WS-MSGAREA-X
      PERFORM DISPLAY-MESSAGE
    ELSE
      MOVE DFHCOMMAREA          TO WS-COMMAREA-G.

*****************************************************************
*    INITIALIZE  AND  OPEN CICS FILE HIST AND DATA              *
*    READ FIRST RECORD IN TS QUEUE FOR UPDATE OR WRITE          *
*    HIST RECORD.                                               *
*****************************************************************
    MOVE SPACE  TO WS-HIST-KEY  WS-HIST-APP-X
                   WS-DATA-APPNAME-X  WS-DATA-USERID-X
                   WS-DATA-TRMID-X    WS-DATA-DETAIL-X.
    MOVE ZEROS  TO WS-HIST-NUM-B  WS-DATA-SEQNO-B.
```

```
    MOVE '0'    TO IO-QUEUE-EOF-X.
    MOVE 'W'    TO CHK-CONTROL-HIST-X.
    MOVE '0'    TO IO-EOF-DATA-X.
    MOVE '0'    TO IO-QUEUE-X.

    PERFORM READ-TS-QUEUE-HEADER.
    PERFORM GET-KEY-HIST-FILE.

****************************************************************
*  WRITE THE TS QUEUE RECORDS TO THE CICS FILE (DATAFILE)      *
*     - READ A TS QUEUE RECORD                                 *
*     - LOOP UNTIL NO MORE RECORDS IN THE TS QUEUE             *
*        - WRITE THE TS QUEUE TO THE DATAFILE                  *
*        - READ NEXT TS QUEUE RECORD                           *
*     - END LOOP                                               *
****************************************************************
    PERFORM OPEN-CICS-DATA.
    PERFORM READ-TS-NEXT.

    PERFORM LOOP-WRITE-DATA-RECORD
          UNTIL EOF-QUEUE.
*
****************************************************************
*    UPDATE OR WRITE  HIST RECORD                              *
*    -  TO UPDATE NUMBER RECORDS OF DATA  TO HIST              *
*    -  OR WRITE NEW HIST RECORD                               *
****************************************************************
    IF SWITCH-WRITE-HIST
      PERFORM WRITE-HIST-RECORD
    ELSE
      PERFORM REWRITE-HIST-RECORD.

****************************************************************
```

```
*    CLEAN UP                                          *
*    - UNLOCK THE CICS FILE AFTER THE MASSINSERT        *
*    - CLOSE THE CICS FILE                              *
*    - ISSUE OK MESSAGE                                 *
*****************************************************************
    PERFORM UNLOCK-HIST-DATA.

    PERFORM CLOSE-CICS-DATA.
    PERFORM CLOSE-CICS-HIST.

    MOVE MSG-TRANSFER-OK-X      TO WS-MSGAREA-X.
    PERFORM DISPLAY-MESSAGE.
*
    GOBACK.
/
*****************************************************************
*    PROGRAM SECTIONS                                   *
*****************************************************************
*****************************************************************
* WRITE A MESSAGE TO THE SCREEN AND TERMINATE THE PROGRAM      *
*                                                       *
*    - THE TS QUEUE IS DELETED BEFORE THE PROGRAMS TERMINATES  *
*    - WRITE THE MESSAGE TO THE SCREEN                  *
*                                                       *
*                                                       *
*****************************************************************
 DISPLAY-MESSAGE SECTION.
*
    IF QUEUE-FOUND
      EXEC CICS
          DELETEQ TS QUEUE(COMM-UQUEUE-X) NOHANDLE
      END-EXEC.
```

```
    EXEC CICS
        SEND FROM(WS-MSGAREA-X)
            LENGTH(80) NOHANDLE
            ERASE WAIT
    END-EXEC.
*
*****************************************************************
*   A RECEIVE IS ISSUED TO ENSURE THAT THE MESSAGE IS NOT       *
*   ERASED. THE USER MUST PRESS ENTER TO ACKNOWLEDGE THE MSG    *
*****************************************************************
    EXEC CICS
        RECEIVE INTO(IO-ASCIIREC-G)
              LENGTH(IO-RECLEN-B)
              NOHANDLE
    END-EXEC.
*
*****************************************************************
*   ISSUE A SEND WITH LENGTH 0 TO UNLOCK THE KEYBOARD           *
*****************************************************************
    EXEC CICS
        SEND FROM(WS-MSGAREA-X)
            LENGTH(0)
            ERASE
            NOHANDLE
    END-EXEC.
*
*****************************************************************
*   RETURN TO CICS                                              *
*****************************************************************
    EXEC CICS RETURN END-EXEC.
*
 DISPLAY-MESSAGE-EXIT.
    EXIT.
```

```
/
*****************************************************************
* READ THE FIRST TS QUEUE RECORD                                *
* THE FIRST RECORD IN THE TS QUEUE IS ALWAYS A QUEUE HEADER     *
* RECORD.                                                       *
*****************************************************************
 READ-TS-QUEUE-HEADER SECTION.
*    MOVE LENGTH OF TS-HDR-RECORD-G TO IO-RECLEN-B.
     MOVE 160 TO IO-RECLEN-B.
     EXEC CICS
          READQ TS QUEUE(COMM-UQUEUE-X)
                INTO(TS-HDR-RECORD-G)
                LENGTH(IO-RECLEN-B)
                ITEM(IO-FIRST-B)
                RESP(IO-RESPONSE-B)
     END-EXEC.
*
       IF IO-RESPONSE-B = DFHRESP(NORMAL)
           IF TS-HDR-FILE-X NOT = ' '
             MOVE TS-HDR-FILE-X   TO WS-HIST-APPNAME-X
             MOVE TS-HDR-USID-X   TO WS-HIST-USERID-X
             MOVE COMM-TRMID-X    TO WS-HIST-TRMID-X
             MOVE '1'   TO IO-QUEUE-X
           ELSE
             MOVE MSG-TS-NOFILENAME-X  TO WS-MSGAREA-X
             PERFORM DISPLAY-MESSAGE
       ELSE
       IF IO-RESPONSE-B = DFHRESP(LENGERR)
           MOVE MSG-TS-LENGTHERR-X     TO WS-MSGAREA-X
           PERFORM DISPLAY-MESSAGE
       ELSE
       IF IO-RESPONSE-B = DFHRESP(QIDERR)
           MOVE MSG-TS-QIDERR-X        TO WS-MSGAREA-X
```

```
            PERFORM DISPLAY-MESSAGE
        ELSE
        IF IO-RESPONSE-B = DFHRESP(ITEMERR)
            MOVE MSG-TS-HDR-ITEMERR-X   TO WS-MSGAREA-X
            PERFORM DISPLAY-MESSAGE
        ELSE
            MOVE MSG-TS-ERROR-X         TO WS-MSGAREA-X
            PERFORM DISPLAY-MESSAGE.

*
 READ-TS-QUEUE-HEADER-EXIT.
     EXIT.
/
*****************************************************************
* READ THE KEY FROM CICS FILE HIST TO CHECK THE KEY IS ALREADY *
* EXIST.                                          *
*****************************************************************
 GET-KEY-HIST-FILE SECTION.
*
    EXEC CICS
        SET DATASET(C-HIST-X)
            OPEN
            RESP(IO-RESPONSE-B)
    END-EXEC.
*
        IF IO-RESPONSE-B = DFHRESP(NORMAL)
            NEXT SENTENCE
        ELSE
        IF IO-RESPONSE-B = DFHRESP(IOERR)
            MOVE MSG-FILE-IOERR-X       TO WS-MSGAREA-X
            PERFORM DISPLAY-MESSAGE
        ELSE
            MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
```

```
           PERFORM DISPLAY-MESSAGE.
*
*     MOVE LENGTH OF WS-HIST-RECORD-G TO IO-RECLEN-B.
      MOVE 30 TO IO-RECLEN-B.
     MOVE +0                    TO IO-RECRBA-B.

     EXEC CICS
         READ DATASET(C-HIST-X)
             INTO(WS-HIST-RECORD-G)
             RIDFLD(WS-HIST-KEY)
             UPDATE
             RESP(IO-RESPONSE-B)
     END-EXEC.

       IF IO-RESPONSE-B = DFHRESP(NORMAL)
           MOVE 'U'  TO CHK-CONTROL-HIST-X
           PERFORM DELETE-OLD-DATA
       ELSE
           MOVE 'W'  TO CHK-CONTROL-HIST-X.
 GET-KEY-HIST-FILE-EXIT.
     EXIT.
/
*****************************************************************
*  LOOP DELETE OLD RECORDS FROM CICS DATA FILE                  *
*     - READ FIRST DATA RECORD FROM CICS FILE                   *
*     - LOOP UNTIL NO MORE RECORDS OR CHANGED KEY               *
*        - DELETE DATA RECORDS                                  *
*        - READ NEXT CICS DATA RECORD                           *
*     - END LOOP                                                *
*****************************************************************
 DELETE-OLD-DATA SECTION.
*
     PERFORM OPEN-CICS-DATA.
```

```
    PERFORM STARTBR-CICS-DATA.
    PERFORM READNEXT-CICS-DATA.

    PERFORM LOOP-DELETE-OLD-DATA
          UNTIL EOF-DATA  OR WS-DATA-KEY-B NOT = WS-HIST-KEY.

    PERFORM ENDBR-CICS-DATA.
    PERFORM CLOSE-CICS-DATA.
 DELETE-OLD-DATA-EXIT.
    EXIT.
/
 LOOP-DELETE-OLD-DATA SECTION.
    PERFORM DELETE-CICS-DATA-RECORD
    PERFORM READNEXT-CICS-DATA.
 DELETE-OLD-DATA-EXIT.
    EXIT.
/
*****************************************************************
* READ THE NEXT TS QUEUE RECORD                                 *
* IF ANY ERRORS WHILE READING THE RECORD, THE PROGRAM ISSUES A  *
* MESSAGE TO THE SCREEN AND TERMINATES.                         *
*****************************************************************
 READ-TS-NEXT SECTION.
*
*     MOVE LENGTH OF IO-ASCIIREC-G    TO IO-RECLEN-B.
      MOVE 200 TO IO-RECLEN-B.
    EXEC CICS
        READQ TS QUEUE(COMM-UQUEUE-X)
              INTO(IO-ASCIIREC-G)
              LENGTH(IO-RECLEN-B)
              NEXT
              RESP(IO-RESPONSE-B)
    END-EXEC.
```

```
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
         MOVE IO-ASCIIREC-G   TO WS-DATA-DETAIL-X
         ADD +1  TO WS-DATA-SEQNC-B
      ELSE
      IF IO-RESPONSE-B = DFHRESP(ITEMERR)
         MOVE '1'  TO  IO-QUEUE-EOF-X
      ELSE
      IF IO-RESPONSE-B = DFHRESP(LENGERR)
         MOVE MSG-TS-LENGTHERR-X     TO WS-MSGAREA-X
         PERFORM DISPLAY-MESSAGE
      ELSE
         MOVE MSG-TS-ERROR-X         TO WS-MSGAREA-X
         PERFORM DISPLAY-MESSAGE.
 READ-TS-NEXT-EXIT.
    EXIT.
/
*****************************************************************
*    WRITE  CICS DATA RECORD FROM TS QUEUE.                     *
*****************************************************************
 LOOP-WRITE-DATA-RECORD SECTION.
    PERFORM WRITE-DATA-RECORD.
    PERFORM READ-TS-NEXT.
 LOOP-WRITE-DATA-RECORD-EXIT.
    EXIT.
/

*****************************************************************
* PREPARE THE CICS FILE FOR UPDATE                              *
*****************************************************************
 OPEN-CICS-DATA SECTION.
*
    EXEC CICS
```

```
           SET DATASET(C-DATA-X)
               OPEN
               RESP(IO-RESPONSE-B)
       END-EXEC.
*
         IF IO-RESPONSE-B = DFHRESP(NORMAL)
             NEXT SENTENCE
         ELSE
         IF IO-RESPONSE-B = DFHRESP(IOERR)
             MOVE MSG-FILE-IOERR-X       TO WS-MSGAREA-X
             PERFORM DISPLAY-MESSAGE
         ELSE
             MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
             PERFORM DISPLAY-MESSAGE.
 OPEN-CICS-DATA-EXIT.
       EXIT.
/
*****************************************************************
* WRITE A RECORD TO THE CICS DATA FILE                          *
* IF ANY ERRORS WHILE WRITING THE RECORD, THE PROGRAM ISSUES A  *
* MESSAGE TO THE SCREEN AND TERMINATES.                         *
*****************************************************************
 WRITE-DATA-RECORD SECTION.
*
*    MOVE LENGTH OF WS-DATA-RECORD-G TO IO-RECLEN-B.
       MOVE 220 TO IO-RECLEN-B.
       MOVE WS-HIST-APPNAME-X  TO WS-DATA-APPNAME-X.
       MOVE WS-HIST-USERID-X   TO WS-DATA-USERID-X.
       MOVE WS-HIST-TRMID-X    TO WS-DATA-TRMID-X.

       EXEC CICS
           WRITE DATASET(C-DATA-X)
           FROM(WS-DATA-RECORD-G)
```

```
           RIDFLD(WS-DATA-KEY)
           MASSINSERT
           RESP(IO-RESPONSE-B)
       END-EXEC.
*
         IF IO-RESPONSE-B = DFHRESP(NCRMAL)
             NEXT SENTENCE
         ELSE
         IF IO-RESPONSE-B = DFHRESP(IOERR)
             MOVE MSG-FILE-IOERR-X       TO WS-MSGAREA-X
             PERFORM DISPLAY-MESSAGE
         ELSE
             MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
             PERFORM DISPLAY-MESSAGE.
 WRITE-DATA-RECORD-EXIT.
       EXIT.
/
*
****************************************************************
* WRITE A RECORD TO THE CICS HIST FILE                         *
* IF ANY ERRORS WHILE WRITING THE RECORD, THE PROGRAM ISSUES A *
* MESSAGE TO THE SCREEN AND TERMINATES.                        *
****************************************************************
 WRITE-HIST-RECORD SECTION.
*
*    MOVE LENGTH OF WS-HIST-RECORD-G TO IO-RECLEN-B.
       MOVE 30 TO IO-RECLEN-B.
       MOVE WS-DATA-SEQNO-B     TO WS-HIST-NUM-B.

       EXEC CICS
           WRITE DATASET(C-HIST-X)
           FROM(WS-HIST-RECORD-G)
           RIDFLD(WS-HIST-KEY)
```

```
           RESP(IO-RESPONSE-B)
       END-EXEC.
*
         IF IO-RESPONSE-B = DFHRESP(NORMAL)
             NEXT SENTENCE
         ELSE
         IF IO-RESPONSE-B = DFHRESP(IOERR)
             MOVE MSG-FILE-IOERR-X       TO WS-MSGAREA-X
             PERFORM DISPLAY-MESSAGE
         ELSE
             MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
             PERFORM DISPLAY-MESSAGE.
 WRITE-HIST-RECORD-EXIT.
       EXIT.
/
*
*****************************************************************
* REWRITE RECORD TO THE CICS HIST FILE                          *
* THE MASSINSERT OPTION IS USED IN CICS COMMAND FOR PERFORMANCE *
* REASONS ONLY.                                                 *
* IF ANY ERRORS WHILE WRITING THE RECORD, THE PROGRAM ISSUES A  *
* MESSAGE TO THE SCREEN AND TERMINATES.                         *
*****************************************************************
 REWRITE-HIST-RECORD SECTION.
*
*    MOVE LENGTH OF WS-HIST-RECORD-G TO IO-RECLEN-B.
       MOVE 30 TO IO-RECLEN-B.
       MOVE WS-DATA-SEQNO-B     TO WS-HIST-NUM-B.

       EXEC CICS
           REWRITE DATASET(C-HIST-X)
           FROM(WS-HIST-RECORD-G)
           RESP(IO-RESPONSE-B)
```

```
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
      IF IO-RESPONSE-B = DFHRESP(IOERR)
          MOVE MSG-FILE-IOERR-X       TO WS-MSGAREA-X
          PERFORM DISPLAY-MESSAGE
      ELSE
          MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
          PERFORM DISPLAY-MESSAGE
 REWRITE-HIST-RECORD-EXIT.
    EXIT.
/
*****************************************************************
* DECLARE STARTBR FOR THE CICS FILE FOR A SEQUENTIAL READ       *
*****************************************************************
 STARTBR-CICS-DATA SECTION.
*
    MOVE WS-HIST-APPNAME-X  TO WS-DATA-APPNAME-X.
    MOVE WS-HIST-USERID-X   TO WS-DATA-USERID-X.
    MOVE WS-HIST-TRMID-X    TO WS-DATA-TRMID-X.

    EXEC CICS
        STARTBR  DATASET(C-DATA-X)
                 RIDFLD(WS-DATA-KEY)
                 GTEQ
                 RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
```

```
           MOVE MSG-ERR-START-X       TO WS-MSGAREA-X
           PERFORM DISPLAY-MESSAGE.
 STARTBR-CICS-DATA-EXIT.
     EXIT.
/
*****************************************************************
* READNEXT TO GET A RECORD FROM THE CICS FILE                   *
*****************************************************************
 READNEXT-CICS-DATA SECTION.
*
*    MOVE LENGTH OF WS-DATA-RECORD-G TO IO-RECLEN-B.
     MOVE 220 TO IO-RECLEN-B.
     EXEC CICS
         READNEXT DATASET(C-DATA-X)
                  INTO(WS-DATA-RECORD-G)
                  RIDFLD(WS-DATA-KEY)
                  RESP(IO-RESPONSE-B)
     END-EXEC.
*
       IF IO-RESPONSE-B = DFHRESP(NORMAL)
           NEXT SENTENCE
       ELSE
       IF IO-RESPONSE-B = DFHRESP(ENDFILE)
           MOVE '1' TO  IO-EOF-DATA-X
       ELSE
           MOVE MSG-ERR-READNEXT-X     TO WS-MSGAREA-X
           PERFORM DISPLAY-MESSAGE.
 READNEXT-CICS-DATA-EXIT.
     EXIT.
/
*****************************************************************
* DELETE CICS OLD DATA RECORD                                   *
*****************************************************************
```

```
    DELETE-CICS-DATA-RECORD SECTION.
*
    EXEC CICS
        DELETE DATASET(C-DATA-X)
        RIDFLD(WS-DATA-KEY)
        RESP(IO-RESPONSE-B)
    END-EXEC.

      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE MSG-ERR-DELETE-X     TO WS-MSGAREA-X
          PERFORM DISPLAY-MESSAGE.
 DELETE-CICS-DATA-RECORD-EXIT.
    EXIT.
*****************************************************************
* CLOSE THE STARTBR WHEN END OF FILE IS REACHED                 *
*****************************************************************
 ENDBR-CICS-DATA SECTION.
*
    EXEC CICS
        ENDBR  DATASET(C-DATA-X)
            RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE MSG-ERR-END-X        TO WS-MSGAREA-X
          PERFORM DISPLAY-MESSAGE.
 ENDBR-CICS-DATA-EXIT.
    EXIT.
/
```

```
*****************************************************************
*  UNLOCK THE CICS FILE AFTER THE MASSINSERT                    *
*****************************************************************
 UNLOCK-HIST-DATA SECTION.
*
    EXEC CICS
        UNLOCK DATASET(C-HIST-X)
            RESP(IO-RESPONSE-B)
    END-EXEC.

      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
          PERFORM DISPLAY-MESSAGE.


    EXEC CICS
        UNLOCK DATASET(C-DATA-X)
            RESP(IO-RESPONSE-B)
    END-EXEC.

      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
          PERFORM DISPLAY-MESSAGE.
 UNLOCK-HIST-DATA-EXIT.
    EXIT.
/
*****************************************************************
* CLOSE THE CICS FILE                                           *
*****************************************************************
```

```
 CLOSE-CICS-DATA SECTION.
*
    EXEC CICS
        SET DATASET(C-DATA-X)
            CLOSED
            RESP(IO-RESPONSE-B)
    END-EXEC.

      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
          PERFORM DISPLAY-MESSAGE.
 CLOSE-CICS-DATA-EXIT.
    EXIT.
/
 CLOSE-CICS-HIST SECTION.
*
    EXEC CICS
        SET DATASET(C-HIST-X)
            CLOSED
            RESP(IO-RESPONSE-B)
    END-EXEC.

      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE MSG-FILE-ERROR-X       TO WS-MSGAREA-X
          PERFORM DISPLAY-MESSAGE.
 CLOSE-CICS-HIST-EXIT.
    EXIT.
**********************************************************
******************** END OF PROGRAM ************************
```

```
IDENTIFICATION DIVISION.
PROGRAM-ID. CFTRREC.

*****************************************************************
*                                                               *
* CFTRREC  - THIS PROGRAM IS USED TO DOWNLOAD THE DATA FROM THE *
*          DATA FILE WHICH SPECIFIED KEY.                       *
*          THE PROGRAM READS DATA FROM A CICS FILE (DATA FILE)* *
*          AND WRITES THE RECORDS TO A TS QUEUE - CFTRXXXX -    *
*          WHERE XXXX = TERMINAL NAME                           *
*                                                               *
*          A DESCRIPTION OF THE STANDARDS FOR FILE TRANSFER     *
*          CAN BE FOUND IN THE FOLLOWING VSE MANUAL.            *
*          >> VSE/ESA ADMINISTRATION                            *
*          >>        CHAPTER 11. USING WORKSTATION FILE         *
*          >>        TRANSFER INTERFACES AND FUNCTIONS.         *
*          >>        - FILE TRANSFER TO AND FROM CICS/VSE       *
*          >>          TEMPORARY STORAGE                        *
*                                                               *
*          THIS PROGRAM USES A SET OF CICS COMMANDS, WHICH      *
*          HAVE BEEN MADE SUBJECT FOR SECURITY CHECKING.        *
*          A DESCRIPTION OF THE SECURITY FOR THESE CICS         *
*          COMMANDS CAN BE FOUND IN THE CICS MANUALS.           *
*          >> CICS/VSE SYSTEM PROGRAMMING REFERENCE             *
*          >>        CHAPTER 1. INTRUDUCTION                    *
*
*                                                               *
*                                                               *
* VERSION 1.0.0                                                 *
* modify :  11/03/95 : Check - If History file exist, continue  *
*                      process.                                 *
*****************************************************************
```

```
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*****************************************************************
*                                                               *
*   DEFINITION OF THE COMMUNICATION AREA                        *
*     (LAYOUT FROM VSE/ESA MANUAL)                              *
*                                                               *
*****************************************************************
01  WS-COMMAREA-G.
      05 COMM-ACTION-X          PIC X.
        88  COMM-ACTION-SEND        VALUE 'U'.
        88  COMM-ACTION-RECEIVE     VALUE 'D'.
      05 FILLER                 PIC X.
      05 COMM-RETURNCODE-B      PIC S9(4) COMP.
      05 COMM-UFILE-X           PIC X(8).
      05 COMM-UQUEUE-X          PIC X(8).
      05 FILLER  REDEFINES COMM-UQUEUE-X.
        07 COMM-CFTR-X          PIC X(4).
        07 COMM-TRMID-X         PIC X(4).
      05 FILLER                 PIC X(8).
      05 COMM-HEADER-RECORD-X   PIC X(160).
*
*****************************************************************
*                                                               *
*   DEFINITION OF THE TS QUEUE HEADER RECORD                    *
*     (LAYOUT FROM VSE/ESA MANUAL)                              *
*                                                               *
*****************************************************************
01  TS-HDR-RECORD-G.
      05 TS-HDR-FILE-X          PIC X(8).
      05 FILLER                 PIC X.
      05 TS-HDR-TIME-G.
```

```
           10 TS-HDR-TIME-H          PIC X(2).
           10 TS-HDR-TIME-M          PIC X(2).
           10 TS-HDR-TIME-S          PIC X(2).
        05 FILLER                    PIC X.
        05 TS-HDR-DATE-G.
           10 TS-HDR-DATE-Y          PIC X(2).
           10 TS-HDR-DATE-D          PIC X(3).
        05 FILLER                    PIC X(1).
        05 TS-HDR-LPAR-X             PIC X.
        05 TS-HDR-OPT1-X             PIC X(6).
        05 FILLER                    PIC X.
        05 TS-HDR-OPT2-X             PIC X(6).
        05 FILLER                    PIC X.
        05 TS-HDR-TYPE-X             PIC X(8).
        05 FILLER                    PIC X.
        05 TS-HDR-DCDF-X             PIC X(8).
        05 FILLER                    PIC X.
        05 TS-HDR-USID-X             PIC X(8).
        05 FILLER                    PIC X.
        05 TS-HDR-RESERVD-X          PIC X(28).
        05 FILLER                    PIC X.
        05 TS-HDR-RPAR-X             PIC X
        05 TS-HDR-COMM-X             PIC X(66).

*****************************************************************
*                                                               *
*    DEFINITION OF THE WORKING FOR HIST RECORD                  *
*****************************************************************
 01  WS-HIST-RECORD-G.
     02 WS-HIST-KEY.
        05 WS-HIST-APPNAME-X          PIC X(8).
        05 WS-HIST-USERID-X           PIC X(4).
        05 WS-HIST-TRMID-X            PIC X(4).
```

```
    02 WS-HIST-NUM-B                PIC S9(4)  COMP.
    02 WS-HIST-APP-X                PIC X(10).

*****************************************************************
*   DEFINITION OF THE WORKING FOR DATA  RECORD                  *
*****************************************************************
 01  WS-DATA-RECORD-G.
    02 WS-DATA-KEY.
       03 WS-DATA-INQUIRY-KEY.
          05 WS-DATA-APPNAME-X        PIC X(8).
          05 WS-DATA-USERID-X         PIC X(4).
          05 WS-DATA-TRMID-X          PIC X(4).
       03 WS-DATA-SEQNO-B             PIC S9(4)   COMP.
    02 WS-DATA-DETAIL-X               PIC X(200).

*****************************************************************
*                                                               *
*   ASKTIME/FORMATTIME FIELDS                                   *
*     USED TO UPDATE THE DATE AND TIME IN THE TS QUEUE HEADER   *
*                                                               *
*****************************************************************
 77  WS-ABSTIME-P                  PIC S9(15) COMP-3.

 01  WS-INQUIRY-KEY.
       05 WS-INQUIRY-APPNAME-X     PIC X(8).
       05 WS-INQUIRY-USERID-X      PIC X(4).
       05 WS-INQUIRY-TRMID-X       PIC X(4).

 01  WS-TSM-YYDDD-G.
       05 WS-TSM-YY-X              PIC X(2).
       05 FILLER                PIC X.
       05 WS-TSM-DDD-X             PIC X(3).
```

```
01  WS-TSM-TIME-G.
        05 WS-TSM-HH-X          PIC X(2).
        05 FILLER               PIC X.
        05 WS-TSM-MM-X          PIC X(2).
        05 FILLER               PIC X.
        05 WS-TSM-SS-X          PIC X(2).

*****************************************************************
*                                                               *
*      RETURN CODES                                             *
*                                                               *
*****************************************************************
 77  C-TRANSFER-OK-B            PIC S9(4) COMP VALUE +0.
 77  C-NO-COMMAREA-B            PIC S9(4) COMP VALUE +1.
 77  C-TRMID-ERROR-B            PIC S9(4) COMP VALUE +2.
 77  C-STARTBR-ERROR-B          PIC S9(4) COMP VALUE +3.
 77  C-READNEXT-ERROR-B         PIC S9(4) COMP VALUE +4.
 77  C-ENDBR-ERROR-B            PIC S9(4) COMP VALUE +5.
 77  C-READ-TS-ERROR-B          PIC S9(4) COMP VALUE +6.
 77  C-DELETE-TS-ERROR-B        PIC S9(4) COMP VALUE +7.
 77  C-WRITE-TS-ERROR-B         PIC S9(4) COMP VALUE +8.
 77  C-FILE-ERROR-B             PIC S9(4) COMP VALUE +9.
 77  C-ASKTIME-ERROR-B          PIC S9(4) COMP VALUE +10.
 77  C-FORMATTIME-ERROR-B       PIC S9(4) COMP VALUE +11.
 77  C-SEND-COMM-B              PIC S9(4) COMP VALUE +12.
 77  C-IO-ERROR-B               PIC S9(4) COMP VALUE +13.
 77  C-HISTREC-NOTFND-B         PIC S9(4) COMP VALUE +14.

*****************************************************************
*                                                               *
*   INPUT/OUTPUT DEFINITIONS                                    *
*                                                               *
*****************************************************************
```

```
 77  SW-FILE-EOF-X                PIC X  VALUE '0'.
         88 NOT-EOF-FILE                VALUE '0'.
         88 EOF-FILE                    VALUE '1'.
*
 77  SW-QUEUE-X                   PIC X  VALUE '0'.
         88 QUEUE-NOT-CREATED            VALUE '0'.
         88 QUEUE-CREATED                VALUE '1'.
*
 77  SW-OPEN-X                    PIC X  VALUE '0'.
         88 FILE-NOT-OPEN                VALUE '0'.
         88 FILE-OPEN                    VALUE '1'.

*
 01  IO-ASCIIREC-G.
         05 FILLER                PIC X(200).
*
 77  IO-RECNO-B                   PIC S9(4) COMP VALUE +0.
 77  IO-RECLEN-B                  PIC S9(4) COMP VALUE +220.
 77  IO-RECRBA-B                  PIC S9(8) COMP.
 77  IO-RESPONSE-B                PIC S9(8) COMP.
*
 77  C-ZERO-B                     PIC S9(4) COMP VALUE +0.
 77  C-ONE-B                      PIC S9(4) COMP VALUE +1.
*
 77  C-HIST-X                     PIC X(6) VALUE 'PF002L'.
 77  C-DATA-X                     PIC X(6) VALUE 'PF001L'.
/
*****************************************************************
*                                                               *
*      LINKAGE AREA -  DFHCOMMAREA                              *
*                                                               *
*****************************************************************
 LINKAGE SECTION.
```

```
 01  DFHCOMMAREA                PIC X(188).
*
*****************************************************************
*                                                               *
*   START OF EXECUTION                                          *
*                                                               *
*****************************************************************
 PROCEDURE DIVISION.
*
*****************************************************************
*   CHECK IF A COMMAREA WAS PASSED                              *
*   IF NOT, TERMINATE THE PROGRAM WITH AN ERROR-CODE            *
*****************************************************************
    IF EIBCALEN = 0
      MOVE C-NO-COMMAREA-B      TO COMM-RETURNCODE-B
      PERFORM TERMINATE-PROGRAM
    ELSE
      MOVE DFHCOMMAREA          TO WS-COMMAREA-G.

*****************************************************************
*   INITIALIZE THE RETURNCODE TO ZERO                           *
*****************************************************************
    MOVE C-ZERO-B               TO COMM-RETURNCODE-B.

*****************************************************************
*   CHECK IF THE PROGRAM IS ACTIVATE BY A RECEIVE COMMAND       *
*   IF NOT, TERMINATE THE PROGRAM WITH AN ERROR-CODE            *
*****************************************************************
    IF COMM-ACTION-SEND
      MOVE C-SEND-COMM-B        TO COMM-RETURNCODE-B
      PERFORM TERMINATE-PROGRAM.

*****************************************************************
```

```
*    CHECK IF THE STANDARD TS QUEUE NAME IS USED              *
*    IF NOT, TERMINATE THE PROGRAM WITH AN ERROR-CODE         *
****************************************************************
    IF  COMM-TRMID-X = EIBTRMID AND COMM-CFTR-X = 'CFTR'
        NEXT SENTENCE
    ELSE
      MOVE C-TRMID-ERROR-B      TO COMM-RETURNCODE-B
      PERFORM TERMINATE-PROGRAM.

****************************************************************
* READ THE KEY FROM CICS FILE HIST TO CHECK THE KEY IS EXIST.  *
* IF HIST EXIST,  CONTINUE THE PROCESS                         *
****************************************************************
* CHECK-HIST-KEY    SECTION.
*
    EXEC CICS
        SET DATASET(C-HIST-X)
           OPEN
           RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
      IF IO-RESPONSE-B = DFHRESP(IOERR)
          MOVE C-IO-ERROR-B       TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM
      ELSE
          MOVE C-FILE-ERROR-B     TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
*
*     MOVE LENGTH OF WS-HIST-RECORD-G TO IO-RECLEN-B.
      MOVE 30 TO IO-RECLEN-B.
```

```
      MOVE +0                    TO IO-RECRBA-B.

******* move comm header area to ts-hdr-record-g *****
******* and move comm area to hist key fields ********
     MOVE COMM-HEADER-RECORD-X    TO TS-HDR-RECORD-G.
     MOVE COMM-UFILE-X            TO WS-HIST-APPNAME-X.
     MOVE TS-HDR-USID-X           TO WS-HIST-USERID-X.
     MOVE COMM-TRMID-X            TO WS-HIST-TRMID-X.

     EXEC CICS
        READ DATASET(C-HIST-X)
            INTO(WS-HIST-RECORD-G)
            RIDFLD(WS-HIST-KEY)
            RESP(IO-RESPONSE-B)
     END-EXEC.

       IF IO-RESPONSE-B = DFHRESP(NORMAL)
          EXEC CICS
             SET DATASET(C-HIST-X)
             CLOSED NOHANDLE
          END-EXEC
       ELSE
         IF IO-RESPONSE-B = DFHRESP(NOTFND)
           MOVE C-HISTREC-NOTFND-B  TO COMM-RETURNCODE-B
           PERFORM  TERMINATE-PROGRAM
         ELSE
           MOVE C-FILE-ERROR-B      TC COMM-RETURNCODE-B
           PERFORM  TERMINATE-PROGRAM.

 CHECK-HIST-KEY-EXIT.
     EXIT.
```

```
*****************************************************************
*    CHECK IF THE TS QUEUE EXISTS                               *
*****************************************************************
    PERFORM CLEAR-OLD-QUEUE.

*****************************************************************
*    OPEN THE CICS FILE                                         *
*****************************************************************
    PERFORM OPEN-CICS-FILE.

*****************************************************************
*    WRITE THE TS QUEUE HEADER TO THE TS QUEUE                  *
*****************************************************************
    PERFORM CREATE-HEADER-RECORD.
    MOVE TS-HDR-RECORD-G         TO IO-ASCIIREC-G.
*    MOVE LENGTH OF TS-HDR-RECORD-G
*                              TO IO-RECLEN-B.
    MOVE 160 TO IO-RECLEN-B.
    MOVE +0                      TO IO-RECNO-B.
    EXEC CICS
        WRITEQ TS QUEUE(COMM-UQUEUE-X)
        FROM(IO-ASCIIREC-G)
        LENGTH(IO-RECLEN-B)
        ITEM(IO-RECNO-B)
        RESP(IO-RESPONSE-B)
    END-EXEC.
    MOVE '1'                     TO SW-QUEUE-X

*****************************************************************
* THE CICS FILE IS READ SEQUENTIALLY UNTIL END OF FILE.         *
* NO CHECKING OF THE DATA RECORD IN THE CICS FILE IS DONE       *
*                                                               *
*  - READ FIRST DATA RECORD FROM THE CICS FILE                  *
```

```
*  - LOOP UNTIL NO MORE CICS FILE RECORDS                    *
*     - WRITE CICS FILE RECORD TO TS QUEUE                   *
*     - READ NEXT CICS FILE RECORD                           *
*  - END LOOP                                                *
*************************************************************
    PERFORM STARTBR-CICS-FILE.
    PERFORM READNEXT-CICS-FILE.
    MOVE WS-DATA-INQUIRY-KEY    TO WS-INQUIRY-KEY.

    PERFORM LOOP-WRITE-TS-QUEUE
          UNTIL EOF-FILE  OR
               WS-INQUIRY-KEY   NOT = WS-DATA-INQUIRY-KEY.
    PERFORM ENDBR-CICS-FILE.

*****************************************************************
*    - CLOSE THE CICS FILE                                      *
*    - ISSUE OK MESSAGE                                         *
*****************************************************************
    PERFORM CLOSE-CICS-FILE.

    MOVE C-TRANSFER-OK-B        TO COMM-RETURNCODE-B.
    PERFORM TERMINATE-PROGRAM.
*
    GOBACK.
/
*****************************************************************
*    PROGRAM SECTIONS                                           *
*****************************************************************
 LOOP-WRITE-TS-QUEUE SECTION.
       PERFORM WRITE-TS-QUEUE.
       PERFORM READNEXT-CICS-FILE.
 LOOP-WRITE-TS-QUEUE-EXIT.
    EXIT.
```

```
/
*****************************************************************
* CLEAN UP BEFORE RETURNING TO CICS                             *
* IF THE TS QUEUE HAS BEEN CREATED (RECORDS ARE WRITTEN TO IT)  *
* AND THE PROGRAM TERMINATES WITH AN ERRORCODE, THEN THE TS     *
* QUEUE IS DELETED.                                             *
* THE CICS FILE IS CLOSED BEFORE LEAVING THE PROGRAM            *
* THE COMMAREA IS RESTORED IN ORDER TO SETUP THE RETURN CODE    *
* CORRECTLY.                                                    *
*****************************************************************
 TERMINATE-PROGRAM SECTION.
*
    IF  QUEUE-CREATED  AND
        COMM-RETURNCODE-B NOT = C-TRANSFER-OK-B
        EXEC CICS
            DELETEQ TS QUEUE(COMM-UQUEUE-X) NOHANDLE
        END-EXEC.

    IF FILE-OPEN
      EXEC CICS
          SET DATASET(C-DATA-X)
          CLOSED NOHANDLE
      END-EXEC.

    IF HIST-OPEN
      EXEC CICS
          SET DATASET(C-HIST-X)
          CLOSED NOHANDLE
      END-EXEC.
*
    MOVE WS-COMMAREA-G          TO DFHCOMMAREA.
*
    EXEC CICS
```

```
        RETURN
    END-EXEC.
 TERMINATE-PROGRAM-EXIT.
    EXIT.
/
*****************************************************************
* CREATE THE TS QUEUE HEADER RECORD                             *
* THE RECORD IS INITIALLY TRANSFER TO THE PROGRAM IN THE        *
* COMMAREA.  THIS SECTION FILLS IN THE BLANKS ACCORDING TO THE  *
* MANUALS.                                                      *
*****************************************************************
 CREATE-HEADER-RECORD SECTION.
*
    MOVE COMM-HEADER-RECORD-X    TO TS-HDR-RECORD-G.

    MOVE COMM-UFILE-X            TO TS-HDR-FILE-X.
*
    EXEC CICS
        ASKTIME ABSTIME(WS-ABSTIME-P)
              RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
        NEXT SENTENCE
      ELSE
          MOVE C-ASKTIME-ERROR-B TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
*
    EXEC CICS
        FORMATTIME ABSTIME(WS-ABSTIME-P)
                YYDDD(WS-TSM-YYDDD-G)
                TIME(WS-TSM-TIME-G)
                RESP(IO-RESPONSE-B)
```

```
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
        NEXT SENTENCE
      ELSE
         MOVE C-FORMATTIME-ERROR-B TO COMM-RETURNCODE-B
         PERFORM TERMINATE-PROGRAM.

*
    MOVE WS-TSM-HH-X          TO TS-HDR-TIME-H.
    MOVE WS-TSM-MM-X          TO TS-HDR-TIME-M.
    MOVE WS-TSM-SS-X          TO TS-HDR-TIME-S.

    MOVE WS-TSM-YY-X          TO TS-HDR-DATE-Y.
    MOVE WS-TSM-DDD-X         TO TS-HDR-DATE-D.
*
    MOVE '('                  TO TS-HDR-LPAR-X.
    MOVE 'ASCII'              TO TS-HDR-OPT1-X.
    MOVE 'CRLF'               TO TS-HDR-OPT2-X.
    MOVE ')'                  TO TS-HDR-RPAR-X.
 CREATE-HEADER-RECORD-EXIT.
    EXIT.
/
*****************************************************************
* THE CICS SET FILE COMMAND REQUIRES A SPECIAL AUTHORIZATION.   *
* THE CICS FILE IS KNOWN TO THE CICS SYSTEM AS DATA. THE        *
* CICS FILE IS ASSUMED TO BE IN THE CLOSED STATE, IF NOT THE    *
* PROGRAM TERMINATES WITH A ERRORCODE                           *
*****************************************************************
 OPEN-CICS-FILE SECTION.
*
    EXEC CICS
        SET DATASET(C-DATA-X)
```

```
            OPEN
            RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          MOVE '1'    TO SW-OPEN-X
      ELSE
          MOVE C-FILE-ERROR-B  TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
 OPEN-CICS-FILE-EXIT.
    EXIT.
/
*****************************************************************
* DECLARE STARTBR FOR THE CICS FILE FOR A SEQUENTIAL READ       *
*****************************************************************
 STARTBR-CICS-FILE SECTION.
*
    MOVE COMM-UFILE-X       TO WS-DATA-APPNAME-X.
    MOVE TS-HDR-USID-X      TO WS-DATA-USERID-X.
    MOVE COMM-TRMID-X       TO WS-DATA-TRMID-X.
    MOVE ZEROS              TO WS-DATA-SEQNO-B.

    EXEC CICS
        STARTBR  DATASET(C-DATA-X)
                 RIDFLD(WS-DATA-KEY)
                 GTEQ
                 RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
         NEXT SENTENCE
      ELSE
          MOVE C-STARTBR-ERROR-B TO COMM-RETURNCODE-B
```

```
           PERFORM TERMINATE-PROGRAM.
 STARTBR-CICS-FILE-EXIT.
     EXIT.
/
*****************************************************************
* READNEXT TO GET A RECORD FROM THE CICS FILE                   *
*****************************************************************
 READNEXT-CICS-FILE SECTION.
*
*    MOVE LENGTH OF IO-ASCIIREC-G TO IO-RECLEN-B
     MOVE 200 TO IO-RECLEN-B.
     EXEC CICS
          READNEXT DATASET(C-DATA-X)
                   INTO(WS-DATA-RECORD-G)
                   RIDFLD(WS-DATA-KEY)
                   RESP(IO-RESPONSE-B)
     END-EXEC.
*
       IF IO-RESPONSE-B = DFHRESP(NORMAL)
           NEXT SENTENCE
       ELSE
       IF IO-RESPONSE-B = DFHRESP(ENDFILE)
           MOVE '1'    TO  SW-FILE-EOF-X
       ELSE
           MOVE C-READNEXT-ERROR-B TO COMM-RETURNCODE-B
           PERFORM TERMINATE-PROGRAM.
 READNEXT-CICS-FILE-EXIT.
     EXIT.
/
*****************************************************************
* CLOSE THE STARTBR WHEN END OF FILE IS REACHED                 *
*****************************************************************
 ENDBR-CICS-FILE SECTION.
```

```
*
    EXEC CICS
        ENDBR  DATASET(C-DATA-X)
            RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE C-ENDBR-ERROR-B TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
 ENDBR-CICS-FILE-EXIT.
    EXIT.
/
*****************************************************************
* CLOSE THE CICS FILE                                           *
*****************************************************************
 CLOSE-CICS-FILE SECTION.
*
    EXEC CICS
        SET DATASET(C-DATA-X)
            CLOSED
            RESP(IO-RESPONSE-B)
    END-EXEC.

      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE C-FILE-ERROR-B  TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
 CLOSE-CICS-FILE-EXIT.
    EXIT.
/
```

```
*****************************************************************
* PREPARE THE TS QUEUE FOR THE RECEIVE COMMAND                  *
* IF THE TS QUEUE DOES EXIST, THE QUEUE IS DELETED              *
*****************************************************************
 CLEAR-OLD-QUEUE SECTION.
*
**** MOVE LENGTH OF IO-ASCIIREC-G     TO IO-RECLEN-B.
    MOVE 200 TO IO-RECLEN-B.
    EXEC CICS
        READQ TS QUEUE(COMM-UQUEUE-X)
                INTO(IO-ASCIIREC-G)
                LENGTH(IO-RECLEN-B)
                ITEM(C-ONE-B)
                RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(QIDERR)
          NEXT SENTENCE
      ELSE
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          PERFORM DELETE-OLD-QUEUE
      ELSE
          MOVE C-READ-TS-ERROR-B TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
 CLEAR-OLD-QUEUE-EXIT.
    EXIT.
/
*****************************************************************
* DELETE OLD TS QUEUE                                           *
* THE REASON FOR THE SYNCPOINT IS DESCRIBED IN THE CICS MANUAL  *
* AS NECESSARY FOR THE SECURITY CHECKING                        *
*****************************************************************
 DELETE-OLD-QUEUE SECTION.
```

```
*
    EXEC CICS
        DELETEQ TS
                QUEUE(COMM-UQUEUE-X)
                RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          EXEC CICS
              SYNCPOINT
          END-EXEC
      ELSE
          MOVE C-DELETE-TS-ERROR-B TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM
 DELETE-OLD-QUEUE-EXIT.
    EXIT.
/
*****************************************************************
* WRITE A RECORD TO THE TS QUEUE                                *
*****************************************************************
 WRITE-TS-QUEUE SECTION.
*
    ADD +1                  TO IO-RECNO-B.
    MOVE WS-DATA-DETAIL-X   TO IO-ASCIIREC-G.
*
    EXEC CICS
        WRITEQ TS QUEUE(COMM-UQUEUE-X)
                FROM(IO-ASCIIREC-G)
                LENGTH(IO-RECLEN-B)
                ITEM(IO-RECNO-B)
                RESP(IO-RESPONSE-B)
    END-EXEC.
*
```

```
           IF IO-RESPONSE-B = DFHRESP(NORMAL)
               NEXT SENTENCE
           ELSE
               MOVE C-WRITE-TS-ERROR-B TO COMM-RETURNCODE-B
               PERFORM TERMINATE-PROGRAM.
       WRITE-TS-QUEUE-EXIT.
           EXIT.
```

```
IDENTIFICATION DIVISION.
PROGRAM-ID. CFTRDEL.

*****************************************************************
*                                                               *
* CFTRDEL  - THIS PROGRAM IS USED TO DELETE THE DATA FROM THE   *
*          DATA AND HISTORY FILE ACCORDING TO THE KEY FROM      *
*          PC. THEN THE PROGRAM USE THE RECEIVE FUNCTION TO     *
*          DOWNLOAD THE DATA TO THE PC FILE.                    *
*                                                               *
*          A DESCRIPTION OF THE STANDARDS FOR FILE TRANSFER     *
*          CAN BE FOUND IN THE FOLLOWING VSE MANUAL.            *
*          >>  VSE/ESA ADMINISTRATION                           *
*          >>          CHAPTER 11.  USING WORKSTATION FILE      *
*          >>          TRANSFER INTERFACES AND FUNCTIONS.       *
*          >>          - FILE TRANSFER TO AND FROM CICS/VSE     *
*          >>            TEMPORARY STORAGE                      *
*                                                               *
*          THIS PROGRAM USES A SET OF CICS COMMANDS, WHICH      *
*          HAVE BEEN MADE SUBJECT FOR SECURITY CHECKING.        *
*          A DESCRIPTION OF THE SECURITY FOR THESE CICS         *
*          COMMANDS CAN BE FOUND IN THE CICS MANUALS.           *
*          >> CICS/VSE SYSTEM PROGRAMMING REFERENCE             *
*          >>          CHAPTER 1. INTRUDUCTION                  *
*
*                                                               *
*                                                               *
* VERSION 1.0.0                                                 *
* modify :  11/03/95 : Check - If History file exist, continue  *
*                   process.                                    *
*                                                               *
*****************************************************************
```

```
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*****************************************************************
*                                                               *
*   DEFINITION OF THE COMMUNICATION AREA                        *
*     (LAYOUT FROM VSE/ESA MANUAL)                              *
*                                                               *
*****************************************************************
01  WS-COMMAREA-G.
      05 COMM-ACTION-X          PIC X.
        88  COMM-ACTION-SEND        VALUE 'U'.
        88  COMM-ACTION-RECEIVE     VALUE 'D'.
      05 FILLER                 PIC X.
      05 COMM-RETURNCODE-B      PIC S9(4) COMP.
      05 COMM-UFILE-X           PIC X(8).
      05 COMM-UQUEUE-X          PIC X(8).
      05 FILLER  REDEFINES COMM-UQUEUE-X.
        07 COMM-CFTR-X          PIC X(4).
        07 COMM-TRMID-X         PIC X(4).
      05 FILLER                 PIC X(8).
      05 COMM-HEADER-RECORD-X   PIC X(160).
*
*****************************************************************
*                                                               *
*   DEFINITION OF THE TS QUEUE HEADER RECORD                    *
*     (LAYOUT FROM VSE/ESA MANUAL)                              *
*                                                               *
*****************************************************************
01  TS-HDR-RECORD-G.
      05 TS-HDR-FILE-X          PIC X(8).
      05 FILLER                 PIC X.
      05 TS-HDR-TIME-G.
```

```
           10 TS-HDR-TIME-H        PIC X(2).
           10 TS-HDR-TIME-M        PIC X(2)
           10 TS-HDR-TIME-S        PIC X(2).
        05 FILLER                  PIC X.
        05 TS-HDR-DATE-G.
           10 TS-HDR-DATE-Y        PIC X(2).
           10 TS-HDR-DATE-D        PIC X(3).
        05 FILLER                  PIC X(1).
        05 TS-HDR-LPAR-X           PIC X.
        05 TS-HDR-OPT1-X           PIC X(6).
        05 FILLER                  PIC X.
        05 TS-HDR-OPT2-X           PIC X(6).
        05 FILLER                  PIC X.
        05 TS-HDR-TYPE-X           PIC X(8).
        05 FILLER                  PIC X.
        05 TS-HDR-DCDF-X           PIC X(8).
        05 FILLER                  PIC X.
        05 TS-HDR-USID-X           PIC X(8).
        05 FILLER                  PIC X.
        05 TS-HDR-RESERVD-X        PIC X(28).
        05 FILLER                  PIC X.
        05 TS-HDR-RPAR-X           PIC X.
        05 TS-HDR-COMM-X           PIC X(66).

*****************************************************************
*    DEFINITION OF THE WORKING FOR HISTORY RECORD               *
*    WS-HIST-APP-X WILL BE BLANK (USE FOR NEXT VERSION)         *
*****************************************************************
 01  WS-HIST-RECORD-G.
     02 WS-HIST-KEY.
        05 WS-HIST-APPNAME-X          PIC X(8).
        05 WS-HIST-USERID-X           PIC X(4).
        05 WS-HIST-TRMID-X            PIC X(4).
```

```
    02 WS-HIST-NUM-B                PIC 9(4).
    02 WS-HIST-APP-X                PIC X(10).

*****************************************************************
*   DEFINITION OF THE WORKING FOR DATA  RECORD                  *
*****************************************************************
 01  WS-DATA-RECORD-G.
    02 WS-DATA-KEY.
       03 WS-DATA-INQUIRY-KEY.
          05 WS-DATA-APPNAME-X       PIC X(8).
          05 WS-DATA-USERID-X        PIC X(4).
          05 WS-DATA-TRMID-X         PIC X(4).
       03 WS-DATA-SEQNO-B            PIC 9(4).
    02 WS-DATA-DETAIL-X              PIC X(200).

*****************************************************************
*                                                               *
*   ASKTIME/FORMATTIME FIELDS                                   *
*     USED TO UPDATE THE DATE AND TIME IN THE TS QUEUE HEADER   *
*                                                               *
*****************************************************************
 77  WS-ABSTIME-P                PIC S9(15) COMP-3.

 01  WS-INQUIRY-KEY.
       05 WS-INQUIRY-APPNAME-X     PIC X(8).
       05 WS-INQUIRY-USERID-X      PIC X(4).
       05 WS-INQUIRY-TRMID-X       PIC X(4).

 01  WS-TSM-YYDDD-G.
       05 WS-TSM-YY-X              PIC X(2).
       05 FILLER                PIC X.
       05 WS-TSM-DDD-X             PIC X(3).
```

```
01  WS-TSM-TIME-G.
        05 WS-TSM-HH-X          PIC X(2).
        05 FILLER               PIC X.
        05 WS-TSM-MM-X          PIC X(2).
        05 FILLER               PIC X.
        05 WS-TSM-SS-X          PIC X(2).

*****************************************************************
*                                                               *
*      RETURN CODES                                             *
*                                                               *
*****************************************************************
 77  C-TRANSFER-OK-B            PIC S9(4) COMP VALUE +0.
 77  C-NO-COMMAREA-B            PIC S9(4) COMP VALUE +1.
 77  C-TRMID-ERROR-B            PIC S9(4) COMP VALUE +2.
 77  C-STARTBR-ERROR-B          PIC S9(4) COMP VALUE +3.
 77  C-READNEXT-ERROR-B         PIC S9(4) COMP VALUE +4.
 77  C-ENDBR-ERROR-B            PIC S9(4) COMP VALUE +5.
 77  C-READ-TS-ERROR-B          PIC S9(4) COMP VALUE +6.
 77  C-DELETE-TS-ERROR-B        PIC S9(4) COMP VALUE +7.
 77  C-WRITE-TS-ERROR-B         PIC S9(4) COMP VALUE +8.
 77  C-FILE-ERROR-B             PIC S9(4) COMP VALUE +9.
 77  C-ASKTIME-ERROR-B          PIC S9(4) COMP VALUE +10.
 77  C-FORMATTIME-ERROR-B       PIC S9(4) COMP VALUE +11.
 77  C-SEND-COMM-B              PIC S9(4) COMP VALUE +12.
 77  C-IO-ERROR-B               PIC S9(4) COMP VALUE +13.
 77  C-HISTREC-NOTFND-B         PIC S9(4) COMP VALUE +14.

*****************************************************************
*                                                               *
*   INPUT/OUTPUT DEFINITIONS                                    *
*                                                               *
*****************************************************************
```

```
 77  SW-FILE-EOF-X                PIC X  VALUE '0'.
         88 NOT-EOF-FILE                VALUE '0'.
         88 EOF-FILE                    VALUE '1'.
*
 77  SW-QUEUE-X                   PIC X  VALUE '0'.
         88 QUEUE-NOT-CREATED            VALUE '0'.
         88 QUEUE-CREATED                VALUE '1'.
*
 77  SW-DATA-OPEN-X               PIC X  VALUE '0'.
         88 DATA-NOT-OPEN                VALUE '0'.
         88 DATA-OPEN                    VALUE '1'.

 77  SW-HIST-OPEN-X               PIC X  VALUE '0'.
         88 HIST-NOT-OPEN                VALUE '0'.
         88 HIST-OPEN                    VALUE '1'.

*
 01  IO-ASCIIREC-G.
         05 FILLER                PIC X(200).
*
 77  IO-RECNO-B                   PIC S9(4) COMP VALUE +0.
 77  IO-RECLEN-B                  PIC S9(4) COMP VALUE +220.
 77  IO-RECRBA-B                  PIC S9(8) COMP.
 77  IO-RESPONSE-B                PIC S9(8) COMP.
*
 77  C-ZERO-B                     PIC S9(4) COMP VALUE +0.
 77  C-ONE-B                      PIC S9(4) COMP VALUE +1.
 77  C-DATA-X                     PIC X(6) VALUE 'PF001L'.
 77  C-HIST-X                     PIC X(6) VALUE 'PF002L'.
*
/
*****************************************************************
*                                                               *
```

```
*      LINKAGE AREA -  DFHCOMMAREA                    *
*                                                     *
*******************************************************************
 LINKAGE SECTION.
 01  DFHCOMMAREA                 PIC X(188).
*
*******************************************************************
*                                                     *
*   START OF EXECUTION                                *
*                                                     *
*******************************************************************
 PROCEDURE DIVISION.
*
*******************************************************************
*   CHECK IF A COMMAREA WAS PASSED                    *
*   IF NOT, TERMINATE THE PROGRAM WITH AN ERROR-CODE  *
*******************************************************************
    IF EIBCALEN = 0
      MOVE C-NO-COMMAREA-B      TO COMM-RETURNCODE-B
      PERFORM TERMINATE-PROGRAM
    ELSE
      MOVE DFHCOMMAREA          TO WS-COMMAREA-G.

*******************************************************************
*   INITIALIZE THE RETURNCODE TO ZERO                 *
*******************************************************************
    MOVE C-ZERO-B               TO COMM-RETURNCODE-B.

*******************************************************************
*   CHECK IF THE PROGRAM IS ACTIVATE BY A RECEIVE COMMAND  *
*   IF NOT, TERMINATE THE PROGRAM WITH AN ERROR-CODE       *
*******************************************************************
    IF COMM-ACTION-SEND
```

```
      MOVE C-SEND-COMM-B       TO COMM-RETURNCODE-B
      PERFORM TERMINATE-PROGRAM.

*****************************************************************
*   CHECK IF THE STANDARD TS QUEUE NAME IS USED                 *
*   IF NOT, TERMINATE THE PROGRAM WITH AN ERROR-CODE            *
*****************************************************************
   IF  COMM-TRMID-X = EIBTRMID AND COMM-CFTR-X = 'CFTR'
       NEXT SENTENCE
   ELSE
      MOVE C-TRMID-ERROR-B      TO COMM-RETURNCODE-B
      PERFORM TERMINATE-PROGRAM.

*****************************************************************
* READ THE KEY FROM CICS FILE HIST TO CHECK THE KEY IS EXIST.   *
* IF HIST EXIST,  CONTINUE THE PROCESS.                         *
*****************************************************************
 CHECK-HIST-KEY    SECTION.
*
   EXEC CICS
       SET DATASET(C-HIST-X)
          OPEN
          RESP(IO-RESPONSE-B)
   END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
         NEXT SENTENCE
      ELSE
      IF IO-RESPONSE-B = DFHRESP(IOERR)
         MOVE C-IO-ERROR-B      TO COMM-RETURNCODE-B
         PERFORM TERMINATE-PROGRAM
      ELSE
         MOVE C-FILE-ERROR-B      TC COMM-RETURNCODE-B
```

```
           PERFORM TERMINATE-PROGRAM.
*
*     MOVE LENGTH OF WS-HIST-RECORD-G TO IO-RECLEN-B.
      MOVE 30 TO IO-RECLEN-B.
      MOVE +0                    TO IO-RECRBA-B.

******* move comm header area to ts-hdr-record-g *****
******* and move comm area to hist key fields ********
     MOVE COMM-HEADER-RECORD-X    TO TS-HDR-RECORD-G.
     MOVE COMM-UFILE-X            TO WS-HIST-APPNAME-X.
     MOVE TS-HDR-USID-X           TO WS-HIST-USERID-X.
     MOVE COMM-TRMID-X            TO WS-HIST-TRMID-X.

     EXEC CICS
         READ DATASET(C-HIST-X)
              INTO(WS-HIST-RECORD-G)
              RIDFLD(WS-HIST-KEY)
              RESP(IO-RESPONSE-B)
     END-EXEC.

       IF IO-RESPONSE-B = DFHRESP(NORMAL)
           NEXT SENTENCE
       ELSE
         IF IO-RESPONSE-B = DFHRESP(NOTFND)
           MOVE C-HISTREC-NOTFND-B  TO COMM-RETURNCODE-B
           PERFORM  TERMINATE-PROGRAM
         ELSE
           MOVE C-FILE-ERROR-B      TO COMM-RETURNCODE-B
           PERFORM  TERMINATE-PROGRAM.

 CHECK-HIST-KEY-EXIT.
     EXIT.
```

```
*****************************************************************
*    CHECK IF THE TS QUEUE EXISTS                               *
*****************************************************************
    PERFORM CLEAR-OLD-QUEUE.

*****************************************************************
*    OPEN THE CICS FILE                                         *
*****************************************************************
    PERFORM OPEN-CICS-FILE.

*****************************************************************
*    WRITE THE TS QUEUE HEADER TO THE TS QUEUE                  *
*****************************************************************
    PERFORM CREATE-HEADER-RECORD.
    MOVE TS-HDR-RECORD-G        TO IO-ASCIIREC-G.
*    MOVE LENGTH OF TS-HDR-RECORD-G
*                                TO IO-RECLEN-B.
    MOVE 160 TO IO-RECLEN-B.
    MOVE +0                     TO IO-RECNO-B.
    EXEC CICS
        WRITEQ TS QUEUE(COMM-UQUEUE-X)
        FROM(IO-ASCIIREC-G)
        LENGTH(IO-RECLEN-B)
        ITEM(IO-RECNO-B)
        RESP(IO-RESPONSE-B)
    END-EXEC.
    MOVE '1'                    TO SW-QUEUE-X

*****************************************************************
* THE CICS FILE IS READ SEQUENTIALLY UNTIL END OF FILE.         *
* NO CHECKING OF THE DATA RECORD IN THE CICS FILE IS DONE       *
*                                                               *
*  - READ FIRST DATA RECORD FROM THE CICS FILE                  *
```

```
*  - LOOP UNTIL NO MORE CICS FILE RECORDS                    *
*     - WRITE CICS FILE RECORD TO TS QUEUE                   *
*     - READ NEXT CICS FILE RECORD                           *
*  - END LOOP                                                *
*************************************************************
    PERFORM STARTBR-CICS-FILE.
    PERFORM READNEXT-CICS-FILE.
    MOVE WS-DATA-INQUIRY-KEY    TO WS-INQUIRY-KEY.

    PERFORM LOOP-DELETE-DATA-RECORDS
          UNTIL EOF-FILE  OR
              WS-INQUIRY-KEY   NOT = WS-DATA-INQUIRY-KEY.

    PERFORM DELETE-HISTORY-RECORD.
    PERFORM ENDBR-CICS-FILE.

*************************************************************
*    - CLOSE THE CICS FILE                                  *
*    - ISSUE OK MESSAGE                                     *
*************************************************************
    PERFORM CLOSE-CICS-FILE.

    MOVE C-TRANSFER-OK-B        TO COMM-RETURNCODE-B.
    PERFORM TERMINATE-PROGRAM.
*
    GOBACK.
/
*************************************************************
*************************************************************
*************************************************************
*    PROGRAM SECTIONS '                                     *
*************************************************************
 LOOP-DELETE-DATA-RECORDS SECTION.
```

```
       PERFORM WRITE-TS-QUEUE.
       PERFORM DELETE-DATA-RECORDS.
       PERFORM READNEXT-CICS-FILE.
 LOOP-DELETE-DATA-RECORDS-EXIT.
     EXIT.
/
*****************************************************************
* CLEAN UP BEFORE RETURNING TO CICS                             *
* IF THE TS QUEUE HAS BEEN CREATED (RECORDS ARE WRITTEN TO IT)  *
* AND THE PROGRAM TERMINATES WITH AN ERRORCODE, THEN THE TS     *
* QUEUE IS DELETED.                                             *
* THE CICS FILE IS CLOSED BEFORE LEAVING THE PROGRAM            *
* THE COMMAREA IS RESTORED IN ORDER TO SETUP THE RETURN CODE    *
* CORRECTLY.                                                    *
*****************************************************************
 TERMINATE-PROGRAM SECTION.
*
    IF  QUEUE-CREATED  AND
        COMM-RETURNCODE-B NOT = C-TRANSFER-OK-B
        EXEC CICS
            DELETEQ TS QUEUE(COMM-UQUEUE-X) NOHANDLE
        END-EXEC.

    IF DATA-OPEN
      EXEC CICS
          SET DATASET(C-DATA-X)
          CLOSED NOHANDLE
      END-EXEC.

    IF HIST-OPEN
      EXEC CICS
          SET DATASET(C-HIST-X)
          CLOSED NOHANDLE
```

```
      END-EXEC.
*
    MOVE WS-COMMAREA-G          TO DFHCOMMAREA.
*
    EXEC CICS
        RETURN
    END-EXEC.
 TERMINATE-PROGRAM-EXIT.
    EXIT.
/
*****************************************************************
* CREATE THE TS QUEUE HEADER RECORD                             *
* THE RECORD IS INITIALLY TRANSFER TO THE PROGRAM IN THE        *
* COMMAREA.  THIS SECTION FILLS IN THE BLANKS ACCORDING TO THE  *
* MANUALS.                                                      *
*****************************************************************
 CREATE-HEADER-RECORD SECTION.
*
    MOVE COMM-HEADER-RECORD-X    TO TS-HDR-RECORD-G.

    MOVE COMM-UFILE-X            TO TS-HDR-FILE-X.
*
    EXEC CICS
        ASKTIME ABSTIME(WS-ABSTIME-P)
             RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
        NEXT SENTENCE
      ELSE
          MOVE C-ASKTIME-ERROR-B TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
*
```

```
      EXEC CICS
          FORMATTIME ABSTIME(WS-ABSTIME-P)
                     YYDDD(WS-TSM-YYDDD-G)
                     TIME(WS-TSM-TIME-G)
                     RESP(IO-RESPONSE-B)
      END-EXEC.
*
        IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
        ELSE
            MOVE C-FORMATTIME-ERROR-B TO COMM-RETURNCODE-B
            PERFORM TERMINATE-PROGRAM.

*
      MOVE WS-TSM-HH-X          TO TS-HDR-TIME-H.
      MOVE WS-TSM-MM-X          TO TS-HDR-TIME-M.
      MOVE WS-TSM-SS-X          TO TS-HDR-TIME-S.

      MOVE WS-TSM-YY-X          TO TS-HDR-DATE-Y.
      MOVE WS-TSM-DDD-X         TO TS-HDR-DATE-D.
*
      MOVE '('                  TO TS-HDR-LPAR-X.
      MOVE 'ASCII'              TO TS-HDR-OPT1-X.
      MOVE 'CRLF'               TO TS-HDR-OPT2-X.
      MOVE ')'                  TO TS-HDR-RPAR-X.
 CREATE-HEADER-RECORD-EXIT.
      EXIT.
/
*****************************************************************
* THE CICS SET FILE COMMAND REQUIRES A SPECIAL AUTHORIZATION.   *
* THE CICS FILE IS KNOWN TO THE CICS SYSTEM AS RDBUPD. THE      *
* CICS FILE IS ASSUMED TO BE IN THE CLOSED STATE, IF NOT THE    *
* PROGRAM TERMINATES WITH A ERRORCODE                           *
```

```
*****************************************************************
 OPEN-CICS-FILE SECTION.
***** OPEN CICS DATA FILE **************
    EXEC CICS
        SET DATASET(C-DATA-X)
            OPEN
            RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          MOVE '1'    TO SW-DATA-OPEN-X
      ELSE
          MOVE C-FILE-ERROR-B  TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.

 OPEN-CICS-FILE-EXIT.
    EXIT.
/
*****************************************************************
* DECLARE STARTBR FOR THE CICS FILE FOR A SEQUENTIAL READ       *
*****************************************************************
 STARTBR-CICS-FILE SECTION.
*
    MOVE COMM-UFILE-X       TO WS-DATA-APPNAME-X.
    MOVE TS-HDR-USID-X      TO WS-DATA-USERID-X.
    MOVE COMM-TRMID-X       TO WS-DATA-TRMID-X.
    MOVE 1                  TO WS-DATA-SEQNO-B.

    EXEC CICS
        STARTBR  DATASET(C-DATA-X)
                 RIDFLD(WS-DATA-KEY)
                 GTEQ
                 RESP(IO-RESPONSE-B)
```

```
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE C-STARTBR-ERROR-B TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
 STARTBR-CICS-FILE-EXIT.
    EXIT.
/
*****************************************************************
* READNEXT TO GET A RECORD FROM THE CICS FILE                   *
*****************************************************************
 READNEXT-CICS-FILE SECTION.
*
**** MOVE LENGTH OF IO-ASCIIREC-G TO IO-RECLEN-B
    MOVE 200 TO IO-RECLEN-B.
*
    EXEC CICS
        READNEXT DATASET(C-DATA-X)
                 INTO(WS-DATA-RECORD-G)
                 RIDFLD(WS-DATA-KEY)
                 RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
      IF IO-RESPONSE-B = DFHRESP(ENDFILE)
          MOVE '1'    TO  SW-FILE-EOF-X
      ELSE
          MOVE C-READNEXT-ERROR-B TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
```

```
 READNEXT-CICS-FILE-EXIT.
     EXIT.
/
*****************************************************************
* CLOSE THE STARTBR WHEN END OF FILE IS REACHED                 *
*****************************************************************
 ENDBR-CICS-FILE SECTION.
*
     EXEC CICS
          ENDBR  DATASET(C-DATA-X)
                 RESP(IO-RESPONSE-B)
     END-EXEC.
*
       IF IO-RESPONSE-B = DFHRESP(NORMAL)
           NEXT SENTENCE
       ELSE
           MOVE C-ENDBR-ERROR-B TO COMM-RETURNCODE-B
           PERFORM TERMINATE-PROGRAM.
 ENDBR-CICS-FILE-EXIT.
     EXIT.
/
*****************************************************************
* CLOSE THE CICS FILE                                           *
* THE CICS SET FILE COMMAND REQUIRES A SPECIAL AUTHORIZATION.   *
*****************************************************************
 CLOSE-CICS-FILE SECTION.
*
     EXEC CICS
          SET DATASET(C-DATA-X)
              CLOSED
              RESP(IO-RESPONSE-B)
     END-EXEC.
```

```
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE C-FILE-ERROR-B  TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.

    EXEC CICS
        SET DATASET(C-HIST-X)
            CLOSED
            RESP(IO-RESPONSE-B)
    END-EXEC.

      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
          MOVE C-FILE-ERROR-B  TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
 CLOSE-CICS-FILE-EXIT.
     EXIT.
/
*****************************************************************
* PREPARE THE TS QUEUE FOR THE RECEIVE COMMAND                  *
* IF THE TS QUEUE DOES EXIST, THE QUEUE IS DELETED              *
*****************************************************************
 CLEAR-OLD-QUEUE SECTION.
*
*****MOVE LENGTH OF IO-ASCIIREC-G    TO IO-RECLEN-B.
    MOVE 200 TO IO-RECLEN-B.

    EXEC CICS
        READQ TS QUEUE(COMM-UQUEUE-X)
              INTO(IO-ASCIIREC-G)
              LENGTH(IO-RECLEN-B)
```

```
                ITEM(C-ONE-B)
                RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(QIDERR)
          NEXT SENTENCE
      ELSE
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          PERFORM DELETE-OLD-QUEUE
      ELSE
          MOVE C-READ-TS-ERROR-B TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM.
 CLEAR-OLD-QUEUE-EXIT.
    EXIT.
/
*****************************************************************
* DELETE OLD TS QUEUE                                           *
* THE REASON FOR THE SYNCPOINT IS DESCRIBED IN THE CICS MANUAL  *
* AS NECESSARY FOR THE SECURITY CHECKING                        *
*****************************************************************
 DELETE-OLD-QUEUE SECTION.
*
    EXEC CICS
        DELETEQ TS
                QUEUE(COMM-UQUEUE-X)
                RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          EXEC CICS
              SYNCPOINT
          END-EXEC
      ELSE
```

```
           MOVE C-DELETE-TS-ERROR-B TO COMM-RETURNCODE-B
           PERFORM TERMINATE-PROGRAM.
 DELETE-OLD-QUEUE-EXIT.
     EXIT.
/
*****************************************************************
* WRITE A RECORD TO THE TS QUEUE                                *
*****************************************************************
 WRITE-TS-QUEUE SECTION.
*
     ADD +1                    TO IO-RECNO-B.
     MOVE WS-DATA-DETAIL-X     TO IO-ASCIIREC-G.
*
     EXEC CICS
         WRITEQ TS QUEUE(COMM-UQUEUE-X)
                   FROM(IO-ASCIIREC-G)
                   LENGTH(IO-RECLEN-B)
                   ITEM(IO-RECNO-B)
                   RESP(IO-RESPONSE-B)
     END-EXEC.
*
       IF IO-RESPONSE-B = DFHRESP(NORMAL)
           NEXT SENTENCE
       ELSE
           MOVE C-WRITE-TS-ERROR-B TO COMM-RETURNCODE-B
           PERFORM TERMINATE-PROGRAM.
 WRITE-TS-QUEUE-EXIT.
     EXIT.
/
*****************************************************************
*    DELETE RECORDS FROM CICS DATA FILE                         *
*****************************************************************
 DELETE-DATA-RECORDS  SECTION.
```

```
    EXEC CICS
        DELETE  DATASET(C-DATA-X)
        RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
      IF IO-RESPONSE-B = DFHRESP(IOERR)
          MOVE C-IO-ERROR-B          TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM
      ELSE
          MOVE C-FILE-ERROR-B        TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM
 DELETE-DATA-RECORDS-EXIT.
    EXIT.
*************************************************************
*    DELETE HISTORY RECORD FROM CICS HIST FILE              *
*************************************************************
 DELETE-HISTORY-RECORD  SECTION.
    EXEC CICS
        DELETE  DATASET(C-HIST-X)
        RIDFLD(WS-INQUIRY-KEY)
        RESP(IO-RESPONSE-B)
    END-EXEC.
*
      IF IO-RESPONSE-B = DFHRESP(NORMAL)
          NEXT SENTENCE
      ELSE
      IF IO-RESPONSE-B = DFHRESP(IOERR)
          MOVE C-IO-ERROR-B          TO COMM-RETURNCODE-B
          PERFORM TERMINATE-PROGRAM
      ELSE
```

```
            MOVE C-FILE-ERROR-B         TO COMM-RETURNCODE-B
            PERFORM TERMINATE-PROGRAM.
DELETE-HISTORY-RECORD-EXIT.
    EXIT.
```

ประวัติผู้เขียน

นาวาอากาศตรีหญิง ลดารัตน์ ผ่องอุไร เกิดเมื่อวันที่ 6 กุมภาพันธ์ 2500 สำเร็จการศึกษาระดับปริญญาตรี ศิลปศาสตรบัณฑิต (รัฐศาสตร์) คณะรัฐศาสตร์ มหาวิทยาลัยรามคำแหง เมื่อ พ.ศ. 2521

เข้าศึกษาต่อในหลักสูตร วิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2533