52

TCP/SNA update

December 2003

In this issue

TCP/SNA Update

Published by

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

North American office

Xephon
Post Office Box 350100
Westminster, CO 80035-0100
USA
Telephone: (303) 410-9344

Subscriptions and back-issues

A year's subscription to *TCP/SNA Update*, comprising four quarterly issues, costs £130.00 in the UK; $190.00 in the USA and Canada; £136.00 in Europe; £142.00 in Australasia and Japan; and £140.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the March 1999 issue, are available separately to subscribers for £33.00 ($49.50) each including postage.

Editorial panel

Articles published in TCP/SNA Update are reviewed by our panel of experts. Members include John Bradley (UK), Carlson Colomb (Canada), Anura Gurugé (USA), Jon Pearkins (Canada), and Tod Yampel (USA).

Editor

Fiona Hewitt

Contributions

When Xephon is given copyright, articles published in *TCP/SNA Update* are paid for at £100 ($160) per 1000 words and £50 ($80) per 100 lines of code for the first 200 lines of original material. To find out more, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

TCP/SNA Update on-line

Code from *TCP/SNA Update*, and complete issues in Acrobat PDF format, can be downloaded from http://www.xephon.com/tcpsna; you will need to supply a word from the printed issue..

OS/390 ftp acknowledgement

As the standard communications protocol, TCP/IP offers an array of tools that companies can use to provide better service to their clients. Among the suite of services that TCP/IP offers, ftp is the one that provides an easy method of transferring files between hosts. But as the number of ftp users increases, so too does the need to acknowledge them, especially if these clients are external ones.

At our company we needed to develop a system that would enable us to confirm to our external customers that we had received their ftp transmission. Although there are several commercial packages that can perform this task, we decided to do it ourselves by following the procedure shown below. Although the methodology we ended up with was very simple, it's proved to be a cost-effective solution.

I've included all the source code. It all works fine for us, but feel free to modify it to meet your company's requirements.

SYSTEM OVERVIEW

A very simple ftp acknowledgement system was developed in the OS/390 environment in order to send a short e-mail message to ftp clients; the message can be customized for each client. As of this writing, a maximum of three e-mail addresses can be entered in the system. One of the main benefits of the system is that it provides 'real-time' response to the clients.

TECHNICAL OVERVIEW

The main technical element is the activation of the FTPSERVER exit, FTPSMFEX. This is done by modifying the TCPIP.FTP.DATA dataset. Every time a transmission is received, and just before writing to the SMF buffer, the exit sends a message to the system console. This message will have FTPSMFEX: as its message id and will contain the client userid, dataset name, PDS member (if applicable), and byte count.

Member MPFLST00 in SYS1.PARMLIB was modified so that it could process every FTPSMFEX: message. The MPF activates MPFTPEX, which at the same time starts STC FTPSMTP. This STC checks whether the FTP client is defined in a special VSAM file (EDSSJ.HTS.MDVJFTPE). If so, an e-mail message is sent to the client with the information about the ftp transmission. The e-mail is sent using Lyonel B Dyck's XMITIP product (www.lbdsoftware.com).

VSAM FILE – EDSSJ.HTS.MDVJFTPE

The EDSSJ.HTS.MDVJFTPE VSAM file contains the information about the FTP client; its key is the RACF userid. Each userid has at least one e-mail address, which will be used to send the acknowledgement; a maximum of three e-mail addresses can be specified. A short message can also be included. The file can be updated using the CICS FTPE transaction.

FTPSMFEX

```
****   FTPSMFEX Example, BY Jose Ramirez (jramirez@ssspr.com)
****   This source is provided with no warranty, you should change
****   it in order to meet your own requirements.
****
FTPSMFEX CSECT
FTPSMFEX AMODE 31
FTPSMFEX RMODE 24
         SAVE  (14,12)
         BALR  12,Ø
         USING *,12
         B     BEGIN
         DC    C'FTPSMFEX '
         DC    C' &SYSDATE '
         DC    C' &SYSTIME '
         DS    ØF
BEGIN    LR    R2,R1                    *Parm pointer
         USING PARMS,R2
         L     R4,PTRRC                 *-> Return code field
         L     R9,PTRSMFR
         USING SMFREC,R9
         L     R3,SMFREMIP              *Foreign IP Address
```

```
*
DPLYINFO MVC   WUSERI,SMFUSERI
         MVC   WDSN,SMFDSN
         MVC   WPDS,SMFPDS
         MVC   WBCNT,SMFBYCNT
         LA    R6,WTOLST1
         WTO   MF=(E,(R6))

         LA    R15,Ø                    *DO NOT WRITE RECORD
DONE     ST    R15,Ø(R4)                *Return the RC
         RETURN (14,12),RC=(15)    RETURN TO CALLER
**
DATAAREA DS ØC
DATALEN  DC AL2(DATAEND-DATACMD)
DATACMD  DC C'S TCPDROP,T=YCPØØ5ØØ'
DATAEND  EQU *
**
@COMMAND  DS F
@DATAAREA DC A(DATAAREA)
@EPA      DS F
**
WTOLST1  DC AL2(WTOLST1X-WTOLST1)
         DC H'Ø'
         DC C'FTPSMFEX: '
WUSERI   DS CL8
WDSN     DS CL4Ø
WPDS     DS CL8
WBCNT    DS AL4(Ø)
WTOLST1X EQU *
**
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
*
PARMS    DSECT
PTRRC    DC    F'Ø'
PTRSMFR  DC    F'Ø'
```

```
*
SMFREC   DSECT                        *FTP Server SMF record
         DC    24X'ØØ'                *Std. SMF header
SMFCMD   DC    CL4' '                 *FTP subcommand
SMFFTYPE DC    CL4' '                 *File type (SEQ,JCL,SQL)
SMFREMIP DC    AL4(Ø)                 *Foreign host IP address
SMFLOCIP DC    AL4(Ø)                 *Local IP address
FILL1    DC    CL8' '                 *FILLER NOT USED
SMFUSERI DC    CL8' '                 *USER ID
FILL2    DC    CL4' '                 *FILLER NOT USED
SMFSTIME DC    AL4(Ø)
SMFETIME DC    AL4(Ø)
SMFBYCNT DC    AL4(Ø)
FILL3    DC    CL4' '                 *FILLER NOT USED
SMFDSN   DC    CL44' '
SMFPDS   DC    CL8' '
         DS    ØF                     *Remainder of record not use
d
*
         END
```

MPFTPEX

```
//WTOEXIT  JOB (5B1ØB954,Ø1),'FRANK',
JOBCARD
//             MSGLEVEL=(1,1),
JOBCARD
//             REGION=6M,
CARD
//             MSGCLASS=T,
JOBCARD
//             CLASS=Z
JOBCARD
//*
JOBCARD
//*****************************************************************
//*****************************************************************
//*
//ASMA9Ø   EXEC PGM=ASMA9Ø,
//             PARM='NODECK,OBJ,RENT',
//             REGION=6M
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//         DD  DSN=SYS1.MODGEN,DISP=SHR
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=VIO,SPACE=(17ØØ,(6ØØ,1ØØ))
//SYSUT2   DD  DSN=&&SYSUT2,UNIT=VIO,SPACE=(17ØØ,(3ØØ,5Ø))
//SYSUT3   DD  DSN=&&SYSUT3,UNIT=VIO,SPACE=(17ØØ,(3ØØ,5Ø))
//SYSPRINT DD  SYSOUT=*,DCB=BLKSIZE=1Ø89
//SYSPUNCH DD  DUMMY
//SYSLIN   DD  DSN=&&OBJSET,UNIT=SYSSQ,SPACE=(8Ø,(2ØØ,5Ø)),
//             DISP=(MOD,PASS)
//SYSIN    DD  *
MPFTPEX    TITLE 'MPFTPEX -- WTO EXIT TO ISSUE S XXX AFTER FTP TRANS'
MPFTPEX    CSECT
MPFTPEX    AMODE 31
MPFTPEX    RMODE ANY
***********************************************************************
*                                                                     *
*         REGISTER ASSIGNMENTS                                        *
*                                                                     *
***********************************************************************
RØ       EQU   Ø                        REGISTER Ø
R1       EQU   1                        REGISTER 1
R2       EQU   2                        REGISTER 2
R3       EQU   3                        REGISTER 3
R4       EQU   4                        REGISTER 4
R5       EQU   5                        REGISTER 5 - POINTS TO CTXT
R6       EQU   6                        REGISTER 6
R7       EQU   7                        REGISTER 7
R8       EQU   8                        REGISTER 8
R9       EQU   9                        REGISTER 9
R1Ø      EQU   1Ø                       REGISTER 1Ø - CURRENTLY UNUSED
R11      EQU   11                       REGISTER 11 - DYNAMIC DATA AREA
R12      EQU   12                       REGISTER 12
BASEREG  EQU   12                       REGISTER 12 - MODULE BASE
R13      EQU   13                       REGISTER 13
R14      EQU   14                       REGISTER 14
R15      EQU   15                       REGISTER 15
         SPACE 1
         USING *,R12
         STM   R14,R12,12(R13)
         LR    R12,R15
         B     START

         DC    C'MPFTPEX:'
         DC    C'&SYSDATE',C' &SYSTIME'
         PRINT NOGEN

START    DS    ØH
         L     R11,Ø(R1)               PICK UP ADDRESS OF CTXT
         USING CTXT,R11

         L     R1Ø,CTXTTXPJ            POINTER TO MESSAGE ATTRIBUTES
         USING CTXTATTR,R1Ø

*        WTO   'MPFTPEX: Entrando exit MPFTPEX!!'

         CLC   =C'FTPSMFEX: ',CTXTTMSG  FTP END OF TRANSMISION  ???
         BNE   RETURN                  NOT CORRECT MESSAGE, FORGET IT
```

```
PROCESS  DS    ØH
         WTO   'MPFTPEX: Se activa exit MPFTPEX!!'

         GETMAIN RU,LV=WORKEND,SP=23Ø,LOC=(BELOW,ANY)
         ST    R13,4(R1)
         ST    R1,8(R13)
         LR    R13,R1
         USING WORKAREA,R13

         XC    PARMAREA,PARMAREA
         XC    MGCRPL(MGCRSIZ),MGCRPL CLEAR THE MGCR AREA

         MVC   PARMAREA(REPLYLEN),REPLY
         MVC   PARMAREA+13(L'REPLDATA),CTXTTMSG+1Ø

         MVC   MGCRTEXT(REPLYLEN),PARMAREA
*        MVC   MGCRTEXT(REPLYLEN),REPLY

         LA    R3,(MGCRTEXT-MGCRPL)+REPLYLEN
         STC   R3,MGCRLGTH              SET THE LENGTH OF THE REPLY
         SR    RØ,RØ                    CLEAR RØ FOR SOME REASON
         MGCR  MGCRPL

         L     R1Ø,4(R13)               RETURN TO MPF
         FREEMAIN RU,LV=WORKEND,A=(R13),SP=23Ø
         LR    R13,R1Ø
         CLC   CTXTDCLN,=H'2'           ANY DESCRIPTOR CODES?
         BL    RETURN                   NO, FORGET IT
         L     R1Ø,CTXTDCP              POINTER TO DESCRIPTOR CODES
         USING CTXTDESC,R1Ø
         MVC   CTXTDC1(L'CTXTDC1+L'CTXTDC2),SETSTATS
         OI    CTXTRFB1,CTXTRCRC+CTXTRCDC  CHANGE CODES NOW
         DROP  R1Ø

RETURN   LM    R14,R12,12(R13)
         SR    R15,R15
         BR    R14

SETSTATS DC    AL1(CTXTDCØ4),X'ØØ'      SYSTEM STATUS WILL DO NICELY

REPLY    DC    C'S FTPSMTP,T='
         DC    C''''
REPLDATA DC    CLØ6Ø' '                  PARM DATA
         DC    C''''
REPLYLEN EQU *-REPLY

         LTORG

         PRINT NOGEN
WORKAREA DSECT
```

```
         DS    18F            SAVE AREA

DWORD    DS    D
         IEZMGCR DSECT=NO
         ORG

PARMAREA DS CL126

WORKEND  EQU   *-WORKAREA

         IEZVX1ØØ
*        PRINT NOGEN
         END   MPFTPEX
/*
//LKED1     EXEC PGM=IEWL,PARM='LIST,LET,MAP,XREF,RENT,REUS,REFR',
//          COND=(8,LT,ASMA9Ø)
//SYSUT1    DD  DSN=&&SYSUT1,UNIT=VIO,SPACE=(1Ø24,(5Ø,2Ø))
//SYSPRINT DD   SYSOUT=*,DCB=(RECFM=FB,LRECL=121,BLKSIZE=121Ø)
//SYSLIN    DD  DSN=&&OBJSET,DISP=(OLD,PASS)
//          DD  DDNAME=SYSIN
//SYSLMOD   DD  DSN=SYS1.LINKLIB,DISP=SHR
//SYSIN     DD *
  SETCODE  AC(1)
  NAME     MPFTPEX(R)
/*
```

FTPSMTP

```
//*-------------------------------------------------------------* FTPSMTP
//*                                                             * FTPSMTP
//* START   PROC FTPSMTP - PROCESS FTPSMFEX WTO MESSAGES        * FTPSMTP
//*         STARTED BY WTO EXIT MPFTPEX                         * FTPSMTP
//*                                                             * FTPSMTP
//*-------------------------------------------------------------* FTPSMTP
//*                                                               FTPSMTP
//XRAYFTPX PROC T=XXXXXXXX                                        FTPSMTP
//*                                                               FTPSMTP
//PSØØØ     EXEC PGM=IEFBR14,COND=(Ø,NE),                         FTPSMTP
//          PARM=('&T')                                           FTPSMTP
//*                                                               FTPSMTP
//PSØØØ     EXEC PGM=COMMAND,PARM='$P JOBQ,JM=FTPSMTP'            FTPSMTP
//          EXEC PGM=COMMAND,PARM='DELAY=1'                       FTPSMTP
//*                                                               FTPSMTP
//PSØ2Ø     EXEC PGM=HTSPFTPE,COND=(Ø,NE),REGION=6M,              FTPSMTP
//          PARM=('&T')                                           FTPSMTP
//*-------------------------------------------------------------* FTPSMTP
//*         PROCESS FTP SMF DATA                                * FTPSMTP
//*-------------------------------------------------------------* FTPSMTP
//STEPLIB   DD DSN=xxxxxxx.loadlib,DISP=SHR                       FTPSMTP
//*                                                               FTPSMTP
```

```
//HTSNFTPE  DD DSN=EDSSJ.HTS.MDVJFTPE,DISP=SHR                   FTPSMTP
//*                                                              FTPSMTP
//SYSTIN    DD DSN=&&STIN,DISP=(NEW,PASS,DELETE)                 FTPSMTP
//*                                                              FTPSMTP
//MESSAGE1  DD DSN=&&MSG1,DISP=(NEW,PASS,DELETE)                 FTPSMTP
//*                                                              FTPSMTP
//SYSPRINT DD  SYSOUT=T                                          FTPSMTP
//SYSDBOUT DD  SYSOUT=T                                          FTPSMTP
//SYSABOUT DD  SYSOUT=T                                          FTPSMTP
//SYSOUT   DD  SYSOUT=T                                          FTPSMTP
//*                                                              FTPSMTP
//PSØ2Ø    EXEC PGM=IKJEFT1B,COND=(Ø,NE)                         FTPSMTP
//*-----------------------------------------------               FTPSMTP
//*        SEND CONFIRMATION E-MAIL VIA SMTP                     FTPSMTP
//*-----------------------------------------------               FTPSMTP
//*                                                              FTPSMTP
//SYSEXEC  DD DISP=SHR,DSN=MVSXMITP.PROD.EXEC                    FTPSMTP
//SYSPRINT DD SYSOUT=T                                           FTPSMTP
//SYSTSPRT DD SYSOUT=T,DCB=(RECFM=FBA,LRECL=8Ø,BLKSIZE=8Ø)       FTPSMTP
//*                                                              FTPSMTP
//SYSTSIN  DD DSN=&&STIN,DISP=(OLD,PASS)                         FTPSMTP
//*                                                              FTPSMTP
//MESSAGE1 DD DSN=&MSG1,DISP=(OLD,PASS)                          FTPSMTP
//*                                                              FTPSMTP
//*-----------------------------------------------------------* FTPSMTP
//*                                                           * FTPSMTP
//* END OF PROC FTPSMTP -                                     * FTPSMTP
//*                                                           * FTPSMTP
//*-----------------------------------------------------------* FTPSMTP
```

HTSMFTP

```
*--------------------------------------------------------------------*
*                              H T S M F T P                         *
*--------------------------------------------------------------------*
*  JOSE L. RAMIREZ -- AUGUST 23,2ØØ1 -- FTP ACKNOWLEDGE              *
*
*  AUGUST 13, 2ØØ2 - FIX USER ID FIELD BUG...                        *
*
*--------------------------------------------------------------------*
***
HTSMFTP  DFHMSD TYPE=MAP,MODE=INOUT,LANG=COBOL,STORAGE=AUTO,
               TIOAPFX=YES,CTRL=FREEKB
*
MAPA1ØØ  DFHMDI SIZE=(24,8Ø),JUSTIFY=FIRST
         DFHMDF POS=(Ø1,Ø1),LENGTH=7,ATTRB=(ASKIP,BR),                X
               INITIAL='HTSMFTP'
         DFHMDF POS=(Ø1,27),LENGTH=25,ATTRB=(ASKIP,BR),               X
               INITIAL='INTERACTIVE SYSTEMS, INC.'
```

```
FECHA    DFHMDF POS=(Ø1,69),LENGTH=1Ø,ATTRB=(BR,ASKIP),                X
               PICOUT='99B99B9999'
*
         DFHMDF POS=(Ø2,27),LENGTH=27,ATTRB=(ASKIP,BR),                X
               INITIAL='** FTP Acknowledgment **'
*
HORA     DFHMDF POS=(Ø2,71),LENGTH=Ø8,ATTRB=(BR,ASKIP,PROT),           X
               PICOUT='99B99B99'
*
         DFHMDF POS=(4,1),LENGTH=4Ø,ATTRB=(ASKIP,BRT),                 X
               INITIAL='______________________________________________'
         DFHMDF POS=(4,4Ø),LENGTH=4Ø,ATTRB=(ASKIP,BRT),
               INITIAL='______________________________________________'
*
*** Client User Id
         DFHMDF POS=(6,4),LENGTH=15,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Client User Id:'
CUSERID  DFHMDF POS=(6,2Ø),LENGTH=Ø8,ATTRB=(IC,UNPROT,FSET,BLANK)
*
         DFHMDF POS=(6,29),LENGTH=1,ATTRB=(ASKIP,BRT),                 X
               INITIAL='|'
*** Reply To email address
         DFHMDF POS=(6,31),LENGTH=9,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Reply To:'
RTEMAIL  DFHMDF POS=(6,41),LENGTH=35,ATTRB=(UNPROT,FSET,BLANK)
         DFHMDF POS=(6,77),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
*** Carbon Copy 1 email addres
         DFHMDF POS=(7,36),LENGTH=4,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Cc1:'
C1EMAIL  DFHMDF POS=(7,41),LENGTH=35,ATTRB=(UNPROT,FSET,BLANK)
         DFHMDF POS=(7,77),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
*** Carbon Copy 2 email addres
         DFHMDF POS=(8,36),LENGTH=4,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Cc2:'
C2EMAIL  DFHMDF POS=(8,41),LENGTH=35,ATTRB=(UNPROT,FSET,BLANK)
         DFHMDF POS=(8,77),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
         DFHMDF POS=(Ø9,1),LENGTH=4Ø,ATTRB=(ASKIP,BRT),                X
               INITIAL='______________________________________________'
         DFHMDF POS=(Ø9,4Ø),LENGTH=4Ø,ATTRB=(ASKIP,BRT),               X
               INITIAL='______________________________________________'
*
         DFHMDF POS=(11,4),LENGTH=31,ATTRB=(ASKIP,BRT),                X
               INITIAL='Please enter your text message:'
*
         DFHMDF POS=(13,4),LENGTH=7,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Line 1:'
LINE1    DFHMDF POS=(13,12),LENGTH=6Ø,ATTRB=(UNPROT,FSET,BLANK)
```

```
         DFHMDF POS=(13,73),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
         DFHMDF POS=(14,4),LENGTH=7,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Line 2:'
LINE2    DFHMDF POS=(14,12),LENGTH=6Ø,ATTRB=(UNPROT,FSET,BLANK)
         DFHMDF POS=(14,73),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
         DFHMDF POS=(15,4),LENGTH=7,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Line 3:'
LINE3    DFHMDF POS=(15,12),LENGTH=6Ø,ATTRB=(UNPROT,FSET,BLANK)
         DFHMDF POS=(15,73),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
         DFHMDF POS=(16,4),LENGTH=7,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Line 4:'
LINE4    DFHMDF POS=(16,12),LENGTH=6Ø,ATTRB=(UNPROT,FSET,BLANK)
         DFHMDF POS=(16,73),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
         DFHMDF POS=(17,4),LENGTH=7,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Line 5:'
LINE5    DFHMDF POS=(17,12),LENGTH=6Ø,ATTRB=(UNPROT,FSET,BLANK)
         DFHMDF POS=(17,73),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
         DFHMDF POS=(18,4),LENGTH=7,ATTRB=(ASKIP,BRT),                 X
               INITIAL='Line 6:'
LINE6    DFHMDF POS=(18,12),LENGTH=6Ø,ATTRB=(UNPROT,FSET,BLANK)
         DFHMDF POS=(18,73),LENGTH=1,ATTRB=(ASKIP),INITIAL=' '
*
         DFHMDF POS=(19,1),LENGTH=41,ATTRB=(ASKIP,BRT),                X
               INITIAL='_______________________________________________'
         DFHMDF POS=(19,4Ø),LENGTH=4Ø,ATTRB=(ASKIP,BRT),               X
               INITIAL='_______________________________________________'
*
MENSA    DFHMDF POS=(22,Ø2),LENGTH=6Ø,ATTRB=(BRT,ASKIP)
         DFHMDF POS=(24,18),LENGTH=44,ATTRB=(ASKIP,BR),                X
               INITIAL='PF1 Add | PF2 Update | PF3 Delete | PF5 View'
         DFHMSD TYPE=FINAL
```

HTSOFTPE

```
ID DIVISION.
       PROGRAM-ID.      HTSOFTPE.
      *AUTHOR.          JOSE L RAMIREZ.
      *REMARKS.
      *----------------------------------------------------------------*
      *                                                                *
      *  This program updates the FTP Acknowledgement file.  The      *
      *  information in this file is used to reply to FTP clients     *
      *  via SMTP.
      ****
```

```
  ****
      **** BY Jose Ramirez (jramirez@ssspr.com)
****                                                    *
*                                                                *
*----------------------------------------------------------------*
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
*****************************************************************
*    WORKING-STORAGE  SECTION  FOLLOWS FOR THIS PROGRAM         *
*****************************************************************
 Ø1  FILLER                       PIC X(29)      VALUE
     ' WORKING STORAGE STARTS HERE '.

 Ø1  WS-ASKTIME                   PIC 9(15)   COMP-3 VALUE Ø.
 Ø1  WS-UPDATE-FLAG               PIC X(Ø1)   VALUE 'N'.
 Ø1  RESP-CODE                    PIC S9(Ø8)  VALUE +Ø    COMP.

 Ø1  COMMUNICATION-AREA           PIC X(77).

 Ø1  WS-OUT-REC.
     Ø5 WS-OUT-KEY                PIC X(ØØ8)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-TO           PIC X(Ø35)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-CC1          PIC X(Ø35)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-CC2          PIC X(Ø35)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-CC3          PIC X(Ø35)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-CC4          PIC X(Ø35)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-CC5          PIC X(Ø35)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-MSG1         PIC X(Ø6Ø)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-MSG2         PIC X(Ø6Ø)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-MSG3         PIC X(Ø6Ø)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-MSG4         PIC X(Ø6Ø)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-MSG5         PIC X(Ø6Ø)    VALUE SPACES.
     Ø5 WS-OUT-REPLY-MSG6         PIC X(Ø6Ø)    VALUE SPACES.
     Ø5 FILLER                    PIC X(222)    VALUE SPACES.

 Ø1  WS-EMAIL-INFO.
     Ø5 WS-ACTION                 PIC X(8Ø)     VALUE 'Action   :'.
     Ø5 WS-CLIENT-ID              PIC X(8Ø)     VALUE 'Client Id:'.
     Ø5 WS-RACF-ID                PIC X(8Ø)     VALUE 'Racf Id  :'.
     Ø5 WS-SEC-ADMIN              PIC X(8Ø)     VALUE 'Sec Admin:'.

 Ø1  WS-LENGTH                    PIC S9(Ø4)    VALUE +8ØØ  COMP.

 COPY HTSCFTP.
 COPY DFHAID.
 COPY DFHBMSCA SUPPRESS.

 LINKAGE SECTION.
 Ø1  DFHCOMMAREA                 PIC X(73).

*-----------------------------------------------------------------*
 PROCEDURE DIVISION.
*-----------------------------------------------------------------*

 ØØØØ-MAIN-RTN.

     EXEC CICS IGNORE CONDITION
          MAPFAIL
     END-EXEC.

     IF  EIBCALEN EQUAL ZEROS
         MOVE LOW-VALUES TO MAPA1ØØO
         PERFORM 6ØØØ-SEND-MAP
     ELSE
         PERFORM Ø1ØØ-PROCESS-RTN
     END-IF.

*-----------------------------------------------------------------*
 Ø1ØØ-PROCESS-RTN.
*-----------------------------------------------------------------*

     EXEC CICS HANDLE AID PF1    (Ø2ØØ-ADD-USER)
                          PF2    (Ø4ØØ-UPDATE-USER)
                          PF3    (Ø3ØØ-DELETE-USER)
                          PF5    (Ø8ØØ-VIEW-USER)
                          ENTER  (Ø5ØØ-ANY-KEY)
                          ANYKEY (Ø5ØØ-ANY-KEY)
                          CLEAR  (Ø7ØØ-END)
     END-EXEC.

     EXEC CICS RECEIVE MAP ('HTSMFTP')
                      INTO (MAPA1ØØI)
     END-EXEC.

*-----------------------------------------------------------------*
 Ø2ØØ-ADD-USER.
*-----------------------------------------------------------------*

     IF CUSERIDI = SPACES
     OR CUSERIDI = HIGH-VALUES
     OR CUSERIDI = LOW-VALUES
        MOVE 'Invalid Input, Please Verify' TO MENSAO
        PERFORM 6ØØØ-SEND-MAP
     END-IF.

     PERFORM Ø6ØØ-SET-FIELDS.

     EXEC CICS  WRITE
```

```
          DATASET ('HTSNFTPE')
          RIDFLD  (WS-OUT-KEY)
          FROM    (WS-OUT-REC)
          LENGTH  (WS-LENGTH)
          RESP    (RESP-CODE)
     END-EXEC.

     EVALUATE RESP-CODE
        WHEN ØØ
          MOVE 'NEW USER SUCESSFULLY ADDED'            TO MENSAO
        WHEN 14
          MOVE 'USER ALREADY DEFINED'                  TO MENSAO
        WHEN OTHER
          MOVE 'UNEXPECTED ERROR, CALL SYS PROGRAMMER' TO MENSAO
     END-EVALUATE.

     PERFORM 6ØØØ-SEND-MAP.

*----------------------------------------------------------------*
 Ø3ØØ-DELETE-USER.
*----------------------------------------------------------------*

     IF CUSERIDI = SPACES
     OR CUSERIDI = HIGH-VALUES
     OR CUSERIDI = LOW-VALUES
        MOVE 'Invalid Input, Please Verify' TO MENSAO
        PERFORM 6ØØØ-SEND-MAP
     END-IF.

     EXEC CICS READ UPDATE
          DATASET   ('HTSNFTPE')
          RIDFLD    (CUSERIDI)
          KEYLENGTH (Ø8)
          INTO      (WS-OUT-REC)
          LENGTH    (WS-LENGTH)
          RESP      (RESP-CODE)
     END-EXEC.

     IF RESP-CODE = ØØ
        EXEC CICS DELETE FILE('HTSNFTPE') END-EXEC
     END-IF.

     EVALUATE RESP-CODE
        WHEN ØØ
          MOVE 'USER HAS BEEN DELETED'                 TO MENSAO
        WHEN 13
          MOVE 'USER NOT FOUND IN DATABASE'            TO MENSAO
        WHEN OTHER
          MOVE 'UNEXPECTED ERROR, CALL SYS PROGRAMMER' TO MENSAO
     END-EVALUATE.
     PERFORM 6ØØØ-SEND-MAP.

*----------------------------------------------------------------*
 Ø4ØØ-UPDATE-USER.
*----------------------------------------------------------------*

     IF CUSERIDI = SPACES
     OR CUSERIDI = HIGH-VALUES
     OR CUSERIDI = LOW-VALUES
        MOVE 'Invalid Input, Please Verify' TO MENSAO
        PERFORM 6ØØØ-SEND-MAP
     END-IF.

     EXEC CICS READ UPDATE
          DATASET    ('HTSNFTPE')
          RIDFLD     (CUSERIDI)
          KEYLENGTH  (Ø8)
          INTO       (WS-OUT-REC)
          LENGTH     (WS-LENGTH)
          RESP       (RESP-CODE)
     END-EXEC.

     PERFORM Ø6ØØ-SET-FIELDS.

     IF RESP-CODE = ØØ
        EXEC CICS REWRITE FILE('HTSNFTPE')
                  LENGTH (WS-LENGTH)
                  FROM   (WS-OUT-REC)
        END-EXEC
     END-IF.

     EVALUATE RESP-CODE
        WHEN ØØ
          MOVE 'USER HAS BEEN UPDATED'                 TO MENSAO
        WHEN 13
          MOVE 'USER NOT FOUND IN DATABASE'            TO MENSAO
        WHEN OTHER
          MOVE 'UNEXPECTED ERROR, CALL SYS PROGRAMMER' TO MENSAO
     END-EVALUATE.

     PERFORM 6ØØØ-SEND-MAP.

*----------------------------------------------------------------*
 Ø5ØØ-ANY-KEY.
*----------------------------------------------------------------*

     MOVE 'Please enter a valid PF Key' TO MENSAO.
     MOVE SPACES TO CUSERIDI.
     PERFORM 6ØØØ-SEND-MAP.
```

```
*----------------------------------------------------------------*
 Ø6ØØ-SET-FIELDS.
*----------------------------------------------------------------*

     MOVE LOW-VALUES TO WS-OUT-REC.

     MOVE CUSERIDI TO WS-OUT-KEY.

     MOVE RTEMAILI TO WS-OUT-REPLY-TO.

     MOVE C1EMAILI TO WS-OUT-REPLY-CC1.
     MOVE C2EMAILI TO WS-OUT-REPLY-CC2.

     MOVE LINE1I   TO WS-OUT-REPLY-MSG1.
     MOVE LINE2I   TO WS-OUT-REPLY-MSG2.
     MOVE LINE3I   TO WS-OUT-REPLY-MSG3.
     MOVE LINE4I   TO WS-OUT-REPLY-MSG4.
     MOVE LINE5I   TO WS-OUT-REPLY-MSG5.
     MOVE LINE6I   TO WS-OUT-REPLY-MSG6.

*----------------------------------------------------------------*
 Ø7ØØ-END.
*----------------------------------------------------------------*

     EXEC CICS RETURN END-EXEC.

*----------------------------------------------------------------*
 Ø8ØØ-VIEW-USER.
*----------------------------------------------------------------*

     IF CUSERIDI = SPACES
     OR CUSERIDI = HIGH-VALUES
     OR CUSERIDI = LOW-VALUES
        MOVE 'Invalid Input, Please Verify' TO MENSAO
        PERFORM 6ØØØ-SEND-MAP
     END-IF.

     EXEC CICS READ
          DATASET   ('HTSNFTPE')
          RIDFLD    (CUSERIDI)
          KEYLENGTH (Ø8)
          INTO      (WS-OUT-REC)
          LENGTH    (WS-LENGTH)
          RESP      (RESP-CODE)
     END-EXEC.

     IF RESP-CODE = ØØ
        MOVE WS-OUT-KEY        TO CUSERIDO
```

```
              MOVE WS-OUT-REPLY-TO   TO RTEMAILO

              MOVE WS-OUT-REPLY-CC1  TO C1EMAILO
              MOVE WS-OUT-REPLY-CC2  TO C2EMAILO

              MOVE WS-OUT-REPLY-MSG1 TO LINE1O
              MOVE WS-OUT-REPLY-MSG2 TO LINE2O
              MOVE WS-OUT-REPLY-MSG3 TO LINE3O
              MOVE WS-OUT-REPLY-MSG4 TO LINE4O
              MOVE WS-OUT-REPLY-MSG5 TO LINE5O
              MOVE WS-OUT-REPLY-MSG6 TO LINE6O
           ELSE
      **      MOVE LOW-VALUES        TO MAPA1ØØO
              MOVE CUSERIDI          TO MENSAO(1:8)
              MOVE 'not in Database' TO MENSAO(1Ø:15)
           END-IF.

      *-----------------------------------------------------------------*
       6ØØØ-SEND-MAP.
      *-----------------------------------------------------------------*

           EXEC CICS ASKTIME
                     ABSTIME(WS-ASKTIME)
           END-EXEC.

           EXEC CICS FORMATTIME
                    ABSTIME(WS-ASKTIME)
                    DATESEP ('/')
                    MMDDYYYY(FECHAO)
                    TIMESEP (':')
                    TIME    (HORAO)
           END-EXEC.

           EXEC CICS SEND   MAP('HTSMFTP')
                            FROM(MAPA1ØØO)
                            ERASE
           END-EXEC.

           EXEC CICS RETURN TRANSID('FTPE')
                            COMMAREA(COMMUNICATION-AREA)
                            LENGTH(77)
           END-EXEC.
```

HTSPFTPE

```
IDENTIFICATION DIVISION.

       PROGRAM-ID. HTSPFTPE.
```

```
*****************************************************************
*  AUTHOR. JOSE RAMIREZ.
* (jramirez@ssspr.com)
*
*****************************************************************
/
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

     SELECT SYSTIN-FILE          ASSIGN          SYSTIN
                                 FILE STATUS     SYSTIN-STATUS.

     SELECT MESSAGE1-FILE        ASSIGN          MESSAGE1
                                 FILE STATUS     MESSAGE1-STATUS.

     SELECT FTP-ACK-FILE         ASSIGN          HTSNFTPE
                                 ORGANIZATION    INDEXED
                                 ACCESS          DYNAMIC
                                 RECORD KEY      FTP-ACK-KEY
                                 FILE STATUS     FTP-RESP-CODE.

 DATA DIVISION.
 FILE SECTION.
/
*---------------------------------------------------------------*
 FD  SYSTIN-FILE
*---------------------------------------------------------------*
     LABEL RECORDS ARE STANDARD
     RECORDING MODE IS F
     RECORD CONTAINS 8Ø CHARACTERS
     BLOCK CONTAINS Ø RECORDS
     .

 Ø1  SYSTIN-RECORD.

     1Ø  SYSTIN-REC              PIC X(8Ø).
/
*---------------------------------------------------------------*
 FD  MESSAGE1-FILE
*---------------------------------------------------------------*
     LABEL RECORDS ARE STANDARD
     RECORDING MODE IS F
     RECORD CONTAINS 8Ø CHARACTERS
     BLOCK CONTAINS Ø RECORDS
     .

 Ø1  MESSAGE1-RECORD.
```

```
     1Ø  MESSAGE1-REC            PIC X(8Ø).
/
*----------------------------------------------------------------*
 FD  FTP-ACK-FILE
*----------------------------------------------------------------*
     RECORD CONTAINS 8ØØ CHARACTERS.

 Ø1  FTP-ACK-RECORD.

     Ø5 FTP-ACK-KEY              PIC X(ØØ8).
     Ø5 FTP-ACK-REPLY-TO         PIC X(Ø35).
     Ø5 FTP-ACK-REPLY-CC1        PIC X(Ø35).
     Ø5 FTP-ACK-REPLY-CC2        PIC X(Ø35).
     Ø5 FTP-ACK-REPLY-CC3        PIC X(Ø35).
     Ø5 FTP-ACK-REPLY-CC4        PIC X(Ø35).
     Ø5 FTP-ACK-REPLY-CC5        PIC X(Ø35).
     Ø5 FTP-ACK-REPLY-MSG1       PIC X(Ø6Ø).
     Ø5 FTP-ACK-REPLY-MSG2       PIC X(Ø6Ø).
     Ø5 FTP-ACK-REPLY-MSG3       PIC X(Ø6Ø).
     Ø5 FTP-ACK-REPLY-MSG4       PIC X(Ø6Ø).
     Ø5 FTP-ACK-REPLY-MSG5       PIC X(Ø6Ø).
     Ø5 FTP-ACK-REPLY-MSG6       PIC X(Ø6Ø).
     Ø5 FILLER                   PIC X(222).
/
*-----------------------------------------------------------------*
 WORKING-STORAGE SECTION.
*-----------------------------------------------------------------*

/
*---------------------------------------------------------------*
 Ø1  WS-WORK-AREAS.
*---------------------------------------------------------------*

     1Ø  WS-BYCOUNT              PIC  9(Ø8) VALUE ZEROES.
     1Ø  WS-FLAG                 PIC  X(Ø1) VALUE 'Ø'.
     1Ø  FTP-RESP-CODE           PIC  X(Ø2) VALUE SPACES.
     1Ø  SYSTIN-STATUS           PIC  X(Ø2) VALUE SPACES.
     1Ø  MESSAGE1-STATUS         PIC  X(Ø2) VALUE SPACES.
     1Ø  WS-FROM                 PIC  X(35)
         VALUE 'youremail@ssspr.com'.
     1Ø  WS-SUBJECT              PIC  X(35)
         VALUE "'FTP Confirmation'".
     1Ø  STATUS-DATA             PIC  X(Ø2) VALUE SPACES.
     1Ø  STATUS-PROCP            PIC  X(Ø2) VALUE SPACES.
     1Ø  STATUS-REST             PIC  X(Ø2) VALUE SPACES.
     1Ø  STATUS-TOTM             PIC  X(Ø2) VALUE SPACES.
     1Ø  STATUS-TOTC             PIC  X(Ø2) VALUE SPACES.
     1Ø  STATUS-CTLQ             PIC  X(Ø2) VALUE SPACES.

/
```

```
*-----------------------------------------------------------------*
 LINKAGE SECTION.
*-----------------------------------------------------------------*

 Ø1  LS-PARAMETERS.

     1Ø  LS-LENGTH                PIC S9(Ø4) COMP.

     1Ø  LS-USERID                PIC X(Ø8).

     1Ø  LS-DSN                   PIC X(4Ø).

     1Ø  LS-PDS                   PIC X(Ø8).

     1Ø  LS-BYCOUNT               PIC 9(Ø8) COMP.
/
*-----------------------------------------------------------------*
 PROCEDURE DIVISION USING LS-PARAMETERS.
*-----------------------------------------------------------------*
 PØØØ-MAINLINE.
*-----------------------------------------------------------------*

     PERFORM P1ØØ-PROCESS-USER-RECORD.

     PERFORM P999-SET-RETURN-CODE.

     STOP RUN.

/
*-----------------------------------------------------------------*
 P1ØØ-PROCESS-USER-RECORD.
*-----------------------------------------------------------------*

     PERFORM P2ØØ-OPEN-FILES.

     PERFORM P3ØØ-GET-USER-RECORD.

     PERFORM P4ØØ-CREATE-SYSTIN-RECORD.

     PERFORM P5ØØ-CREATE-MESSAGE-RECORD.

/
*-----------------------------------------------------------------*
 P2ØØ-OPEN-FILES.
*-----------------------------------------------------------------*

     OPEN INPUT  FTP-ACK-FILE.
     OPEN OUTPUT SYSTIN-FILE, MESSAGE1-FILE.

     IF FTP-RESP-CODE NOT EQUAL 'ØØ'
```

```
          DISPLAY 'ERROR OPENING FTP ACK FILE:' FTP-RESP-CODE
          MOVE '1' TO WS-FLAG
       END-IF.

       IF SYSTIN-STATUS NOT EQUAL 'ØØ'
          DISPLAY 'ERROR OPENING SYSTIN FILE:' SYSTIN-STATUS
          MOVE '1' TO WS-FLAG
       END-IF.

       IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
          DISPLAY 'ERROR OPENING MESSAGE1 FILE:' MESSAGE1-STATUS
          MOVE '1' TO WS-FLAG
       END-IF.

 /
 *-----------------------------------------------------------------*
  P3ØØ-GET-USER-RECORD.
 *-----------------------------------------------------------------*

       MOVE LS-USERID TO FTP-ACK-KEY.

       DISPLAY 'LS-USERID ', LS-USERID.

       DISPLAY 'LS-BYCOUNT ', LS-BYCOUNT.

       READ FTP-ACK-FILE.

       EVALUATE FTP-RESP-CODE
         WHEN 'ØØ'
            DISPLAY 'ABOUT TO PROCESS USER... ' LS-USERID
         WHEN '23'
            DISPLAY 'USER NOT IN FILE, ENDING PROGRAM ...'
            MOVE '1' TO WS-FLAG
         WHEN OTHER
            DISPLAY 'ERROR IN FTP ACK FILE, ENDING..', FTP-RESP-CODE
            MOVE '1' TO WS-FLAG
       END-EVALUATE.

 /
 *-----------------------------------------------------------------*
  P4ØØ-CREATE-SYSTIN-RECORD.
 *-----------------------------------------------------------------*

       INITIALIZE SYSTIN-RECORD.
       MOVE 'XMITIP'         TO SYSTIN-REC (Ø2:Ø6).
       MOVE '('              TO SYSTIN-REC (11:Ø1).
       MOVE FTP-ACK-REPLY-TO TO SYSTIN-REC (12:35).
       MOVE '-'              TO SYSTIN-REC (71:Ø1).
       WRITE SYSTIN-RECORD.
```

```
IF SYSTIN-STATUS NOT EQUAL 'ØØ'
   DISPLAY 'ERROR WRITING TO SYSTIN FILE1:' SYSTIN-STATUS
   MOVE '1' TO WS-FLAG
END-IF.

INITIALIZE SYSTIN-RECORD.
MOVE ')'               TO SYSTIN-REC (12:Ø1).
MOVE '-'               TO SYSTIN-REC (71:Ø1).
WRITE SYSTIN-RECORD.

IF SYSTIN-STATUS NOT EQUAL 'ØØ'
   DISPLAY 'ERROR WRITING TO SYSTIN FILE2:' SYSTIN-STATUS
   MOVE '1' TO WS-FLAG
END-IF.

IF FTP-ACK-REPLY-CC1 NOT EQUAL LOW-VALUES
   PERFORM P6ØØ-BUILD-CC
END-IF.

INITIALIZE SYSTIN-RECORD.
MOVE 'FROM'            TO SYSTIN-REC (12:Ø4).
MOVE WS-FROM           TO SYSTIN-REC (17:35).
MOVE '-'               TO SYSTIN-REC (71:Ø1).
WRITE SYSTIN-RECORD.

IF SYSTIN-STATUS NOT EQUAL 'ØØ'
   DISPLAY 'ERROR WRITING TO SYSTIN FILE3:' SYSTIN-STATUS
   MOVE '1' TO WS-FLAG
END-IF.

INITIALIZE SYSTIN-RECORD.
MOVE 'SUBJECT'         TO SYSTIN-REC (12:Ø7).
MOVE WS-SUBJECT        TO SYSTIN-REC (2Ø:35).
MOVE '-'               TO SYSTIN-REC (71:Ø1).
WRITE SYSTIN-RECORD.

IF SYSTIN-STATUS NOT EQUAL 'ØØ'
   DISPLAY 'ERROR WRITING TO SYSTIN FILE4:' SYSTIN-STATUS
   MOVE '1' TO WS-FLAG
END-IF.

INITIALIZE SYSTIN-RECORD.
MOVE 'MSGDD'           TO SYSTIN-REC (12:Ø5).
MOVE 'MESSAGE1'        TO SYSTIN-REC (21:Ø8).
WRITE SYSTIN-RECORD.

IF SYSTIN-STATUS NOT EQUAL 'ØØ'
   DISPLAY 'ERROR WRITING TO SYSTIN FILE5:' SYSTIN-STATUS
   MOVE '1' TO WS-FLAG
END-IF.
```

```
/
*------------------------------------------------------------------*
 P5ØØ-CREATE-MESSAGE-RECORD.
*------------------------------------------------------------------*

     MOVE FTP-ACK-REPLY-MSG1 TO MESSAGE1-REC.
     WRITE MESSAGE1-RECORD.

     IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
        DISPLAY 'ERROR WRITING TO MESSAGE FILE1:' MESSAGE1-STATUS
        MOVE '1' TO WS-FLAG
     END-IF.

     MOVE FTP-ACK-REPLY-MSG2 TO MESSAGE1-REC.
     WRITE MESSAGE1-RECORD.

     IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
        DISPLAY 'ERROR WRITING TO MESSAGE FILE2:' MESSAGE1-STATUS
        MOVE '1' TO WS-FLAG
     END-IF.

     MOVE FTP-ACK-REPLY-MSG3 TO MESSAGE1-REC.
     WRITE MESSAGE1-RECORD.

     IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
        DISPLAY 'ERROR WRITING TO MESSAGE FILE3:' MESSAGE1-STATUS
        MOVE '1' TO WS-FLAG
     END-IF.

     MOVE FTP-ACK-REPLY-MSG4 TO MESSAGE1-REC.
     WRITE MESSAGE1-RECORD.

     IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
        DISPLAY 'ERROR WRITING TO MESSAGE FILE4:' MESSAGE1-STATUS
        MOVE '1' TO WS-FLAG
     END-IF.

     MOVE FTP-ACK-REPLY-MSG5 TO MESSAGE1-REC.
     WRITE MESSAGE1-RECORD.

     IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
        DISPLAY 'ERROR WRITING TO MESSAGE FILEA:' MESSAGE1-STATUS
        MOVE '1' TO WS-FLAG
     END-IF.

     MOVE FTP-ACK-REPLY-MSG6 TO MESSAGE1-REC.
     WRITE MESSAGE1-RECORD.

     IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
```

```
              DISPLAY 'ERROR WRITING TO MESSAGE FILEB:' MESSAGE1-STATUS
              MOVE '1' TO WS-FLAG
           END-IF.

           INITIALIZE MESSAGE1-REC.
           WRITE MESSAGE1-RECORD.

           IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
              DISPLAY 'ERROR WRITING TO MESSAGE FILE5:' MESSAGE1-STATUS
              MOVE '1' TO WS-FLAG
           END-IF.

           MOVE ALL '_' TO MESSAGE1-REC.
           WRITE MESSAGE1-RECORD.

           IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
              DISPLAY 'ERROR WRITING TO MESSAGE FILE6:' MESSAGE1-STATUS
              MOVE '1' TO WS-FLAG
           END-IF.

           INITIALIZE MESSAGE1-REC.
           WRITE MESSAGE1-RECORD.

           IF MESSAGE1-STATUS NOT EQUAL 'ØØ'
              DISPLAY 'ERROR WRITING TO MESSAGE FILE7:' MESSAGE1-STATUS
              MOVE '1' TO WS-FLAG
           END-IF.

           MOVE 'USER ID       :' TO MESSAGE1-REC (1:15).
           MOVE LS-USERID         TO MESSAGE1-REC (16:8).
           WRITE MESSAGE1-RECORD.

           MOVE 'DSN           :' TO MESSAGE1-REC (Ø1:15).
           MOVE LS-DSN            TO MESSAGE1-REC (16:4Ø).
           WRITE MESSAGE1-RECORD.

           MOVE 'PDS           :' TO MESSAGE1-REC (Ø1:15).
           MOVE LS-PDS            TO MESSAGE1-REC (16:Ø8).
           WRITE MESSAGE1-RECORD.

           MOVE 'BYTE COUNT    :' TO MESSAGE1-REC (Ø1:15).
           MOVE LS-BYCOUNT        TO MESSAGE1-REC (16:Ø8).
           WRITE MESSAGE1-RECORD.

      *-----------------------------------------------------------------*
       P6ØØ-BUILD-CC.
      *-----------------------------------------------------------------*

           INITIALIZE SYSTIN-RECORD.
           MOVE 'CC ('            TO SYSTIN-REC(12:Ø4).
           MOVE '-'                TO SYSTIN-REC (71:Ø1).
           WRITE SYSTIN-RECORD.

           INITIALIZE SYSTIN-RECORD.
           MOVE FTP-ACK-REPLY-CC1 TO SYSTIN-REC (16:35).
           MOVE '-'                TO SYSTIN-REC (71:Ø1).
           WRITE SYSTIN-RECORD.

           IF FTP-ACK-REPLY-CC2 NOT EQUAL LOW-VALUES
              INITIALIZE SYSTIN-RECORD
              MOVE FTP-ACK-REPLY-CC2 TO SYSTIN-REC (16:35)
              MOVE '-'                TO SYSTIN-REC (71:Ø1)
              WRITE SYSTIN-RECORD
           END-IF.

           INITIALIZE SYSTIN-RECORD.
           MOVE ')'               TO SYSTIN-REC (12:Ø1).
           MOVE '-'               TO SYSTIN-REC (71:Ø1).
           WRITE SYSTIN-RECORD.

      *-----------------------------------------------------------------*
       P999-SET-RETURN-CODE.
      *-----------------------------------------------------------------*

           IF WS-FLAG = '1'
              MOVE 8 TO RETURN-CODE
           END-IF.

      /
```

José Luis Ramirez
OS/390 Systems Administrator, Triple-S (USA)

What does HPR have to offer?

High Performance Routing (HPR) is an extension to the APPN architecture which can be implemented on an APPN network node or an APPN end node. HPR doesn't change the basic functions of the architecture, but provides enhancements to APPN.

HPR is a new SNA technology that surpasses today's dominant LAN interconnection technologies in performance, dynamic rerouting, priority and class of service, congestion avoidance, scability, and economy. A superset of APPN, HPR is fully compatible with AIW-standard APPN. HPR features a simple drop-in migration into existing SNA networks with no complex configuration or hardware upgrades.

HPR's Automatic Network Routing (ANR) is much faster than other existing dominant routing protocols. It eliminates error recovery and flow control overhead from interior nodes, allowing products to deliver price/performance without compromising mission-critical features like priority and class of service. ANR is a sophisticated new source-routing method developed by IBM for its gigabit-speed Nways products with Networking Broadband Services.

HPR's Rapid Transport Protocol safely reroutes data around failed links or nodes. HPR also eliminates the performance bottlenecks of other LAN-interconnect protocols. Selective retransmission lets HPR drive fast links at near capacity. By contrast, TCP/IP Data Link Switch and LAN 802.2 LLC Type 2 (remote source route bridging) performance degrades at high speeds because of their frequent responses.

Many network managers worry about congestion, the main cause of data loss, and chaotic performance in IP-based routers. The usual, costly, solution is excess link capacity. With Adaptive Rate Based (ARB) congestion avoidance, HPR networks do not succumb to congestion. Instead, HPR adaptively meters data at the highest sustainable rates, handling peak loads economically.

HPR eclipses other protocols by guaranteeing the stable, predictable performance required for mission-critical networking. Its full native support for SNA Class of Service and priority satisfies each application's special requirements. HPR networks are scalable to many thousands of nodes as a result of IBM's APPN directory services with Central Directory Server and Border Node.

Native SNA routing provided by HPR stands in sharp contrast to Data Link Switching, which is inherently limited by its reliance on the popular but ageing TCP/IP protocol as a transport.

VTAM EXAMPLE DEFINITION FOR EE BETWEEN HOST

ATCSTART

```
   BN=YES,
   CDSERVR=YES,
   CONNTYPE=LEN,
   CPCP=YES,
   HPR=(RTP,RTP),
   HPRARB=RESPMODE,
   IPADDR=1Ø.1ØØ.1Ø.1ØØ,
   NODETYPE=NN,
   SACONNS=YES,
   TCPNAME=TCPIP,
   SORDER=APPN,
```

Major nodes VTAM XCA

```
*
 *- XCA FOR ENTERPRISE EXTENDER
 *

 XCATCPIP VBUILD TYPE=XCA
 PORTXCAT PORT  MEDIUM=HPRIP,
                IPTOS=(2Ø,4Ø,8Ø,CØ,CØ),
                LIVTIME=1Ø,
                IPPORT=12ØØØ,
                SAPADDR=4,
                SRQTIME=15,
                SRQRETRY=3
 GRPXCAT   GROUP DIAL=YES,
                ISTATUS=ACTIVE,
                AUTOGEN=(2,LNEE,PUEE),
                CALL=INOUT
```

Major nodes SWITCHED

```
*
*    ENTERPRISE EXTENDER SU XCA HPRIP DEVELOPMENT ENVIRONMENT
*
         VBUILD TYPE=SWNET
PUEEBPLS PU     CAPACITY=64K,
                PUTYPE=2,
                DYNLU=YES,
                CONNTYPE=APPN,
                CPNAME=VTMXCAT,
                DWACT=YES,
                ISTATUS=ACTIVE,
                TGP=TRING16M
         PATH   IPADDR=1Ø.18Ø.8Ø.1Ø,
                CALL=INOUT,
                GRPNM=GRPXCAT,
                SAPADDR=16
```

ADJCLUST

```
*
         VBUILD TYPE=ADJCLUST
ITICRIØØ NETWORK NETID=ITICCIØØ
         NEXTCP  CPNAME=ITICCIØØ.ITICCIØA
IT5151ØØ NETWORK NETID=IT5151ØØ
         NEXTCP  CPNAME=IT5151ØØ.VTM5151B
```

Fabio Lolli
Systems Programmer (Italy)

UDDI – why you should register today!

Although Universal Description, Discovery and Integration (UDDI) is closely associated with Web services, it's not by any means restricted to the Web services scenario. Rather, it relates to a global Web-based business directory that has been operational since late 2000, whose scope covers the activities of all enterprises – irrespective of whether or not they relate to e-business, let alone Web services! This may come as a shock to anyone who thought there was no need to worry about UDDI until they started thinking about Web services. Far from it: there are three very good reasons why you should register your enterprise in the global UDDI Business Registry (UBR) at once, at uddi.ibm.com, uddi.microsoft.com, or uddi.sap.com:

- As with domain names in the past, if you don't register your company now, somebody else could do so and say whatever they wish about your company. It will then take time and effort to get the listing rectified.
- The UBR, based on the UDDI standard, provides a powerful and flexible mechanism to showcase your enterprise's capabilities – complete with contact information and Web-based information on how others can go about using any of the services you offer.
- There is currently no charge for UBR registration, although this is likely to change in the future when the registries start enforcing authentication mechanisms such as digital certificates.

DON'T LET SOMEBODY STEAL *YOUR* REGISTRATION

The key urgency for registering with the UBR without delay is that the current procedures effectively let anyone register a previously unregistered entity, irrespective of their affiliation with the entity being registered. It's therefore possible at the moment for one of your competitors to go ahead and register your company and make derogatory statements about the services your company offers! The easiest way to prevent this happening is to register your enterprise today, yourself, so that you become the owner-of-record when it comes to that registration.

IBM, Microsoft, SAP (as in ERP market leaders), and NTT Communications of Japan collaboratively run the UBR, which consists of four registry nodes working as a single, distributed directory. Within this UBR, you'll find that IBM lists 170 service categorizations that it offers, ranging from 'optical jukebox server software' to 'sales financing'. IBM's list also includes 'spreadsheet software', 'word processing software', and 'presentation software'. By contrast, Microsoft (which, with IBM, is an instigator of both the UDDI standard and the UBR) categorizes itself simply as a 'software publisher' (which also happens to be another of the 170 categories that IBM uses to describe itself). This demonstrates why it's important to get your stake with the UBR today: you don't want anybody else getting there before you and categorizing the services offered by your enterprise as something that they aren't.

This ability to make arbitrary registrations isn't really a flaw. The underlying rationale for the UBR is that it is meant to be public, open-to-all, and egalitarian. It is, by definition, meant to be a public repository. The UBR nodes insist that you register with them before you can publish an entry. In the case of IBM, this involves selecting a previously unused userid and an appropriate password, and providing a set of contact information with a valid e-mail address. Your registration is activated only when you invoke a link sent to the e-mail address provided. This e-mail-based validation approach isn't new, and has been widely used in the last five years or so as a minimally acceptable, baseline method.

The rationale is that the service provider (ie the UBR node operator) has at least an e-mail address which once worked, that points to the user trying to register. However, this e-mail-based validation isn't really worth the electrons used to execute it! A wily hacker will have no problems opening numerous, hard-to-track e-mail accounts. Microsoft insists that registration to use its UBR node has to be made via its now severely tarnished and undermined Passport service. Down the road there will be digital certificates and digital signatures to provide better authentication.

The latest version of the UDDI standard – UDDI Version 3 – supports these mechanisms. It now seems inevitable that in the future, all the UBR node operators will insist on third-party certification, via digital certificates, before anyone can register as a *bona fide* UBR publisher. And that's when some nominal fee for UBR registrations will probably be introduced. Until this happens, there will always be concerns about the integrity of the information available in the UBR. The way to prevent your company getting misrepresented is to prevent anybody else from getting there before you.

Once an entry is published, only the user that published that entry can modify or delete it. This is good and reassuring, assuming that the original user that did the publishing was legitimate and was who they actually claimed to be. The original publisher, through the custody transfer API set, can also assign ownership to another registered user. The key point here is that the original publisher has all the initial rights. This is why it's so imperative to ensure, via digital certificates, that the original publisher is not a charlatan. Once there's a validated publisher, a UBR node's policies could further dictate that it will accept only digitally signed entries from that user. Once such policies are implemented, we'll be able to put stock in the integrity of the information maintained in the UBR. During the transition to a digitally verified UBR structure, it will be possible, with Version 3 implementations, for users to request that the searches they conduct on the UBR are restricted to those entries whose authenticity has been digitally verified.

PRIVATE UDDI REGISTRIES AND GREEN PAGES

Both IBM and Microsoft are also actively promoting the notion of private UDDI implementations:

- Private intranet UDDI implementations will serve just one enterprise.

- Private extranet UDDI implementations will serve a group of enterprises collaborating with each other as partners – typically for e-business.

Microsoft's new Windows Server 2003, which is now shipping, includes a full-blown enterprise UDDI capability for realizing intranet or extranet implementations. IBM offers a similar capability in its new WebSphere Application Server Network Deployment (WAS ND) Version 5, which has subsumed the previously available WebSphere UDDI Registry offering. UDDI-related development tools (some of which include private UDDI registries) are also available from a wide variety of other vendors, including BEA, Sun, Novell, Fujitsu, IONA, and Cape Clear Software.

With the ready availability of these products and tools, there will now be a surge in intranet and extranet UDDI implementations as organizations gingerly start to evaluate this latest development in pan-enterprise electronic directories – but one focused on enterprise identity, mission, and services as opposed to enterprise resources and personnel. Intranet/extranet UDDI registries will also let enterprises determine the true potential and power of Web services by using them 'in-house' and in conjunction with trusted partners, for pilot projects. Enterprise UDDI registries will enable these 'not-ready-for-prime-time-as-yet' Web services to still be properly categorized and represented, using standard UDDI mechanisms, within the privacy of intranets and extranets. Recognizing the appeal and applicability of enterprise UDDI registries, the latest UDDI specification sets out to specify the notion of 'UDDI registry interaction' – ie how private UDDI registries can coexist and interoperate with the public UBR.

Since UDDI is a directory of businesses and organizations that provide various services, it's not surprising that it's invariably thought of as a kind of electronic telephone directory – particularly an electronic *Yellow Pages* of sorts. In actual fact, UDDI is much more than just a yellow page directory that's organized purely by service type. UDDI also is a white page directory, in that it lists service providers by name. The information contained in UDDI is best thought of as being divided into three distinct categories, referred to in telephone directory parlance as:

- *White pages*, containing descriptive information about businesses and organizations providing various services. The business or organization name and a textual description of who they are will be included – where appropriate in multiple languages. Contact information for that entity will also be included, in terms of contact names, phone numbers, fax-numbers, e-mails, and URLs. It will also list any known ‘industrial’ identifiers such as a Dun & Bradstreet (D&B) D-U-N-S, 9-digit identification sequence that uniquely identifies a particular business. The Thomas Registry identification scheme is another option.
- *Yellow pages*, which categorize businesses and organizations according to industry-standard taxonomies. At a minimum, businesses and organizations will be categorized in terms of their industry, the products and services they offer, and their geographic location. The North American Industry Classification System (NAICS), which has superseded the US Standard Industrial Classification (SIC) system, is one of the taxonomies that can be used to specify industry classification. Similarly, the United Nations Standard Products and Services Code System (UNSPSC) can be used to specify product/service classifications. UDDI is also innately extensible. It therefore permits the use of new taxonomies, provided that all users of the UDDI registry can interpret the classification scheme used.
- *Green pages*, containing the technical information needed in order to use an available service. This information specifies how a service may be invoked and, in the case of electronic services, the binding information necessary to activate that service.

THE UDDI MODEL

The mission of UDDI is to provide a standard, uniform service, readily accessible by applications via a programmatic interface or by humans via a GUI, for describing and locating:

- Businesses and organizations that offer various services – Web services being one of them.
- Meaningful description of the services being made available by the businesses and organizations listed as service providers.
- Technical information as to how to locate, access, and use a particular service.

UDDI is highly XML-centric. The core information model used by UDDI, irrespective of the kind of service being described, is based on an XML schema. This UDDI XML schema deals with the following four types of key information as it relates to service providers and the services they offer, whether Web services or otherwise:

- Business information pertaining to a business or organization providing one or more services – which, in the context of the UDDI model, is referred to as the 'businessEntity'.
- Sets of related services (eg a set of Web services to do with purchase order processing and another set to do with on-line inventory checking) being offered by a previously described business or organization – which are referred to as 'businessService' elements.
- The binding information necessary to invoke and make use of a particular, previously described, service (whether a Web service or not) – referred to as a 'bindingTemplate' element.
- More technical information (or even a technical blueprint) about a service, over and above the binding information necessary to connect to it, such as pointers to detailed specifications, protocols used by the service (eg SOAP, http, or SMTP in the case of a Web service), and possible type classification for that service – which is known as a 'technical model' (tModel).

There is a prescribed hierarchy among these four data structures (see Figure 1). These core data structures have been a part of UDDI from the beginning, and form the nucleus of what UDDI is

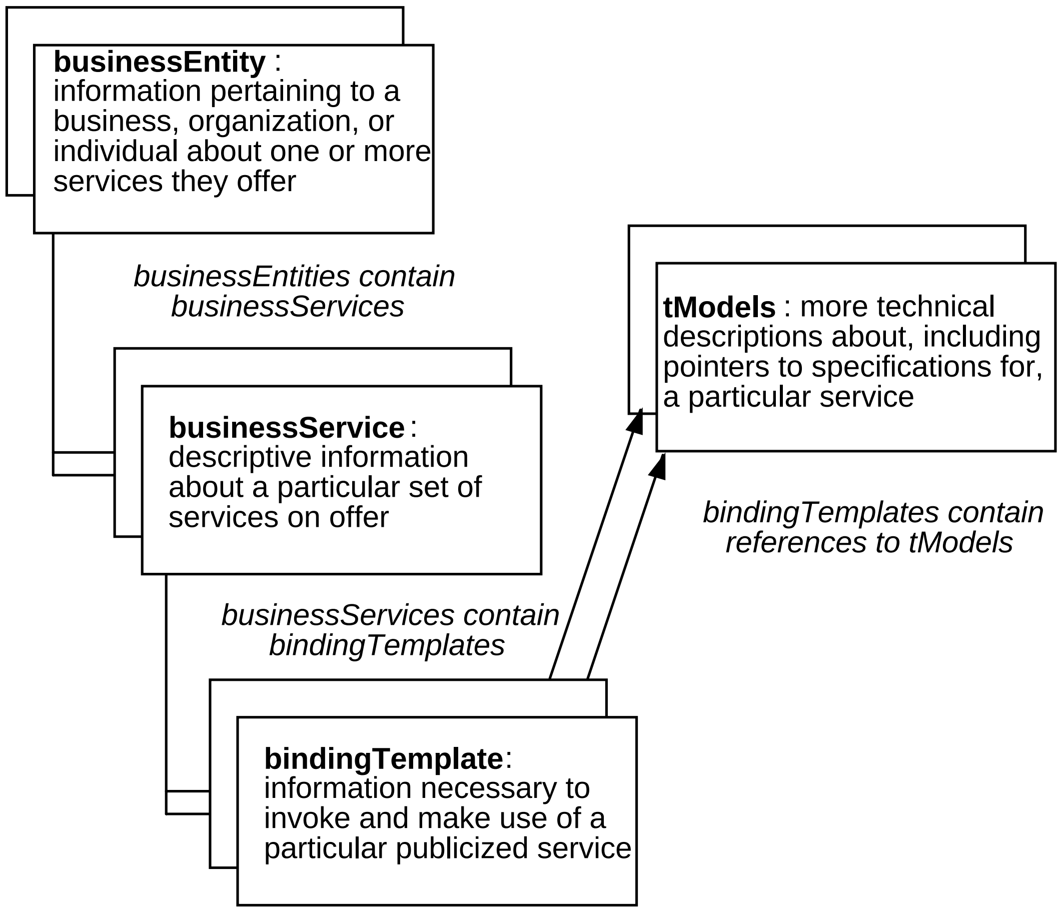
businessEntity : information pertaining to a business, organization, or individual about one or more services they offer
businessEntities contain businessServices
businessService : descriptive information about a particular set of services on offer
businessServices contain bindingTemplates
bindingTemplate: information necessary to invoke and make use of a particular publicized service
tModels : more technical descriptions about, including pointers to specifications for, a particular service
bindingTemplates contain references to tModels

Figure 1: The four key UDDI data structures based on the model shown in various UDDI documents

all about. Two other data structures have been added since, one in UDDI Version 2 and the other in Version 3. These new data structures deal with:

- Relationships between business or organization entities (eg certification, alliances, membership, trading partnerships etc) asserted by one or both of those entities. This data structure, referred to as 'publisher Assertion', was introduced with UDDI Version 2.
- Standing orders subscribed to by companies, organizations,

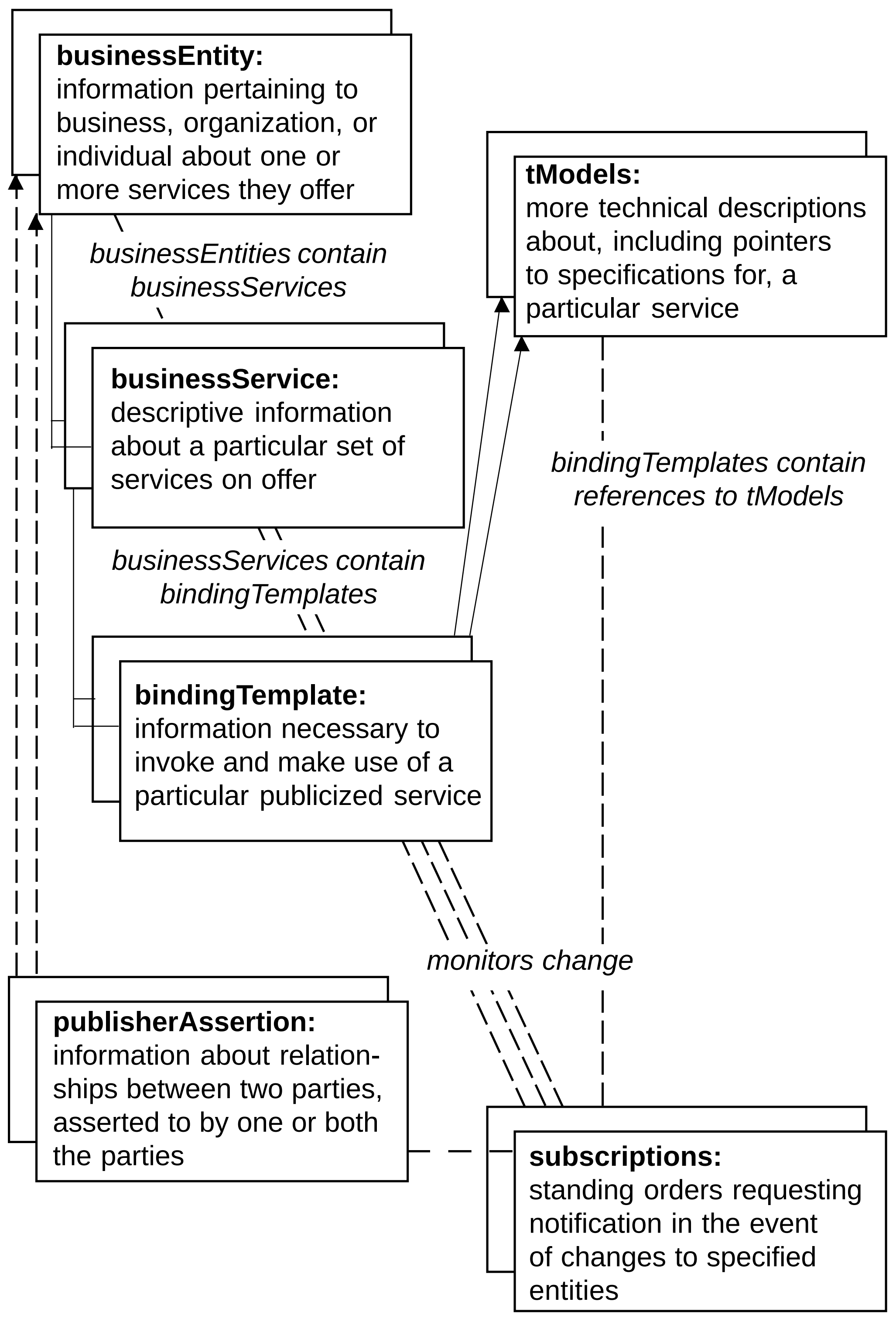
businessEntity:
information pertaining to business, organization, or individual about one or more services they offer
businessEntities contain businessServices
tModels:
more technical descriptions about, including pointers to specifications for, a particular service
businessService:
descriptive information about a particular set of services on offer
bindingTemplates contain references to tModels
businessServices contain bindingTemplates
bindingTemplate:
information necessary to invoke and make use of a particular publicized service
monitors change
publisherAssertion:
information about relation-ships between two parties, asserted to by one or both the parties
subscriptions:
standing orders requesting notification in the event of changes to specified entities

Figure 2: The four core and original UDDI data structures shown in Figure 1 extended to show the two additional structures that were introduced in the later versions of the UDDI specification.

or individuals requesting automatic notification in the event of a change to specific entities within the UDDI registry. This data structure was introduced with UDDI Version 3, and is referred to as ‘operationalInfo’. It’s currently used to provide a change-order subscription mechanism, and is also sometimes referred to as ‘subscription’.

Figure 2 extends the data structure hierarchy shown in Figure 1 to show how these two additional structures come into play.

The UDDI specification *per se* describes the XML schema for this information model, SOAP messages to transport the XML-based information, and a set of APIs to manipulate and manage the UDDI data. There are two key APIs:

- The UDDI inquiry API, which can be used to search or browse through the information contained in a UDDI registry.
- The UDDI publication API, which enables programmers to create or delete the information structures within a UDDI registry.

Invoking a UDDI API results in one or more SOAP messages to be generated. There are about 40 inquiry- and publishing-functions-related SOAP messages that can be generated against a UDDI-compliant registry by the UDDI inquiry and publishing APIs.

UDDI, however, is not just a sterile specification. Ariba, IBM, and Microsoft made sure that there would also be publicly accessible sets of implementations of the UDDI specification. They also ensured that other companies would actively buy into the UDDI initiative. They were successful in that endeavour too. In September 2000, when UDDI was initially unveiled, it was endorsed by about 30 blue-chip companies, including Sun, Compaq, Dell, American Express, Merill Lynch, and Nortel Networks. Today, this number is in excess of 200. UDDI, in the form of UBR, is thus also a standards-based, interoperable service available, for free, on the Web – with a user-friendly, graphical interface as well as a programmatic interface that conforms to the UDDI APIs. The UBR was originally run by the three companies that collaborated to develop the initial UDDI specification, which was published in September 2000 as Version 1.0. Today, the URB is being run by IBM, Microsoft, NTT Communications, and SAP – with Ariba no longer involved. HP, which was supposed to take Ariba's place when Ariba opted out, is also not a UBR player right now – partly because of the distraction of its merger with Compaq and partly to prevent the UBR being seen as a 'wholly-US' operation.

The two pivotal standards exploited by UDDI are XML and SOAP. Note that WSDL is not one of the prerequisite standards used by UDDI. In reality there is no formal relationship between UDDI and WSDL. They do, however, complement each other. Given that WSDL can be used to describe the interface of a Web service, there is obviously a role that WSDL can play with regard to UDDI. The 'tModel' for a Web service could thus point to a WSDL description, as can the 'bindingTemplate'.

UDDI DATA STRUCTURES

The UDDI information model is composed of instances of persistent data structures. A persistent data structure is one that automatically preserves its old versions. With a persistent data structure, you can make changes to the structure without destroying the old version, so that all versions of the structure persist. This enables previous versions of the structure to be queried, as well as the latest version. The fact that UDDI uses persistent data structures means that you could have access to different versions of a service rather than just to the most recent.

UDDI's persistent data structures are referred to as 'entities'. An 'entity' in XML is a special unit of storage. They are the basic building blocks of an XML document. Although UDDI doesn't make explicit reference to this connection, it is implied and will be appreciated by those familiar with XML. The UDDI information model is said to be composed of instances of six core entity types (as shown in Figure 2):

- businessEntity

- businessService entity
- bindingTemplate entity
- tModel entity
- publisherAssertion entity
- subscription entity.

All these entities are represented in XML, with each structure corresponding to predefined XML elements and attributes. They are stored, in a persistent manner, in one or more UDDI nodes. Each entity acquires the type of its outermost XML element.

There is a very definite and immutable hierarchy between these six UDDI data structures as depicted in Figure 2 – although if you wanted to be ultra-pedantic you might claim that this hierarchy related to just the four original UDDI Version 1 structures, with 'publisherAssertion' and 'subscription' being more in the nature of 'flags' or references that apply to the other structures. Each business or organization described in a UDDI registry exists as a separate instance of a businessEntity data structure. Similarly, each service offered by a business or organization is maintained as a separate instance of data.

The fixed hierarchy between the UDDI data structures also results in an automatic, corresponding containment relationship between these structures. The BusinessEntity structure therefore contains one or more unique businessService structures, while each businessService structure, in turn, contains specific instances of bindingTemplate data. BindingTemplate structures contain information that includes pointers (or references) to specific instances of tModel structures.

UDDI KEYS

Each entity in a UDDI registry, be it a business, a service, a service binding information, or a 'tModel', is assigned a specific UDDI identifier or UDDI key. An example of a UDDI key would be:

```
uddi:EØ5BE6AØ-CØ43-11D7-9B41-ØØØ629DCØA53
```

The UDDI key uniquely identifies that entity, for the duration of its existence, within the UDDI registry. The persistence of UDDI data is related to the use of these unique keys. In the case of the UBR, all the nodes that make up the UBR will use and reflect the same UDDI key for a specific instance of information, given that all the nodes that make up the UBR work in concert when it comes to the information maintained by the UBR.

A UDDI registry assigns one of these unique identification keys when a new registry entry is first published (ie saved). Once assigned, a UDDI key can be used at any later time to access that specific instance of data on-demand. Before UDDI Version 3, all UDDI keys had to be generated by the ‘publishing’ UDDI node to ensure the uniqueness of each key. UDDI nodes generated these unique keys in the form of Universally Unique Identifiers (UUIDs). A UUID, also called a Globally Unique Identifier (GUID), is a scheme that permits resources to be uniquely named over time and space.

UUIDs were originally created for use with remote procedure call (RPC) mechanisms in the context of the Network Computing System (NCS) initiative of the 1980s. This RPC mechanism was later adopted by the Open Group’s Distributed Computing Environment (DCE), and further formalized by ISO within the framework of the once rampant Open Systems Interconnection (OSI) model as ISO standard ISO/IEC 11578:1996.

A UUID is 128-bits long. It can thus be represented by 32 hexadecimal digits, given that each hex digit corresponds to 4 bits. A UUID is guaranteed to be different from all other UUIDs generated until 3400 AD, or, in the worst case, still likely to be extremely different from any other even if the UUID generation algorithm is not optimally implemented.

The algorithm used to generate UUIDs relies on a combination of hardware addresses (derived from IEEE 802 MAC addresses uniquely assigned to each and every LAN adapter), time stamps, and random seeds. It is the use of time stamps that results in the caveat that the uniqueness of UUIDs is guaranteed only up until 3400 AD! The 32-hex-digit UUIDs, the only type of keys available with UDDI Versions 1 and 2, are not well suited for use by humans. They are arbitrary, unwieldy, and impossible to remember. With UDDI Version 3, it's possible to have keys that are considerably better suited to consumption by humans. The creator (ie publisher) of a UDDI entity can even request a particular key or a type of key. These so-called 'publisher assigned keys' are one of the major new features of UDDI Version 3.

With Version 3, it's now possible to have so-called 'domainKeys' as well as the existing 'uuidKeys'. Domain keys are based on the now commonplace Internet domain names. Since domain names have to be unique, keys based on domain names can also maintain UDDI's individualism. A domain-name-based UDDI key would look as follows:

```
uudi:acmecompany.com:businessRegistration
```

Thus, a domain named-based UDDI key for my businessEntity might be:

```
uddi:inet-guru:AnuraGurugeBusinessName
```

Domain-name-based UDDI keys are obviously much more flexible and meaningful than uuidKeys. However, the UDDI specification leaves it as an implementation option as to whether a given UDDI registry has to support domainKeys.

As well as these two different key types, it's now also possible to have what are referred to as 'derived keys'. A derived key has either a uuidKey or a domainKey as its base (ie its start), but it also has an alphanumeric ASCII character string, of arbitrary length, appended to it. This, especially where a uuidKey is the base, provides further flexibility when it comes to UDDI key assignment.

The growing trend towards private or semi-private enterprise UDDI registries augmenting the UBR complicates the management of UDDI keys – especially if domainKeys are in use. The issue has to do with UDDI publishers who wish to copy the entire contents of a UDDI entity entry from one UDDI registry (eg private) to another while preserving the key assigned to the

UDDI node API sets	**UDDI client API sets**
1 UDDI inquiry	1 UDDI subscription listener
2 UDDI publication	2 UDDI value set
3 UDDI security	
4 UDDI custody transfer	
5 UDDI subscription	
6 UDDI replication	

Figure 3: API sets specified in Version 3

original entity. UDDI Version 3 includes a capability known as ‘entity promotion’ to cater for such scenarios. With entity promotion, a UDDI publisher can propose the existing key for an entity as the preferred key for that entry. It is, however, left to the implementation and the exact policies of a given registry as to whether the new registry has to acquiesce to this request.

To facilitate publisher-assigned keys – and entity promotion scenarios in particular – Version 3 introduces the notion of root and affiliate UDDI registries. The use of a root UDDI registry by multiple affiliate registries ensures that the affiliate registries can readily share data amongst themselves, relative to that root, and still make sure that they maintain unique UDDI keys – since this would be arbitrated by the root. It’s suggested that the UBR should be considered, whenever possible, as the root registry in such distributed registry scenarios. In this case, the UBR, through the use of either uuidKeys or domainKeys, will ensure the uniqueness of the necessary UDDI keys.

UDDI APIS

As mentioned earlier, the UDDI inquiry and publication APIs are key to any UDDI implementation. They control the core functions related to publishing and locating entities in a UDDI registry. Although so far presented as single APIs for simplicity, the UDDI specification treats all APIs as consisting of sets. It then further splits these sets into node versus client API sets. The API sets specified in Version 3 are thus as shown in Figure 3.

The functionality made possible by these UDDI API sets can be characterized as follows:

- *UDDI inquiry*. This API set is what a programmer would use to locate and obtain details about specific entries stored in a UDDI registry. It supports three distinct patterns of inquiry to address all relevant inquiry modes for this type of data. The three types of inquiry pattern supported are as follows:
  - *Browse pattern inquiry*. As the name suggests, this enables you to conduct an inquiry starting with a broad categorization and then refining the inquiry to more specific criteria as each set of search results is displayed.
  - *Drill-down pattern inquiry*. This involves using a UDDI key, obtained via a browse pattern inquiry or by other means (eg using a domainKey) to access a specific entity.
  - *Invocation pattern inquiry*. This is typically used in the context of bindingTemplate entries for Web services. As suggested by the name, it's used when the application invoking this search is happy to have the Web service located by the search to be dynamically invoked at the end of the search. So this becomes a search-and-activate operation.
- *UDDI publication*. This API set is used to publish, update, and delete information contained in a UDDI registry. It's left to the implementation policies of a particular registry to decide which node of a particular UDDI registry is used by a publisher to publish a given set of data. UDDI itself provides an automated mechanism to check and reconcile information that may get published on different nodes of the same registry at different times. This again is an implementation issue.
- *UDDI security*. This API set is currently used to obtain and manage authentication information.
- *Custody (and ownership) transfer*. The publisher who initially

creates an entry has ownership of that entity. A custodial UDDI node (typically the one at which the entry was published) has to maintain a rigid relationship between an entity and its owner to ensure the integrity of the information in a UDDI registry. In the case of a multi-node registry (eg UBR), every node must guarantee the integrity of an entity's ownership and custody. A node therefore can't permit changes to an entity unless it's the custodial node for that entity. The custody (and ownership) transfer API set enables one node to transfer custody of certain entities to another node, or to transfer ownership of an entity from the current owner to a new one in an authorized and cooperative manner.

- *Subscription*. This is the API set that facilitates the 'subscription' scheme introduced with UDDI Version 3. This API set enables clients, known as 'subscribers', to register their interest in being notified if and when changes are made to specific entries in a registry.
- *Replication*. This is used to replicate entries within a UDDI registry.
- *Value set*. This API set permits authorized third parties to validate certain information being published in a UDDI registry, based on registered 'value sets'.

UDDI NODES AND UDDI REGISTRIES

A software system that supports at least one of the UDDI node API sets described above is considered to be a UDDI node. However, as well as supporting at least one node API set, there are three other criteria to which a UDDI node has to conform:

- A UDDI node must interact with UDDI data via the appropriate UDDI API sets.
- A UDDI node can only be a member of exactly one UDDI registry.
- A UDDI node, at least conceptually, has full access to and can manipulate data structures of a complete logical copy of

the total data managed by the registry of which it is a part. All interactions made via that node's query or publish API sets must always apply to this data and no other – that is, the UDDI APIs should be used to access and manipulate only the data associated with the UDDI registry.

A UDDI registry is made up of one or more UDDI nodes. When a registry comprises multiple nodes, all the nodes have to work in a collaborative manner to ensure that they all have access to the same logical set of data. The nodes making up a registry are collectively responsible for managing the UDDI data associated with that registry. The nodes will typically achieve this goal by using UDDI replication between the nodes using the UDDI replication API set.

As previously stressed, the UDDI specification doesn't specify any implementational criteria for realizing a UDDI node or a UDDI registry. The UBR was therefore implemented, from day one, as a multi-node registry. However, most private (if not semi-private) registries are likely, at least to begin with, to consist of just one node. How a given registry implements policy decisions (eg authentication, integrity, support of digital certificates, and so on) is also left as an implementation option. However, the necessary policy decisions have to be implemented in a consistent manner at each and every applicable point within the registry. A registry, however, has the option of delegating certain policy decisions to individual nodes – which is true in the case of the UBR.

SUMMARY

Despite its associations with Web services, UDDI is not limited to Web services or even e-business. It can and will handle any and all services offered by any enterprise. The UBR, the global implementation of UDDI, is real and active, and gaining recognition. It would be remiss not to register your enterprise in the UBR today, given that it's free and is meant to provide consistent, worldwide visibility as to what an enterprise does. Don't ignore UDDI as irrelevant to your enterprise. UDDI, and, in particular, the UBR, is intended to touch all enterprises. At a minimum, check out the UBR and stake your claim on it before it's too late.

Anura Gurugé
Strategic Consultant (USA)

NetView interface to Open Edition environment

In their quest for an integrated IT environment, many organizations are attempting to add Unix functions to their legacy environment. This can be achieved by means of a simple NetView interface, whose main characteristic is to supply an interface to Unix from an OS/390 environment, and which executes as follows:

- It browses the Unix System Services (OMVS) files.
- It reads information on the mounted file systems and on the Open Edition status.
- It executes the Unix mount function to mount a file system.
- It executes the Unix dismount function to dismount a file system.

The interface uses the NetView Browse command to display the Unix files. This can be restricted using the CMDMDL statement, by assigning a scope class number to it. In order to restrict a command to a specific scope class, code a CMDCLASS definition statement following the Browse command's CMDMDL statement. The CMDMDL statements are located in the NetView DSICMD library and contain the following CMDCLASS definition statement:

```
CNME5ØØ1   CMDMDL   MOD=DSICCP,ECHO=N,TYPE=B
       CMDSYN   BROWSE
       CMDSYN   BR
       CMDCLASS 1,2  (or any numbers from 1 to 2Ø4Ø)
```

where 1,2 specifies that the BROWSE command is restricted to scope classes 1 and 2. This means that only operators with scope classes 1 or 2 in their profiles can issue the command or command lists that invoke the Browse command.

Note that in order to execute the Unix Mount and Dismount commands, you need superuser authority.

This tool can be used to read information or files from an Open Edition environment, and also to execute commands in an Open Edition environment.

For example, you might want to browse some Communications Server (/etc/hosts; /etc/networks; /etc/services; /etc/inetd.conf) or Web Server (/etc/httpd.conf) parameter files, or information about the users defined to Unix (/u/USERXXX/.profile), or to consult configuration files or the Unix printer log (/etc/Printsrv/aopd.conf; /var/Printsrv/printers/prtlog.log).

This utility is essentially made up of a REXX EXEC and NetView panels that are executed under the control of the NetView. It was developed and tested under:

- OS/390 2.6 and OS/390 2.8
- Unix System Services
- REXX
- NetView.

REXX EXECS IN THE NETVIEW ENVIRONMENT

C-list OEXC000

```
/* REXX */
/* NetView OMVS procedure.
   C-List OEXCØØØ.
   Called by user from Netview terminal.
   Execute tool to browse the Open Edition files of the Unix System
Services
   Environment and to execute Unix commands (mount/dismount/display File
Sys.
   The functions are:
   - display panel to select OMVS file and/or execute Unix commands;
   - read the selected file;
   - browse the selected file with Netview function;
- execute command to display mounted file system;
- execute command to mount/dismount file system;
*/
trace ?o
/*____________________________________________________________
                          Variables
   CNMBR     = MVS partition dataset to write OMVS files
   OEVGØ1    = file name of the OMVS environment
   OEOP      = selected option
   OETXTB    = messages to user
  ____________________________________________________________*/
cnmbr = 'NETVIEW.SAØ1.OEMVS'
fillb = copies(' ',7Ø)
SIGNAL ON HALT
SIGNAL ON ERROR
 'UNIQUE PROMOTE'
$command = 'FI UY'
/*____________________________________________________________
                   Define global variables
  ____________________________________________________________*/
'GLOBALV GETT $oevgØ1,$oevgØ2,$oevgØ3,$oevgØ4,$oevgØ5'
'GLOBALV GETT oevgØ1,oevgØ2,oevgØ3,oevgØ4,oevgØ5,oeop'
/*____________________________________________________________
              Init procedure - display first panel
  ____________________________________________________________*/
Entry_point:
DO FOREVER
  command = 'ØØ'X
  oetxtb = copies(' ',7Ø)
  'VIEW OECXØØØ OEXPØØØ INPUT MSG'
  UPPER command
  SELECT
    when viewaid = PF1 then view 8 OEXPØØØH
    when viewaid = PF2 then exit
    when viewaid = PF6 then 'CMD HIGH ROLL'
    when viewaid = PF8 then call init_menu
    when viewaid = ENTER then
        SELECT
          when command = NEXT ] command = 'ØØ'X then call init_menu
          when command ¬= ' ' then
               DO
               'CMD HIGH' COMMAND
               END
          otherwise nop
        END
    otherwise nop
  END
END
init_menu:
trace ?o
```

```
/*___________________________________________________________
             Setting attributes panel variables
  ___________________________________________________________*/
$command = 'FI UY'
oetxtb = copies(' ',7Ø)
$oetxtb = 'FA IH CR HD UN'
$oeop = 'FI IH CG HR UN'
$oevgØ1 = 'FI IH CT HR UN'
$oevgØ2 = 'FI IH CT HR UN'
$oevgØ3 = 'FI IH CT HR UN'
$oevgØ4 = 'FI IH CT HR UN'
$oevgØ5 = 'FA IH CT HR UN'
/*___________________________________________________________
                  Display functions menu
  ___________________________________________________________*/
DO FOREVER
 command = 'ØØ'X
 oeop = copies(' ',4)
  'VIEW OECXØØ1 OEXPØØ1 INPUT MSG'
  UPPER command
  SELECT
    when oeop = Ø1 & oevgØ1 ¬= ' ' then call OEXCØ1Ø
cnmbr,fillb,oeop,oevgØ1
    when oeop = Ø1 & oevgØ1 = ' ' then do
                             oetxtb = 'Specify the name of the OMVS
file.'
                             'globalv putt oetxtb'
                                       End
    when oeop = Ø2 then call OEXCØ2Ø fillb,oeop
    when oeop = Ø3 & oevgØ2 ¬= ' ' & oevgØ3 ¬= ' ' then ,
                             call OEXCØ3Ø fillb,oeop,oevgØ2,oevgØ3
    when oeop = Ø3 & oevgØ2 = ' ' then do
                             oetxtb = 'Specify the HFS name for MOUNT.'
                             'globalv putt oetxtb'
                                       End
    when oeop = Ø3 & oevgØ3 = ' ' then do
                             oetxtb = 'Specify the MountPoint.'
                             'globalv putt oetxtb'
                                       End
    when oeop = Ø4 & oevgØ4 ¬= ' ' then call OEXCØ4Ø fillb,oeop,oevgØ4
    when oeop = Ø4 & oevgØ4 = ' ' then do
                         oetxtb = 'Specify the HFS name for DISMOUNT.'
                         'globalv putt oetxtb'
                                  End
    when oeop = Ø5 then do
                         oetxtb = 'Option not available.'
                         'globalv putt oetxtb'
                        End
    when viewaid = PF1 then do
                             view 8 OEXPØØ1H
```

```
       if oetxtb ¬= fillb then
                              do
                               oetxtb = fillb
                               'globalv putt oetxtb'
                              end
       end
    when viewaid = PF2 then exit
    when viewaid = PF7 then return
    when viewaid = PF6 then CMD HIGH ROLL
    when viewaid = PF11 then signal Entry_point
    when viewaid = ENTER then
        SELECT
          when command = BACK then return
          when command ¬= ' ' then
               DO
                oetxtb = 'Command/option not valid or in error.'
                'globalv putt oetxtb'
               END
          otherwise nop
        END
    otherwise nop
  END
END
/*___________________________________________________________
                        Errors routines
  ___________________________________________________________*/
ERROR:
retc = rc
if retc = 4 then do
          say time()
'******************************************************'
          say time() '***                                      ***'
          say time() '*** OEXCØØØ  - Netview OMVS.             ***'
          say time() '***      Procedure already active. Close.   ***'
          say time() '***                                      ***'
          say time()
'******************************************************'
          end
          else do
          say time()
'******************************************************'
          say time() '***                                      ***'
          say time() '*** OEXCØØØ  - Netview OMVS.             ***'
          say time() '***    Procedure in error. RC 'retc'         ***'
          say time() '***                                      ***'
          say time()
'******************************************************'
          end
 EXIT -1
HALT:
```

```
retc = rc
say time() '*****************************************************'
say time() '***                                               ***'
say time() '*** OEXCØØØ  - Netview OMVS.                      ***'
say time() '***            Procedure cancelled. RC 'retc'         ***'
say time() '***                                               ***'
say time() '***                                               ***'
say time() '*****************************************************'
EXIT
```

C-list OEXC010

```
/* REXX */
/* NetView OMVS procedure.
   C-List OEXCØ1Ø.
   Called by OEXCØØØ clist main.
   Execute tool to browse the Open Edition files of the Unix System
Services
   environment.
   The functions are:
   - read the selected file;
   - browse the selected file with Netview function;
*/
trace ?o
/*____________________________________________________________
                          Variables
   CNMBR     = MVS partition dataset to write OMVS files
   OEVGØ1    = file name of the OMVS environment
   OEOP      = selected option
   OETXTB    = messages to user
  ____________________________________________________________*/
Parse Arg cnmbr,fillb,oeop,oevgØ1
if oeop <= 9 then oeop = 'Ø'oeop
oetxtb = fillb
'globalv putt oetxtb'
/*____________________________________________________________
             Read file from Open Edition environment
  ____________________________________________________________*/
Call syscalls 'ON'
omvsfile = oevgØ1
Address syscall "readfile" omvsfile "omvsrec."
if retval = -1 then do
  say time() '*****************************************************'
  say time() '***                                               ***'
  say time() '*** OEXCØ1Ø  - Netview OMVS.                      ***'
  say time() '***            'omvsfile'   ***'
  say time() '***             Not read. Errno='errno' Errnojr='errnojr'
***'
  say time() '***                                               ***'
  say time() '*****************************************************'
  oetxtb = 'File not read. Errno' errno' - Errnojr' errnojr
  'globalv putt oetxtb'
  return
end
if omvsrec.Ø = Ø then do
  say time() '***************************************************'
  say time() '***                                             ***'
  say time() '*** OEXCØ1Ø  - Netview OMVS.                    ***'
  say time() '***          'omvsfile'   ***'
  say time() '***           Is empty. No records read.        ***'
  say time() '***                                             ***'
  say time() '***************************************************'
  oetxtb = 'File is empty. No records read.'
  'globalv putt oetxtb'
  return
end
trace ?o
mm = substr(time(l),1Ø,6)
filebr = cnmbr'(M'mm')'
/*___________________________________________________________
         Alloc MVS dataset member to write OMVS file
  ___________________________________________________________*/
say time() ' ...Creation 'filebr
"ALLOC DATASET('"filebr"') FILE(browsef) SHR"
/*___________________________________________________________
  If the records of the OMVS file are too long of 8Ø bytes
  then will be divided into records of 77 bytes long.
  These records are characterized with the characters ">> " in
  the first 3 bytes.
  ___________________________________________________________*/
y = Ø
Do i=1 to omvsrec.Ø
  ll = length(omvsrec.i)
  if ll > 8Ø then do
                 qrec = trunc(ll/77)
                 rest = ll//77
                 iniz = 1
                 do x=1 to qrec
                   y = y + 1
                   tab.y = '>> ']]substr(omvsrec.i,iniz,77)
                   iniz = iniz + 77
                 end
                 y = y + 1
                 tab.y = '>> ']]substr(omvsrec.i,iniz,rest)
                End
                else do
                      y = y + 1
                      tab.y = omvsrec.i
                End
End
```

```
/*___________________________________________________________
                  Write MVS dataset member
  ___________________________________________________________*/
ADDRESS MVS
 "NEWSTACK"
 "EXECIO * DISKW BROWSEF (STEM tab. FINIS"
 "DELSTACK"
ADDRESS NETVIEW
"FREE FILE(BROWSEF)"
/*___________________________________________________________
    Browse Netview function to display OMVS file selected
  ___________________________________________________________*/
br 'M'mm
Return
```

C-list OEXC020

```
/* REXX */
/* NetView OMVS procedure.
   C-List OEXCØ2Ø.
   Called by OEXCØØØ clist main.
   Execute tool to display the mounted Files Systems of the Unix System
   Services environment.
   The functions are:
   - command to OMVS for files systems;
   - display to user of the mounted files systems;
*/
trace ?o
/*___________________________________________________________
                            Variables
   OELIN1-3Ø = messages lines for display FS
   OEOP      = selected option
   OETXTB    = messages to user
  ___________________________________________________________*/
Parse Arg fillb,oeop
if oeop <= 9 then oeop = 'Ø'oeop
/*___________________________________________________________
                        Setting variables
  ___________________________________________________________*/
oetxtb = fillb
oehdr1 = fillb
do i=1 to 3Ø
  interpret oelin!!i "= fillb"
end
'globalv putt oetxtb'
'GLOBALV PUTT oehdr1,oelin1,oelin2,oelin3,oelin4,oelin5'
'GLOBALV PUTT oelin6,oelin7,oelin8,oelin9,oelin1Ø'
'GLOBALV PUTT oelin11,oelin12,oelin13,oelin14,oelin15'
'GLOBALV PUTT oelin16,oelin17,oelin18,oelin19,oelin2Ø'
'GLOBALV PUTT oelin21,oelin22,oelin23,oelin24,oelin25'
```

```
'GLOBALV PUTT oelin26,oelin27,oelin28,oelin29,oelin3Ø'
/*__________________________________________________________
              Setting attributes panel variables
  __________________________________________________________*/
$command = 'FI UY'
$oetxtb = 'FA IH CR HD UN'
$oehdr1 = 'FA IH CY HR UN'
$oelin1 = 'FA IH CT HR UN'
$oelin2 = 'FA IH CG HR UN'
$oelin3 = 'FA IH CG HR UN'
$oelin4 = 'FA IH CT HR UN'
$oelin5 = 'FA IH CG HR UN'
$oelin6 = 'FA IH CG HR UN'
$oelin7 = 'FA IH CT HR UN'
$oelin8 = 'FA IH CG HR UN'
$oelin9 = 'FA IH CG HR UN'
$oelin1Ø = 'FA IH CT HR UN'
$oelin11 = 'FA IH CG HR UN'
$oelin12 = 'FA IH CG HR UN'
$oelin13 = 'FA IH CT HR UN'
$oelin14 = 'FA IH CG HR UN'
$oelin15 = 'FA IH CG HR UN'
$oelin16 = 'FA IH CT HR UN'
$oelin17 = 'FA IH CG HR UN'
$oelin18 = 'FA IH CG HR UN'
$oelin19 = 'FA IH CT HR UN'
$oelin2Ø = 'FA IH CG HR UN'
$oelin21 = 'FA IH CG HR UN'
$oelin22 = 'FA IH CT HR UN'
$oelin23 = 'FA IH CG HR UN'
$oelin24 = 'FA IH CG HR UN'
$oelin25 = 'FA IH CT HR UN'
$oelin26 = 'FA IH CG HR UN'
$oelin27 = 'FA IH CG HR UN'
$oelin28 = 'FA IH CT HR UN'
$oelin29 = 'FA IH CG HR UN'
$oelin3Ø = 'FA IH CG HR UN'
/*__________________________________________________________
                Setting global variables
  __________________________________________________________*/
'GLOBALV PUTT $oehdr1,$oelin1,$oelin2,$oelin3,$oelin4,$oelin5'
'GLOBALV PUTT $oelin6,$oelin7,$oelin8,$oelin9,$oelin1Ø'
'GLOBALV PUTT $oelin11,$oelin12,$oelin13,$oelin14,$oelin15'
'GLOBALV PUTT $oelin16,$oelin17,$oelin18,$oelin19,$oelin2Ø'
'GLOBALV PUTT $oelin21,$oelin22,$oelin23,$oelin24,$oelin25'
'GLOBALV PUTT $oelin26,$oelin27,$oelin28,$oelin29,$oelin3Ø'
'GLOBALV GETT $oehdr1,$oelin1,$oelin2,$oelin3,$oelin4,$oelin5'
'GLOBALV GETT $oelin6,$oelin7,$oelin8,$oelin9,$oelin1Ø'
'GLOBALV GETT $oelin11,$oelin12,$oelin13,$oelin14,$oelin15'
'GLOBALV GETT $oelin16,$oelin17,$oelin18,$oelin19,$oelin2Ø'
```

```
'GLOBALV GETT $oelin21,$oelin22,$oelin23,$oelin24,$oelin25'
'GLOBALV GETT $oelin26,$oelin27,$oelin28,$oelin29,$oelin3Ø'
'GLOBALV GETT oehdr1,oelin1,oelin2,oelin3,oelin4,oelin5'
'GLOBALV GETT oelin6,oelin7,oelin8,oelin9,oelin1Ø'
'GLOBALV GETT oelin11,oelin12,oelin13,oelin14,oelin15'
'GLOBALV GETT oelin16,oelin17,oelin18,oelin19,oelin2Ø'
'GLOBALV GETT oelin21,oelin22,oelin23,oelin24,oelin25'
'GLOBALV GETT oelin26,oelin27,oelin28,oelin29,oelin3Ø'
/*____________________________________________________________
           Routine to read the mounted Files Systems
  ____________________________________________________________*/
DSPL:
'TRAP DISPLAY MESSAGES BPXOØ44I'
'MVS D OMVS,F'
IF RC ¬= Ø THEN
 DO
  'TRAP NO MESSAGES'
  'FLUSHQ'
  SIGNAL ERROR2
 END
  'WAIT 4Ø SECONDS FOR MESSAGES'
  SELECT
    WHEN (EVENT()='M') THEN
    DO
      'MSGREAD'
      'GETMSIZE' numlin
      'TRAP NO MESSAGES'
      'FLUSHQ'
      SELECT
        WHEN (MSGID()='BPXOØ44I') THEN SIGNAL Verify_dspl
        OTHERWISE NOP
      END
    END
    WHEN EVENT() = 'G' THEN
     DO
      'TRAP NO MESSAGES'
      'FLUSHQ'
      SIGNAL CANCEL
     END
    WHEN (EVENT() = 'T') THEN
     DO
      'TRAP NO MESSAGES'
      'FLUSHQ'
      SIGNAL TIMEOUT
     END
    OTHERWISE NOP
  END
Return

/*____________________________________________________________
```

```
                  Selection messages lines
                  Setting global variables
  ___________________________________________________________*/
Verify_dspl:
z = 1
if numlin > 33 then do
             numlin = 33
             oetxtb = 'File System mounted > of 1Ø. Display only first
1Ø FS.'
             'globalv putt oetxtb'
             end
Do x=1 to numlin
  if x = 1 then iterate
  if x = 3 then do
                 z = 2
                 iterate
                end
  'GETMLINE MSG' x
  oelin.x = MSG
  y = x - z
  if x = 2 then do
                 w1 = word(oelin.x,1)
                 w2 = word(oelin.x,2)
                 w3 = word(oelin.x,3)
                 w4 = word(oelin.x,4)
                 c1 = 'OE Procname 'w1
                 c2 = 'Asid 'w2
                 c3 = 'Status 'w3
                 c4 = 'OE MemberParm 'w4
                 oelin.x = c1!!'   '!!c2!!'   '!!c3!!'   '!!c4
                 interpret oehdr!!y "= oelin.x"
                 interpret "'globalv putt ' oehdr!!y"
                end
           else do
                 y = y - 1
                 interpret oelin!!y "= oelin.x"
                 interpret "'globalv putt ' oelin!!y"
                end
End
/*___________________________________________________________
              Routine of display files systems
  ___________________________________________________________*/
Dspl_point:
DO FOREVER
  command = 'ØØ'X
  oetxtb = copies(' ',7Ø)
  'VIEW OECXØ2Ø OEXPØØ2 INPUT MSG'
  UPPER command
  SELECT
    when viewaid = PF1 then view 8 OEXPØØ2H
    when viewaid = PF2 then exit
    when viewaid = PF6 then 'CMD HIGH ROLL'
    when viewaid = PF7 then return
    when viewaid = PF8 & numlin > 15 then view 8 OEXPØØ3
    when viewaid = ENTER then
        SELECT
          when command = BACK then return
          when command = NEXT then nop
          when command ¬= ' ' then
               DO
               'CMD HIGH' COMMAND
               END
          otherwise nop
        END
    otherwise nop
  END
END
RETURN
/*____________________________________________________________
                Routines of errors management
  ____________________________________________________________*/
ERROR2:
WTO '***********************************************'
WTO '***                                         ***'
WTO '***  OEXCØ2Ø  -  OMVS Procedure.            ***'
WTO '***                                         ***'
WTO '***              NetView error.             ***'
WTO '***                                         ***'
WTO '***              Verify display function to ***'
WTO '***              File System.               ***'
WTO '***                                         ***'
WTO '***********************************************'
oetxtb = 'OMVS Procedure in error. Verify display FS function.'
'globalv putt oetxtb'
return
CANCEL:
WTO '**************************************************'
WTO '***                                            ***'
WTO '*** OEXCØ2Ø  -  C-LIST CANCELLED BY OPERATOR   ***'
WTO '***                                            ***'
WTO '**************************************************'
oetxtb = 'OMVS Procedure cancelled by operator.'
'globalv putt oetxtb'
return
TIMEOUT:
n = n + 1
IF n > 4 THEN SIGNAL KO
SIGNAL DSPL
KO:
WTO '**************************************************'
WTO '***                                            ***'
WTO '*** OEXCØ2Ø  -  OMVS Procedure.                ***'
WTO '***             SYSTEM PROBLEM.                ***'
WTO '***             TO ADVISE SYSTEM SUPPORT STAFF ***'
WTO '***                                            ***'
WTO '**************************************************'
oetxtb = 'System problem. To advise system support staff.'
'globalv putt oetxtb'
return
```

C-list OEXC030

```
/* REXX */
/* NetView OMVS procedure.
   C-List OEXCØ3Ø.
   Called by OEXCØØØ clist main.
   Execute tool to mount a File System of the Unix Sys Services
environment.
   The functions are:
   - alloc utility file to exec Unix Mount function;
   - call clist OEXCJSUB to submit Unix Mount function;
*/
trace ?o
Parse Arg fillb,oeop,oevgØ2,oevgØ3
if oeop <= 9 then oeop = 'Ø'oeop
func = 'MOUNT'
oetxtb = fillb
microsec = substr(time(l),1Ø,6)
fsubØ = TEMP.UTILITY.OMVS.T]]microsec
/*___________________________________________________________
    Delete dataset "TEMP.UTILITY.OMVS.Tmicrosec" Type=work
  ___________________________________________________________*/
ADDRESS NETVIEW
"ALLOC FI(FJS) DATASET('"fsubØ"') MOD"
"FREE FI(FJS) DATASET('"fsubØ"') DELETE"
/*___________________________________________________________
    Alloc dataset "TEMP.UTILITY.OMVS.Tmicrosec" Type=work
  ___________________________________________________________*/
"ALLOC DATASET('"fsubØ"') FILE(FJS) SPACE(1Ø,1Ø) DSORG(PS)" ,
"RECFM(FB) LRECL(8Ø) BLKSIZE(616Ø)" ,
"UNIT(WORKA) NEW CATALOG"
/*___________________________________________________________
  Creation Job-Utility-OMVS in "TEMP.UTILITY.OMVS.Tmicrosec"
  ___________________________________________________________*/
omvsu=USROMVS
jskj = omvsu]]'#'
js.1='//'jskj' JOB (OEØØØØ25),'
js.2='//*       NOTIFY='omvsu','
js.3='//        CLASS=S,'
js.4='//        MSGCLASS=X,'
js.5='//        MSGLEVEL=(1,1)'
js.6='//*'
js.7='//STEP1   EXEC PGM=IKJEFT1B'
js.8='//SYSTSPRT DD   SYSOUT=*'
js.9='//SYSTSIN  DD   *'
js.1Ø="  MOUNT FILESYSTEM('"oevgØ2"') +"
js.11="        MOUNTPOINT('"oevgØ3"') +"
js.12='        TYPE(HFS)  MODE(RDWR)'
js.13='/*'
js.14='/*'
js.Ø=14
do a=1 to js.Ø
  js.a=left(js.a,8Ø)
end
say time() ' Creation ...'fsubØ
ADDRESS MVS
  "NEWSTACK"
  "EXECIO * DISKW FJS (STEM JS. FINIS"
  "DELSTACK"
ADDRESS NETVIEW
  "FREE FILE(FJS)"
/*___________________________________________________________
            Call clist to submit job-Utility-OMVS
  ___________________________________________________________*/
CALL OEXCJSUB fsubØ jskj func
if result = KO then do
       oetxtb = 'Submit Mount File System in error.'
       'globalv putt oetxtb'
       return
       End
 else do
       jobnum = result
       oetxtb = 'Submit Mount FS executed.' ,
                'To verify choose option 2 or output 'jobnum
       'globalv putt oetxtb'
       return
       End
```

C-list OEXC040

```
/* REXX */
/* NetView OMVS procedure.
   C-List OEXCØ4Ø.
   Called by OEXCØØØ clist main.
   Execute tool to dismount a File System of the Unix Sys Services
   environment.
   The functions are:
   - alloc utility file to exec Unix DisMount function;
   - call clist OEXCJSUB to submit Unix DisMount function;       */
trace ?o
```

```
Parse Arg fillb,oeop,oevgØ4
if oeop <= 9 then oeop = 'Ø'oeop
func = 'DISMOUNT'
oetxtb = fillb
microsec = substr(time(l),1Ø,6)
fsubØ = TEMP.UTILITY.OMVS.T]]microsec
/*____________________________________________________________
    Delete dataset "TEMP.UTILITY.OMVS.Tmicrosec" Type=work
  ____________________________________________________________*/
ADDRESS NETVIEW
"ALLOC FI(FJS) DATASET('"fsubØ"') MOD"
"FREE FI(FJS) DATASET('"fsubØ"') DELETE"
/*____________________________________________________________
    Alloc dataset "TEMP.UTILITY.OMVS.Tmicrosec" Type=work
  ____________________________________________________________*/
"ALLOC DATASET('"fsubØ"') FILE(FJS) SPACE(1Ø,1Ø) DSORG(PS)" ,
"RECFM(FB) LRECL(8Ø) BLKSIZE(616Ø)" ,
"UNIT(WORKA) NEW CATALOG"
/*____________________________________________________________
  Creation job utility OMVS in "TEMP.UTILITY.OMVS.Tmicrosec"
  ____________________________________________________________*/
omvsu=USROMVS
jskj = omvsu]]'#'
js.1='//'jskj' JOB (OEØØØØ25),'
js.2='//*       NOTIFY='omvsu','
js.3='//        CLASS=S,'
js.4='//        MSGCLASS=X,'
js.5='//        MSGLEVEL=(1,1)'
js.6='//*'
js.7='//STEP1   EXEC PGM=IKJEFT1B'
js.8='//SYSTSPRT DD   SYSOUT=*'
js.9='//SYSTSIN  DD   *'
js.1Ø="  UNMOUNT FILESYSTEM('"oevgØ4"')"
js.11='/*'
js.12='/*'
js.Ø=12
do a=1 to js.Ø
  js.a=left(js.a,8Ø)
end
say time() ' Creation ...'fsubØ
ADDRESS MVS
  "NEWSTACK"
  "EXECIO * DISKW FJS (STEM JS. FINIS"
  "DELSTACK"
ADDRESS NETVIEW
  "FREE FILE(FJS)"
/*____________________________________________________________
         Call clist to submit job batch utility OMVS
  ____________________________________________________________*/
CALL OEXCJSUB fsubØ jskj func
```

```
if result = KO then do
       oetxtb = 'Submit DisMount file system in error.'
       'globalv putt oetxtb'
       return
       End
 else do
       jobnum = result
       oetxtb = 'Submit DisMount FS executed.' ,
                'To verify choose option 2 or output 'jobnum
       'globalv putt oetxtb'
       return
       End
```

C-list OEXCJSUB

```
/* REXX */
/* NetView OMVS procedure.
   C-List OEXCJSUB.
   Called by OEXCØ3Ø/OEXCØ4Ø clists.
   Execute submit job to mount/dismount a File System of the Unix
   System Services environment.                                 */
trace ?o
ARG fsubØ jskj func
n = Ø
ADDRESS NETVIEW
JESCODE:
'TRAP DISPLAY MESSAGES  CNM279I'
"SUBMIT '"fsubØ"'"
IF RC ¬= Ø THEN
 DO
  'TRAP NO MESSAGES'
  'FLUSHQ'
  SIGNAL ERROR2
 END
  'WAIT 4Ø SECONDS FOR MESSAGES'
  SELECT
    WHEN (EVENT()='M') THEN
    DO
      'MSGREAD'
      'TRAP NO MESSAGES'
      'FLUSHQ'
      SELECT
        WHEN (MSGID()='CNM279I') THEN SIGNAL Submit_job_ok
        OTHERWISE NOP
      END
    END
    WHEN EVENT() = 'G' THEN
     DO
      'TRAP NO MESSAGES'
      'FLUSHQ'
      SIGNAL CANCEL
     END
    WHEN (EVENT() = 'T') THEN
     DO
      'TRAP NO MESSAGES'
      'FLUSHQ'
      SIGNAL TIMEOUT
     END
    OTHERWISE NOP
  END
Submit_job_ok:
'GETMLINE MSG 1'
jobnum = substr(msg,18,8)
say time() ' Submit 'func' Job-Utility-OMVS ...'
say time() ' Jobname .....' jskj
say time() ' Jobnumber ...' jobnum
/*____________________________________________________________
    Delete dataset "TEMP.UTILITY.OMVS.Tmicrosec" Type=work
  ____________________________________________________________*/
ADDRESS NETVIEW
"ALLOC FI(FJS) DATASET('"fsubØ"') MOD"
"FREE FI(FJS) DATASET('"fsubØ"') DELETE"
return jobnum
/*____________________________________________________________
                Routines of errors management
  ____________________________________________________________*/
ERROR2:
WTO '*************************************************'
WTO '***                                           ***'
WTO '***  OEXCJSUB  - NETVIEW ERROR.               ***'
WTO '***                                           ***'
WTO '***             STOP FUNCTION.                ***'
WTO '***             ADVISE SYSTEM SUPPORT STAFF   ***'
WTO '***                                           ***'
WTO '*************************************************'
return KO
CANCEL:
WTO '***************************************************'
WTO '***                                             ***'
WTO '*** OEXCJSUB -  C-LIST CANCELLED BY OPERATOR    ***'
WTO '***                                             ***'
WTO '***************************************************'
return KO
TIMEOUT:
n = n + 1
IF n > 4 THEN SIGNAL KO
SIGNAL JESCODE
KO:
WTO '*************************************************'
WTO '***                                           ***'
WTO '***  OEXCJSUB  - SYSTEM PROBLEM.             ***'
WTO '***                                          ***'
WTO '***            STOP FUNCTION.                ***'
WTO '***            ADVISE SYSTEM SUPPORT STAFF   ***'
WTO '***                                          ***'
WTO '************************************************'
return KO
```

NETVIEW PANELS

Panel OEXP000

```
/*
*** AT2
+OEXPØØØ%                Mainframe environment: Development
+
%                             System Support - Italy
$
+                 +OPERATOR ID =$&OPID    +APPLICATION =$&APPLID
+
+
[          ======  ==   ==  ======   ======   ======  ======   ======
[          ==      ==   ==  =\   NETVIEW OMVS PROCEDURE   [ =    ==
[          ==      ==   ==  ==   ==  ==   ==  ==  ==  ==   ==    ==
[          ======  ==   ==  =====    =====    ==  ==  ======     ==
[              ==  ==   ==  ==       ==       ==  ==  == ==      ==
[              ==  ==   ==  ==       ==       ==  ==  ==  ==     ==
[          ======  =======  ==       ==       ======  ==   ==    ==
\'Ø1$
$
$
$
$
$
$
[
%CMD==> &COMMAND
$            PF1=Help    PF2=End
$            PF6=Roll                        PF8/ENTER=To continue
```

Panel OEXP000H

```
/*  */
***
+OEXPØØØH%
+                                  [H E L P+
+              \ Procedure to interface Open Edition environment  +
+
+
+ This tool is a Netview interface in order to:
+
+ * consult the files of a system Unix System Services (OMVS)
+ * verify Open Edition status and which files systems they are mounted
+ * execute the MOUNT of file system
+ * execute the DISMOUNT of file system
+
+ Simply, it can be used in order to visualize (from an non-Unix
+ environment) whichever files of world OMVS.
+ A scope can be, as an example, to browse some parameters files of
+ the Communication Server(/etc/hosts; /etc/networks; /etc/services;
+ /etc/inetd.conf) or of the Web server(/etc/httpd.conf), or also
+ informations on the users defined to the Unix(/u/USERXXX/.profile).
+
+
[
+
$              PF2/PF3=Exit Help
$    PF6=Roll  PF7=Previous
```

Panel OEXP001

```
HELP=OEXPØØ1H
*** AT2
+OEXPØØ1%                  N E T V I E W  *@OMVS Procedure%*
+
+                } Procedure interface to Open Edition environment +
+
+      %1| Browse OMVS file:          +
+          &OEVGØ1                                                  %
+      %2| Display mounted File System +
+
+      %3| Mount File System:         +
@        HFS&OEVGØ2                                                 %
@        MP &OEVGØ3                                                 %
+      %4| Unmount File System:       +
+          &OEVGØ4                                                  %
+      %5| N/A              :         +
+          &OEVGØ5                                                  %
+
+
+
$ Selection option : &OEOP%
+
[ &OETXTB
%
%Action==> &COMMAND                                             %
$        PF1=Help   PF2=Exit
$        PF6=Roll   PF7=Previous              PF11=Entry Point
```

Panel OEXP001H

```
/*  */
***
+OEXPØØ1H%
+                                  [H E L P+
+              \ Procedure interface to Open Edition environment  +
+
+
+
+   From this panel it is possible to select OMVS file for its
+   consultation. After you have specified the name of the file,
+   you choose option 1.
+
+   With option "2" it is possible to verify the OMVS status and to
+   visualized the Files Systems mounted (maximum 1Ø File System).
+
+   With option "3" it is possible to execute MOUNT function of a File
    System
+
+   With option "4" it is possible to execute DISMOUNT function of a
    File System
+
+   The other options are not still available.
+
+
+
[
+
$              PF2/PF3=Exit Help
$    PF6=Roll  PF7=Previous
```

Panel OEXP002

```
HELP=OEXPØØ2H
*** AT
+OEXPØØ2%                 N E T V I E W  *@OMVS Procedure%*
+          } Display mounted File System of Open Edition environment +
+
+ &OEHDR1                                                           %
+ &OELIN1                                                           %
+ &OELIN2                                                           %
+ &OELIN3                                                           %
+ &OELIN4                                                           %
+ &OELIN5                                                           %
+ &OELIN6                                                           %
+ &OELIN7                                                           %
+ &OELIN8                                                           %
+ &OELIN9                                                           %
+ &OELIN1Ø                                                          %
+ &OELIN11                                                  %
+ &OELIN12                                                  %
+ &OELIN13                                                  %
+ &OELIN14                                                  %
+ &OELIN15                                                  %
+
[ &OETXTB                                                 %
%Action==> &COMMAND                                     %
$         PF1=Help   PF2=Exit
$         PF6=Roll   PF7=Previous   PF8=Next
```

Panel OEXP002H

```
/*  */
***
+OEXPØØ2H%
+                                [H E L P+
+       \ Display mounted File System of Open Edition environment +
+
+   This option allows to have informations on the mounted Files System.
+   The informations available are:
+
+   * procname of the OS/39Ø UNIX System Services cataloged procedure
+   * address space id of the Kernel
+   * currently status of OS/39Ø UNIX System Services
+   * File System type as defined by the FILESYSTYPE statement of OMVS
+   * device value to uniquely identify the device in Unix System
+     Services
+   * status of the File System
+   * file mode access
+   * jobname that quiesced the File System
+   * process ID that quiesced the File System
+   * dataset name of the Hierarchical File System (in MVS environment)
+   * pathname of the directory where the File System is mounted
+     truncated to 6Ø characters.
+
[
+
$              PF2/PF3=Exit Help
$    PF6=Roll  PF7=Previous
```

Panel OEXP003

```
HELP=OEXPØØ2H
*** AT
+OEXPØØ3%                 N E T V I E W  *@OMVS Procedure%*
+       } Display mounted File System of Open Edition environment +
+
+ &OEHDR1                                                   %
+ &OELIN16                                                          %
+ &OELIN17                                                          %
+ &OELIN18                                                          %
+ &OELIN19                                                          %
+ &OELIN2Ø                                                          %
+ &OELIN21                                                          %
+ &OELIN22                                                          %
+ &OELIN23                                                          %
+ &OELIN24                                                          %
+ &OELIN25                                                          %
+ &OELIN26                                                          %
+ &OELIN27                                                          %
+ &OELIN28                                                          %
+ &OELIN29                                                          %
+ &OELIN3Ø
[ &OETXTB                                                         %
%Action==> &COMMAND                                            %
$         PF1=Help   PF2=Exit
$         PF6=Roll   PF7=Previous
```

Espedito Morvillo and Rita Vacca
System Programmers (Italy)

TCP/SNA news

WRQ has announced the latest release of WRQ Reflection, the company's line of Windows- and Web-based host access and PC-UNIX integration solutions. The new products, WRQ Reflection 11.0 and WRQ Reflection for the Web 6.0, are available immediately, and combine new support for leading enterprise technologies with increased performance and management capabilities to enhance productivity, lower costs, and simplify IT administration.

Enhancements include:

- Templates for duplicating competitive products' user interfaces, for use when transitioning from competitive products to Reflection
- Secure access to HPe3000 applications using NS/VT and SSL/TLS with Windows- and Web-based emulation
- Windows Group Policy support
- Support for the consumption of Web services.

URL: http://www.wrq.com/aboutwrq/news/2003/100603pr.html

* * *

Zephyr Development has announced that its Passport Web to Host, a Web-based terminal emulator program, can now be deployed using the Apache Tomcat Web Server. Passport provides tn3270, tn5250, and VT100/VT220 host access to IBM zSeries (System/390), IBM iSeries (AS/400) and Unix host systems, as well as ftp file transfer.

Zephyr has also announced the release of Passport PC to Host 2004, the eleventh major version of the IP-based multi-host access suite. Special upgrade pricing for Passport PC to Host 2004 is available for Attachmate EXTRA!, IBM Personal Communications and NetManage RUMBA users.

URL: http://www.zephyrcorp.com/News/eNews/

* * *

Foundry Networks has released TrafficWorks IronWare 8.0 For ServerIron 400/800 Systems, which powers its ServerIron Layer 4-7 load balancing switches, enhancing server and network security, application traffic intelligence, and manageability of servers and load balancing switches. ServerIron switches provide Internet traffic and content switching features, including server load balancing, URL- and cookie-based switching, global server load balancing, firewall load balancing, and transparent cache switching. Among the new features is a new denial-of-service protection enhancement, to protect servers against TCP SYN and TCP ACK attacks at the rate of 1.5 million packets per second, while switching legitimate traffic.

URL: http://www.foundrynet.com/about/newsevents/releases/pr4_8_02.html

* * *