31 TCP/SNA

update

September 1998

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: t_eddolls@compuserve.com

North American office
Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75077-2150
USA
Telephone: 940 455 7050

Contributions
Articles published in *TCP/SNA Update* are paid for at the rate of £170 ($250) per 1000 words and £90 ($140) per 100 lines of code for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

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

Editor
Trevor Eddolls

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

Subscriptions and back-issues
A year's subscription to *TCP/SNA Update*, comprising four quarterly issues, costs £125.00 in the UK; $185.00 in the USA and Canada; £129.50 in Europe; £33.00 in Australasia and Japan; and £132.00 elsewhere. In all cases the price includes postage. Individual issues, starting with the March 1991 issue, are available separately to subscribers for £32.00 ($47.00) each including postage.

Printed in England.

Analysing buffer pool statistics

The DISPLAY BFRUSE operator command also indicates when buffer requests have been queued or have failed.Under these conditions the number of buffers requested is also displayed. You can use this information to help you determine whether the expansion limit has been reached. If it has, you may want to increase this value depending on your storage requirements.

Additionally, every time the buffer pool expands, VTAM issues messages to indicate which sessions are using more than 10% of the buffer pool.

The aim of the following procedure is to analyse the output resulting from 'D NET,BFRUSE' commands:

- A table named ISTMSGA is (re-)created.
- Reading the NCLLOG01 file from NetMaster Unique Console, the sequences of ISTxxxI messages, relating to the 'D NET,BFRUSE' commands, are filtered (&ISTMSGS variable).
- The file NCLLOG01 is read sequentially; if a date or progressive number parameter (yy/mm/ddnnnnn) is passed, the file is read from the specified record onwards.
- For each record, the presence of an ISTxxxI message beginning from byte 32 is checked. If the message is not found, the next record is read.
- If the message is not found in the entire file, the following is done:
  - From the first message(IST350I) date and time are extracted (&DATA6,&ORA8 variables).
  - From one of the messages groups (IST920I, IST921I, IST922I, IST923I, IST924I), nine numeric values are extracted with relation to a VTAM buffer named xxxx, eg:

```
IST920I xxxx  BUFF SIZE 319  EXP INCREMENT 48
IST921I   TIMES EXP 2293 EXP/CONT THRESH    60 /156
IST922I   CURR TOTAL    516  CURR AVAILABLE112
IST923I   MAX TOTAL 948  MAX USED  929
```

- The difference between CURR TOTAL and CURR AVAILABLE is calculated.
- The ten values are recorded in the ISTMSGA table with a key made of xxxx/date/time.

The created table is displayed by means of the READBUF procedure.

RUNBUFF PROGRAM

```
-*-------------------------------------------------------------------
-*
-* RUNBUFF PROGRAM
-*
-*-------------------------------------------------------------------
   &CONTROL NOENDMSG
-* &CONTROL TRACE TRACELOG

   &IKEY = &1
   &ISTMSG = &STR 350/+
                  920/921/922/923/924/920/921/922/923/924/+
                  920/921/922/923/924/920/921/922/923/924/+
                  920/921/922/923/924/920/921/922/923/924/+
                  920/921/922/923/924/920/921/922/923/924/+
                  920/921/922/923/924/+
                  449/790/449/790/+
                  595/981/314/

   &PMSG =  1
   &KYL  =  18

   &TABNAMXX = ISTMSGA

   &TABNAMXX FREE ID=&1 +                    -* KILL EXISTING TABLE
                   SCOPE=GLOBAL              -*
   &WRITE LOG=YES DATA=&0: CANCELLAZIONE TABELLA &1 ZFDBK=&ZFDBK.

   &TABPUTXX = 0                      -* NO. ENTERED AND WRITTEN IN TABLE
   &GOSUB .TABVERXX
   &DATA =
   &UDB = NCLLOG01
   &FILEID &UDB EXT-CHAR
   &CNT = 0
   &B = &SETBLNK 1
   &IF .&IKEY NE . &THEN &FILEKEY &IKEY
.LOOP
   &LOOPCTL 1000
   &IF .&IKEY NE . +
```

```
   &THEN &FILEGET KGT &RECORD(256)
   &ELSE &FILEGET SEQ &RECORD(256)
   &GOTO .GET&FILERC
.GETØ
   &CNT = &CNT + 1
   &MSG = &SUBSTR &ISTMSGS &PMSG 3
   &ISTMSG = &CONCAT IST &MSG I
   &X = &FNDSTR &ISTMSG &RECORD
   &IF &X EQ 32  &THEN &DO
     &IF &PMSG EQ 1  &THEN &DO
       &DAT6 = &SUBSTR &FILEKEY 1 8  -* DATE/TIME OF FIRST RECORD.
       &T1 = &SUBSTR &DAT6 1 2
       &T2 = &SUBSTR &DAT6 4 2
       &T3 = &SUBSTR &DAT6 7 2
       &DAT6 = &CONCAT &T1 &T2 &T3
       &ORA8 = &SUBSTR &RECORD 1 8
       &T1 = &SUBSTR &ORA8 1 2
       &T2 = &SUBSTR &ORA8 4 2
       &T3 = &SUBSTR &ORA8 7 2
       &ORA8 = &CONCAT &T1 &T2 &T3
    &DOEND
   &TEMP =&SUBSTR &RECORD 4Ø          -* NORMALIZATION MESSAGE.
   &PARSE ARGS DATA=&TEMP
   &TEMP = &STR &1 &2 &3 &4 &5 &6 &7 &8 &9
               &1Ø &11 &12 &13 &14 &15 &16 &17 &18 &19 +
               &2Ø &21 &22 &23 &24 &25 &26 &27 &28 &3Ø
   &TEMP =&TBLSTR &TEMP
   &SAVE&MSG = &TEMP                  -* SAVE MSG.

-*-------------------------------------------------------------------
-* 7 IST92ØI IOØØ      BUFF SIZE   319          EXP INCREMENT    48
-* 8 IST921I           TIMES EXP   2293         EXP/CONT THRESH  6Ø /156
-* 6 IST922I           CURR TOTAL  516          CURR AVAILABLE   112
-* 6 IST923I           MAX TOTAL   948          MAX USED         929
-*-------------------------------------------------------------------
-*   CURR INUSE  4Ø4  =  (CURR TOTAL  516  -  CURR AVAILABLE  112)
-*                                =  42Ø
-*                                  +48
-*                                  +48
-*   MAX TOTAL   948  =  (42Ø + 11 * 48)
-*
-*-------------------------------------------------------------------
     &IF &ISTMSG EQ IST924I &THEN +   -* SEPARATION PER BUFFER ?
      &DO
       &KEY = &CONCAT &XXXX           -*TESTATA.
       &DATA  &ASISTR  CURI  BSIZ  EXPI  TMEX  EXPC  EXPT +
                       CURT  CURA  MAXT  MAXU
       &GOSUB .TABPUTXX
       &PARSE ARGS DATA=&SAVE92Ø &SAVE921 &SAVE922 &SAVE923
       &IF &ZVARCNT NE 27 &THEN &ENDAFTER &WRITE DATA=PGM ERROR Ø1.
       &CONTROL ALIGNR
```

```
        &BSIZ = &STR &4
        &EXPI = &STR &7
        &TMEX = &STR &1Ø
        &EXPC = &STR &13
        &EXPT = &STR &15
        &CURT = &STR &18
        &CURA = &STR &21
        &MAXT = &STR &24
        &MAXU = &STR &27
        &CURI = &CURT - &CURA
        &KEY  = &CONCAT &XXXX &B &DAT6 &B &ORA8
        &DATA  &ASISTR  CURI  BSIZ  EXPI  TMEX  EXPC  EXPT  +
                        CURT  CURA  MAXT  MAXU
        &CONTROL NOALIGN
        &WRITE DATA=&SAVE&XXXX
        &WRITE DATA=&DATA
        &SAVE&XXXX = &DATA
        &GOSUB .TABPUTXX
       &DOEND
      &IF &ISTMSG EQ IST92ØI OR +      -* IST92ØI: XXXX (NOME BUFFER)
          &ISTMSG EQ IST449I OR +      -* IST449I: CASLIMIT /CSA24LIMIT
          &ISTMSG EQ IST595I OR +      -* IST595I: IRNLIMIT
          &ISTMSG EQ IST981I OR +      -* IST981I: VTAM
        &XXXX = &SUBSTR &RECORD 4Ø 4
      &IF &ISTMSG EQ IST35ØI OR +      -* INIZIO
          &ISTMSG EQ IST314I OR +      -* FINE
          &ISTMSG EQ IST924I OR +      -* SEPARATION PER BUFFER
        &XXXX = &STR NONE
      &IF &ISTMSG EQ IST314I  +
      &THEN &PMSG = 1
      &ELSE &PMSG = &PMSG + 4
       &DOEND
   &M5Ø = &CNT  /  5ØØØ
   &IF &M5Ø EQ Ø &THEN &WRITE DATA=&CNT RECORD LETTI
   &GOTO .LOOP
.GET4
   &WRITE LOG=YES DATA=&Ø: &CNT RECORDS ON FILE &UDB.
   &END
.GET8
   &WRITE LOG=YES DATA=&Ø: &UDB GET ERROR &VSAMFDBK, KEY=&FILEKEY.
   &END

-*-------------------------------------------------------------------------
-*   WRITE IN TABLE
-*-------------------------------------------------------------------------
.TABPUTXX
   &WRITE 'SCRIVO TAB'
   &VARVERB = PUT
   &VARTABLE PUT ID=&TABNAMXX SCOPE=GLOBAL KEY=KEY +
                 FIELDS=DATA VARS=DATA ADJUST=1
   &WRITE &VARTABLE
```

```
   &IF &ZFDBK NE Ø &THEN &GOTO .TABERRXX
   &TABPUTXX = &TABPUTXX + 1        -* NO. ENTERED AND WRITTEN IN TABLE
   &RETSUB

-*-------------------------------------------------------------------------
-*   VERIFIES DEFINITION TABLES
-*-------------------------------------------------------------------------
.TABVERXX
   &VARTABLE QUERY ID=&TABNAMXX SCOPE=GLOBAL FIELDS=TOTAL VARS=TABENTXX
   &IF &ZFDBK NE Ø &THEN &DO
     &VARVERB = ALLOC
     &VARTABLE ALLOC ID=&TABNAMXX SCOPE=GLOBAL KEYLEN=8
     &IF &ZFDBK NE Ø &THEN &GOTO .TABERRXX
    &DOEND
   &RETSUB

.TABERRXX
   &SYSMSG = &STR VARTABLE &VARVERB ID=&TABNAMXX ERROR +
                  - ZFDBK = &ZFDBK  &SYSMSG
   &WRITE DATA=&Ø: &SYSMSG
```

READBUF PROGRAM

```
-*-------------------------------------------------------------------------
-*
-* Readbuf PROGRAM
-*
-*-------------------------------------------------------------------------
  &CONTROL NOENDMSG

    &ID = ISTMSGA
    &WRITE &ID
    &SCOPE = GLOBAL
    &TOTCOUNT = Ø

    &VARTABLE QUERY ID=&ID SCOPE=&SCOPE FIELDS=(KEYLEN,TOTAL) +
                                           VARS=(KEYL,TOT)
    &IF &ZFDBK NE Ø &THEN &ENDAFTER +
      &WRITE DATA=&Ø: VARTABLE QUERY ID=&ID SCOPE=&SCOPE - FDBK=&ZFDBK.

    &KEY = &SETBLNK &KEYL
    &HDR1 = &STR K E Y
    &HDR1 = &SETLENG &KEYL
    &HDR1 = &STR ENT.;&HDR1;D A T A
    &WRITE DATA=*** TABELLA &ID: &TOT ENTRATE ***
    &WRITE DATA=&HDR1

    &CNT = Ø
    &VARTABLE GET ID=&ID SCOPE=&SCOPE KEY=KEY +
                         FIELDS=(KEY,DATA,COUNTER) +
```

```
                        VARS=(KEY,DATA,COUNTER) OPT=KGT
&DOWHILE &ZFDBK EQ Ø
 &LOOPCTL 1ØØØ
 &CNT = &CNT + 1
 &TOTCOUNT = &TOTCOUNT + &COUNTER
 &KEY = &SETLENG &KEYL
 &COUNTER = &INVSTR &COUNTER
 &COUNTER = &SETLENG 8
 &COUNTER = &INVSTR &COUNTER
 &WRITE DATA=&COUNTER;&KEY;&DATA
 &VARTABLE GET ID=&ID SCOPE=&SCOPE KEY=KEY +
                      FIELDS=(KEY,DATA,COUNTER) +
                        VARS=(KEY,DATA,COUNTER) OPT=KGT
&DOEND
&WRITE DATA=*** TABELLA &ID: &TOTCOUNT RIFERIMENTI ***
&END Ø
```

Fabio Lolli
Systems Programmer (Italy)

Using a TCP/ IP socket to send a message

Once, pure unadulterated SNA dominated the enterprise networking scene. When OS/390 appeared, TCP/IP was included in the system. So why not use TCP /IP?

The program below shows how to use a TCP/IP socket in the C language.

```
            *Description  :
            *
            * Internet server program echoes the message sent
            * by the client to the entry port.
            *
    ****************************************************************/

             /* Les -include */

             #include <stdio.h>
             #include <sys/types.h>
```

```
  #include <sys/socket.h>
  #include <sys/ioctl.h>
  #include <fcntl.h>
  #include <sys/termios.h>
  #include <netinet/in.h>
  #include <pwd.h>
  #include <netdb.h>

/* the constants */
#define LGUSER 2Ø
#define LGREP 256
typedef enum IFAUX,VRAII LOGIQUE
struct  sockaddr   in nom       /* Address of the soc@et
int longueur ; /*    length of the address

int creersock(int *port,int type)
/* Creation of a given socket
     type of socket created SOCK DGRAM     SOCK STREAM
     attachment of this socket to an address

     int desc ; /* desc of the created socket *l

     /* Creation of the socket */
     if @desc=socket(AF INET,type,O- == -1)
      perror("Creation @f the socket impossible");exit(2)

     bzero ( (char *) &nom, sizeof (nom)
     nom.sin- port=*port ;
     nom.sin- family=AF_INET
     nom.sin- addr.s -addr=INADDR-ANY
     /* give a name to the socket

  if ( (bind (desc, (struet sockaddr   &nom, sizeof (nom)
       perror("Error in naming the socket
       exit (3)

     longueur=sizeof (nom)

  if (getsockname(desc,(struct sockaddr *)&nom,&longueur-
perror("Error obtaining the name of the socket") ;exit(4);

     /* Chosen port
     *Port=ntohs(nom.sin-port);
     return desc;

  void main(int n, char *argv[])
  1
     int lg port, port2 sock, sock-rep
```

Tuning IP performance

Many network managers are finding that intranet performance can be a major issue at their site. At peak times, there is just too much traffic on the network and the servers can't send the traffic at the speeds required. In addition, TCP/IP has been around for many years and was never designed to handle current workloads.

This article looks at what's available for network managers to improve the performance of their networks. Solutions fall mainly into two categories – those that improve network performance and those that improve server performance.

IMPROVING NETWORK PERFORMANCE

There are three main tools available for improving network performance. These are:

- Bandwidth management
- Data cacheing
- Data compression.

Bandwidth management

Bandwidth management offers the following options:

- Prioritization schemes
- Router flow control
- Stand-alone bandwidth managers. These aim to designate controlled amounts of bandwidth to specific applications or end-users.

Prioritization

Traffic prioritization is supported by most routers. It allows different traffic types to be serviced based on their relative importance – this would typically be IP source or destination address and application

(although other criteria can be applied).

Weighted Fair Queuing (WFQ) is a common prioritization scheme. The router maintains multiple output queues on each output interface. The first queue contains system messages and is always emptied before any of the others. The router then services the remaining queues sequentially. How much data (ie how many bytes) are sent from each queue is site-determined. In this way, more important traffic is sent before less important, eg a database request before a browser.

Router flows

Adding Quality Of Service (QOS) to TCP/IP is very useful for sites running WAN links, where different traffic types contend for scarce bandwidth. This is implemented in different ways by different vendors.

Ipsilon Networks' version maps connectionless IP sessions onto connection-oriented ATM flows, which then could be prioritized using ATM mechanisms.

Cisco has tag switching, a means of classifying traffic by inserting a special field into each packet. A tag is added to each packet as it enters the edge of a network. These tags specify requirements such as minimum bandwidth or maximum latency. Once it's tagged, a packet is then switched through the network without being reclassified.

The IETF (Internet Engineering Task Force) is working on a standards-based tagging scheme called MultiProtocol Label Switching (MPLS) that extends the Cisco and Ipsilon approaches. It allows multiple flows to be joined together into a larger trunk, thus simplifying management. Tag switching and MPLS require routers and switches to keep track of individual flows within a trunk, which requires large amounts of switch memory. The IETF is also looking at using the Type Of Service (TOS) field already present in the IP header.

RSVP (Resource Reservation Protocol) permits applications to request a specific quality of service. RSVP sessions request two parameters – the end-to-end mean data rate and the minimum QOS.

Bandwidth management products running on stand-alone edge devices placed between the intranet and the router affect outbound and inbound TCP traffic. The devices change TCP window sizes (the number of bytes transmitted before an acknowledgment is required) and transmit TCP acknowledgments at a controlled rate. However, these devices can't control some forms of IP traffic, including inbound ICMP (ping) and UDP packets.

Bandwidth managers allow important applications to take priority over less important ones. However, they go beyond simple prioritization in several ways:

- They dynamically allocate queue depth; eg if a video-conferencing session's latency rises above some predefined threshold, the bandwidth manager can give it a bigger share of the pipe.
- They work on both outbound and inbound traffic, while router prioritization affects only outbound traffic.
- They may use more efficient algorithms to control congestion.
- They have autodiscovery. Products can analyse which applications generate the most traffic and suggest an appropriate bandwidth management policy. The policy can then be fine-tuned to meet an organization's requirements.

Data cacheing

Data cacheing is a technique in which a server stores frequently-accessed data so it doesn't have to be repeatedly reloaded across the network.

Whenever a user or application requests some data, the request is redirected to a cache server. The server checks to see whether it holds a copy of the data in its local store. If the data is found, the local store will send the requester a previously accessed copy. If the data is not found, the cache will go to the server or network where it can find the resource, put a copy in its local store, and deliver the data to the requester. Many cache managers also include update mechanisms to ensure the data they hold is timely.

Cacheing can speed performance of both networks and servers. In the network, proxy servers can reduce inbound intranet traffic by storing frequently-requested resources. And because data is stored much closer to the end-users who request it, response times are also improved.

Hierarchical cacheing is a technique that allows a group of caches to speed performance even more. There are two types of hierarchical cache – sibling and parent-child. In a sibling design, a cache that doesn't have a requested resource asks all other caches in the group whether they have the resource. A parent-child design is a vertical hierarchy – the child cache only asks its parent for a resource.

Proxy caches have two big problems:

- Pointing too many users at a cache can dramatically slow intranet performance. That may prompt some users to try circumventing the cache.
- All Web browsers and FTP client software on all end-stations must be reconfigured to point to the cache.

Data compression

Data flows can be optimized by compressing them. A look at the inefficient packet structures of many TCP/IP applications shows why compression is needed. For example, Telnet transmits only one character (one byte) per 64-byte packet. And many other applications use a large number of small packets, resulting in high overhead per unit of data transferred. Compression generally works best only on lower-speed WAN links (typically 2Mbit/s or less).

There are two methods of compression – compressing the TCP header or compressing the entire packet.

TCP header compression (RFC 1144), assumes, that as almost all the data in the 20-byte TCP header is repeated during a sustained transfer, it is unnecessary. By squeezing all but the most essential data, compression can reduce the TCP header from 20 bytes down to 5 (and even 3 bytes in some cases).

Compressing entire packets offers ratios of up to 4:1. Software- or

hardware-based compression works on a point-to-point basis, with two devices on either end of a link compressing and uncompressing data.

The major router vendors all offer compression as part of their software. Bear in mind, however, that software-based compression places a huge burden on a router's processor.

IMPROVING SERVER PERFORMANCE

Ways of improving server performance include:

- DNS load balancers
- IP load balancers
- Tuning the TCP/IP stack.

DNS load balancers

Load-balancing tools share traffic among a group of servers. There are two major types – those that balance DNS requests and those that balance all IP traffic. These come in intelligent and non-intelligent versions.

DNS load balancers present a group of servers to the outside world as one host name. Each server keeps its own IP address, and the load balancer parcels out requests to servers based on their availability.

Non-intelligent DNS load balancers send out requests in a round-robin fashion. But because they don't know the status of each server, they may send requests to a server that's already overloaded or even down for maintenance.

Intelligent DNS load balancers, in contrast, are aware of the load and availability of each server in a cluster. A small agent runs on each server and reports back to the load balancer.

IP load balancers

An example of a non-intelligent IP load balancer is Destination Address Rotary Translation, which is part of Cisco's IOS. It sends out

requests for a given destination IP address to a group of servers in a round-robin fashion. The servers all maintain their own IP addresses, but clients see only the one address. The problem with non-intelligent systems on a busy network is that they do not know about the loads or availability of servers within the group.

Intelligent IP load balancers do keep track of server status. These are generally dedicated hardware devices. Requests can be handed out in a variety of ways:

- At random.
- In round-robin fashion.
- In response to client domain name or IP address (certain clients get access to certain servers).
- By server criteria, eg CPU utilization or response time.

All criteria are user-definable.

Although a choice of server-based criteria sounds ideal, measuring server responsiveness isn't always easy.

IP load balancers don't use host name resolution. That's an improvement in terms of fault tolerance, since other servers won't cache a host name or IP address that may no longer be available. Even with non-intelligent request allocation methods like random or round-robin distribution, a server can still be removed from a group without affecting client requests. Because of their strong fault tolerance, IP or DNS load balancers nicely complement proxy caches.

Tuning the TCP/IP stack

Tuning the TCP/IP stack can speed up servers. Every TCP connection requires a three-way handshake that takes about half a second to complete, and a Web page may involve dozens of connections. Also, the longer the link, the greater the latency. Delays can stretch to three seconds or more over global satellite connections. Unfortunately, the loss of even one packet may require a much larger block of data to be re-transmitted. And TCP's built-in 'slow start' mechanism, which was intended to cut down on congestion, also degrades throughput.

Other performance-enhancing changes to TCP/IP stacks include increasing parameters like TCP window size or Maximum Transmission Unit (MTU, the largest allowable packet size).

However, large window sizes should not be used on slow or error-prone links, where re-transmissions caused by missed acknowledgments will slow performance. Applications that send many small packets (such as HTTP) don't benefit from larger MTUs.

CONCLUSION

There is a variety of tools and techniques available to help improve IP networks, and the good news is that many of them can be used at the same time. However, network managers should balance performance and reliability issues and test out any changes before implementing them on production networks.

Nick Nourse (UK)

Link between NCPs – part 3

This month we conclude the article that allows ten computers to be linked using a switched link.

```
*
*************************************************************************
*************************************************************************
*
GCAØ1B   GROUP LNCTL=CA,                                                *
               ISTATUS=ACTIVE
*
*    SEE NCP74Ø RESOURCE DEFINITION REFERENCE. PAGE 136
*    THE PHYSICAL ADDRESS IS THE ADDRESS OF THE CHANNEL BRANCH
*     3745 (8 POSITIONS ON THE MODEL 21Ø).
*    THE TABLE ON PAGE 136 SHOWS Ø1 WITH ADDRESS = Ø8
*
LCAØ5    LINE  ADDRESS=Ø8,          POSITION PHYSICAL ON THE 3745      *
               CA=TYPE6,            CHANNEL ADAPTOR TYPE               *
```



```
               CASDL=120,           INTERVAL BEFORE CHANNEL SLOWDOWN   *
               DELAY=Ø.2,           CHANNEL ATTN DELAY                 *
               DYNADMP=NONE,     NO EP SUBCHANNELS TO DUMP DATA OVER   *
               NCPCA=ACTIVE,                                           *
               TIMEOUT=12Ø       INTERVAL BEFORE CHANNEL DISCONTACT
*
PCAØ5    PU    PUTYPE=5,                                               *
               TGN=1             MUST BE ONE FOR PU TYPE 5
*              STATOPT='CA5 PU       '
*
         GENEND
*
         END
```

SWITCHED

```
*******************************************
*
* MSNF CVC PRIMARY   :  LILLE    (59)
*          SECONDARY :  TOURS    (37)
*          CCUA 38 APPEL 9..Ø9ØØ1111Ø1Ø4
*          CCUB 12 APPEL 9..Ø9ØØ3Ø1Ø1Ø4
*
*******************************************
X37CVC   VBUILD TYPE=SWNET,MAXGRP=4,MAXNO=4
*******************************************
* LINE ERØ CPU A PRIMARY, CPU B SECONDARY
*******************************************
P37CVC   PU    ADDR=C1,                                                *
               IDNUM=A5937,                                            *
               ISTATUS=ACTIVE,                                         *
               MAXDATA=1Ø24,                                           *
               MAXPATH=1,                                              *
               ANS=CONT,                                               *
               NETID=CP,                                               *
               TGN=1,                                                  *
               PUTYPE=4,                                               *
               SUBAREA=38
        PATH   DIALNO=137Ø8Ø743Ø9ØØ1111Ø1Ø4,                           *
               GID=1,PID=1,                                            *
               VERIFY=NONE,                                            *
               GRPNM=G12INNX
*******************************************
* LINE ER2 CPU A PRIMARY, CPU B SECONDARY
*******************************************
P37CVC2  PU    ADDR=C1,                                                *
               IDNUM=C5937,                                            *
```

```
                ISTATUS=INACTIVE,                                      *
                MAXDATA=1024,                                          *
                MAXPATH=1,                                             *
                ANS=CONT,                                              *
                NETID=CP,                                              *
                TGN=1,                                                 *
                PUTYPE=4,                                              *
                SUBAREA=38
          PATH  DIALNO=13708074309001110104,                           *
                GID=1,PID=1,                                           *
                VERIFY=NONE,                                           *
                GRPNM=G12INNX
*****************************************
* LINE ER1 CPU B PRIMARY, CPU A SECONDARY
*****************************************
P37CVC1   PU    ADDR=C1,                                               *
                IDNUM=B5937,                                           *
                ISTATUS=INACTIVE,                                      *
                MAXDATA=1024,                                          *
                MAXPATH=1,                                             *
                ANS=CONT,                                              *
                NETID=CP,                                              *
                TGN=1,                                                 *
                PUTYPE=4,                                              *
                SUBAREA=12
          PATH  DIALNO=13708074309003010104,                           *
                GID=1,PID=1,                                           *
                VERIFY=NONE,                                           *
                GRPNM=G12INNX
*****************************************
* LINE ER3 CPU B PRIMARY, CPU A SECONDARY
*****************************************
P37CVC3   PU    ADDR=C1,                                               *
                IDNUM=D5937,                                           *
                ISTATUS=INACTIVE,                                      *
                MAXDATA=1024,                                          *
                MAXPATH=1,                                             *
                ANS=CONT,                                              *
                NETID=CP,                                              *
                TGN=1,                                                 *
                PUTYPE=4,                                              *
                SUBAREA=12
          PATH  DIALNO=13708074309003010104,                           *
                GID=1,PID=1,                                           *
                VERIFY=NONE,                                           *
                GRPNM=G12INNX
```


A mailbox system for SMTP under MVS TCP/IP – 3

This issue we continue the code for the implementation of a mailbox system for SMTP, based on ISPF functions.

CLIST MAILSENS

This is used to construct and send mail.

```
/*                                                                  */
/*  MAILSENS
/*  Generel SMTP Mail Send Interface.
/*   Called from Panel INST81 and from Mail clists:
/*    ZCC
/*    ZMAIL
/*    ZREMAIL
/*    ZREPLY
/*
/*  PARAMETERS:
/*   ID      : Name of Mail, ie SUBJECT for SMTP mail;
/*             Default NONSENSE
/*   RC      : Receiver of Mail; can be TSO user-id,
/*             CC:mail username, any Internet username or a X400-
/*             address which if possible will be translated to
/*             Username and Domain.
/*   MEMOFILE: DD-name from which mail-data is taken in case this
/*             ddname is allocated and parameter DS is not indicated;
/*             Default: MEMOFILE.
/*   TIMEOFF : Time offset to GMT; Will normally be calculated from
/*             offset in SYS1.PARMLIB(CLOCKnn) whichever is active;
/*             fixed values if not not located from SYS1.PARMLIB:
/*             Wintertime +0100, Summertime +0200
/*   DS      : Optional input dataset (including evt member name);
/*             DS=blank and under ISPF-online: enter edit,
/*             DS=non-blank and under ISPF-online: take mail from DS
/*             DS=blank and under BATCH: mail from DD-name indicated
/*              in parameter MEMOFILE and this file is allocated, else
/*              take it from DD-name SYSUT1 if allocated.
/*             DS=non-blank and under BATCH: take mail from DS
/*   TRP     : Transporter; default SMTP
/*   DOMAIN  : Receivers DOMAIN; can be JES2-NJE Nodeid,
/*             CC:Mail domain name or
/*             any Internet domain name;
/*             defaults to sc.ccm.com (CC:MAIL-Sweden) normally,
/*             if receiver is TSO-user on current system to HOSTNAME,
/*             if receiver contains a NJE-network address to NJE-node
/*             (receiver%nje-node.hostname)
```

```
/*   RCDOMAIN: Default YES;
/*              if YES Receiver parameter will be scanned for AT-sign;
/*              whatever follows AT-sign will be used as DOMAIN, and in
/*              such case override contents of parameter DOMAIN.
/*              if NO domain name will be taken from DOMAIN parameter
/*              or defaulted if not provided.
/*   STO      : Sweden-CC:Mail Domain; MUST BE sc.ccm.com
/*   BJO      : USA-CC:Mail Domain; MUST BE us.ccm.com
/*   NICKOWN : Nickname for own Domain; default MULP
/*              Define NICKOWN=INST.TCPIP.SMTP.CONFIG:ALTNJEDOMAIN=
/*              INST.TCPIP.TCPIP.DATA.SMTP:HOSTNAME
/*   NICKBJO : Nickname for US Domain; default BJO-site
/*   NICKUS   : Nickname for US Domain; default USA
/*   REMAIL   : NO (default) build normal mail
/*              YES send prebuild mail in SMTP format
/*              REPLY build mail current mail as a reply
/*              FORWARD build mail as forward of current mail.
/*   REROUTE : NO (default) save copy of mail in mailbox
/*              YES rerouting of received mail, don't save copy in
/*              original receiver's mailbox.
/*   CCREQ    : NO (default) build normal mail
/*              YES calculate and return Domain, but don't send any mail
/*   HOSTNAME: Senders Host Name; MUST BE mulp.pmvsØ1.com
/*              as defined in INST.TCPIP.TCPIP.DATA.SMTP in
/*              HOSTNAME+DOMAINORIGIN for a Gateway node.
/*              For NJE-networknodes this will be reset to OWNNODE
/*   SMTPNODE: NJE Nodeid of JES2-node where SMTP is running
/*              as defined in INST.TCPIP.SMTP.CONFIG NJENODENAME;
/*              use the same in INST.TCPIP.SMTP.CONFIG NJEDOMAIN;
/*              this is the gateway node in case NJE-network exists.
/*   OWNNODE:  NJE Nodeid of own JES2-node as defined in SYS1.PARMLIB
/*              member JES2PARM in NAME for OWNNODE. On the gateway
/*              node this is identical to SMTPNODE, but on NJE-network
/*              nodes this is the actual node name.
/*   SMTPSTC : STC Name of SMTP: must be identical to this name (SMTP).
/*   MAILPNL : NO not called from mailpanel; YES called from mailpanel.
/*   MAXBYTES: Max SMTP mail size in bytes; default as defined in
/*              INST.TCPIP.SMTP.CONFIG parameter MAXMAILBYTES.
/*
/*   Subroutines/edit macroes:
/*    %EDITRECV
/*    %ISPFCPY
/*    %MAILDIRS
/*    %TIMEOFFS
/*    %USERNAME
/*    %WAITUNTJ
/*    %ZSMTPC
/*    %ZSMTPN
/*    %ZSMTPR
/*
/*   Utilities used:
```

```
 /*    ADSPNM
 /*    IEBGENER
 /*    ICEGENER
 /*    IKJCT44B (TSO/E CLIST built-in functions)
 /*    INSØ25M
 /*    SLEEP
 /*    TSOLINE1
 /*                                                                    */
PROC Ø ID(NONSENSE) RC() DS() DEBUG(nEBUG) +
TRP(SMTP) +
SMTPNODE(PMVSØ1) +
OWNNODE(PMVSØ1) +
MEMOFILE(MEMOFILE) +
SMTPSTC(SMTP) +
TIMEOFF(+Ø1ØØ) +
MAXBYTES(33554432) +
REMAIL(NO) +
REROUTE(NO) +
CCREQ(NO) +
RCDOMAIN(YES) +
MAILPNL(NO) +
HOSTNAME(mulp.pmvsØ1.com) +
STO(sc.ccm.com) +
BJO(us.ccm.com) +
NICKOWN(mulp) +
NICKUS(usa) +
NICKBJO(bjo) +
DOMAIN(sc.ccm.com)               /* HOSTNAME & DOMAIN lower case     */
 /*                                                                    */
CONTROL NOMSG NOFLUSH NOLIST NOCONLIST NOSYMLIST NOCAPS
ATTN DO                      /* attention routine */
 SET &FLUSH = FLUSH          /* NEXT STATEMENT MUST BE NULL LINE       */

END
ERROR DO                     /* error routine */
 SET &RET = &LASTCC
 RETURN
END
IF &SYSCAPS(&STR(&DEBUG)) NE DEBUG THEN DO
 ISPEXEC VGET (DEBUG)
END
IF &STR(&DEBUG) = &STR() THEN DO
 SET &DEBUG = NEBUG
END
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
 WRITE ======> Entering &SYSICMD <======
END
IF &FLUSH = FLUSH THEN DO   /* is attention active */
 CLOSFILE &STR(&MFILEID)
 CLOSFILE SYSUT1
```

```
 CLOSFILE MAILFILE
 DEL '&DSPREF..&JOBNM..&TSTAMP..SYSUT1.OUTLIST'
 DEL '&DSPREF..&JOBNM..&TSTAMP..MAILFILE.OUTLIST'
 IF &SYSENV = FORE THEN DO
  FREE FI(&STR(&MFILEID))
 END
 IF &SYSENV = FORE THEN DO
  ALLOC FI(SYSPRINT) DA(*) REUSE
  ALLOC FI(SYSIN) DA(*) REUSE
  ALLOC FI(SYSUT1) SPACE(Ø 5) TRACKS UNIT(&VIO) REUSE
  ALLOC FI(SYSUT2) SPACE(Ø 5) TRACKS UNIT(&VIO) REUSE
 END
 SET &SEND = YES
 ISPEXEC VPUT (SEND)
 SET &ZEDSMSG = &str(Function interrupted)
 ISPEXEC SETMSG MSG(ISRZØØ1)
 SET &ZISPFRC = 4
 ISPEXEC VPUT (ZISPFRC) SHARED
 EXIT CODE(&ZISPFRC)
END
ISPEXEC CONTROL ERRORS RETURN
ISPEXEC VGET (ZSCREEN)
ISPEXEC CONTROL DISPLAY LINE START(14)
IF &STR(&MAILPNL) = YES THEN DO
 ISPEXEC VGET (MEMOID,MEMORC)
 SET &ID = &STR(&SYSNSUB(1,&MEMOID))  /* take from ISPF variable to   */
 SET &RC = &STR(&SYSNSUB(1,&MEMORC))  /* allow ampersand and keep case*/
END
SET &MFILEID = &STR(&MEMOFILE)
SET &ATSIGN = &STR(@)       /* value of at-sign (use NLS-value) */
ADSPNM                      /* get current jobname in CLIST variable  */
SET &JOBNM = &ADSPNM
SET &MAILUS = &SYSUID
IF &STR(&SYSISPF) = ACTIVE THEN DO
 ISPEXEC VGET (USERNAME)
 IF &STR(&SYSNSUB(1,&USERNAME)) = &STR() THEN DO
  %USERNAME DEBUG(&STR(&DEBUG))
  IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
   WRITE ======> Reentering &SYSICMD <======
  END
 END
 ISPEXEC VGET (USERNAME)
 IF &STR(&SYSNSUB(1,&USERNAME)) NE &STR() THEN DO
  SET &BATCHUS = &STR((&SYSNSUB(1,&USERNAME)))
 END
END
IF &SYSENV = FORE THEN DO
 SET &DSPREF = &SYSUID
END
ELSE DO
 IF &STR(&SYSNSUB(1,&USERNAME)) NE &STR() THEN DO
```

```
  SET &BATCHUS = &STR((&SYSNSUB(1,&USERNAME), Batch &JOBNM))
 END
 ELSE DO
  SET &BATCHUS = &STR((Batch &JOBNM))
 END
 IF &MAILUS = &STR() THEN DO        /* If not TSO submitted job */
  SET &MAILUS = &JOBNM              /* use jobname as sender id */
 END
 SET &DSPREF = &STR(INSTPREF)      /* HARD CODED GENEREL PREFIX      */
 SET &GENPREF = &DSPREF
 IF &SYSPREF NE &STR() THEN DO
  IF &SYSPREF NE &DSPREF THEN DO
   IF &SYSUID NE &STR() THEN DO
    SET &DSPREF = &SYSUID
    SET &SYSOUTTRAP = 99999
    SET &RET = Ø
    LISTC ENT('&DSPREF')
    SET &LISTCRET = &RET
    SET &SYSOUTTRAP = Ø
    IF &LISTCRET NE Ø THEN DO
     SET &DSPREF = &GENPREF
    END
   END
  END
 END
 PROFILE PREFIX(&DSPREF)
END
SET &TOPREF = &DSPREF
ISPEXEC VPUT (TOPREF)           /* info to ZSMTPR                    */
SET &SYSOUTTRAP = 99999
SET &RET = Ø
LISTC ENT('&TOPREF')
SET &LISTCRET = &RET
SET &SYSOUTTRAP = Ø
  /* find out which machine we are executing on; on the test-mvs      */
  /* there is no SMTP stc, it is only a NJE node connected to SMTP    */
SMFID                             /* get smfid */
IF &STR(&SMFID) = TEST THEN DO    /* is it test MVS machine  */
 SET &OWNNODE = &STR(PMVSØ1T)     /* test MVS is a NJE node only */
END
IF &STR(&SMFID) NE TEST THEN DO   /* accept no alias on test machine */
 IF &LISTCRET NE Ø THEN DO
  WRITE ====> Error &TOPREF does not have a catalog alias (&SYSICMD).
  SET &ZISPFRC = 8
  ISPEXEC VPUT (ZISPFRC) SHARED
  EXIT CODE(&ZISPFRC)
 END
END
SET &EFFTRK = 5ØØØØ    /* set effective no of bytes per track */
SET &TC = 15           /* set no of tracks per cylinder */
SET &SECALC = ((&MAXBYTES+&EFFTRK-1)/&EFFTRK+&TC-1)/&TC /* sec alloc */
```

```
IF &SECALC < 1 THEN DO
 SET &SECALC = 1
END
IF &STR(&CCREQ) NE YES THEN DO
 SET &RET = Ø
 %TIMEOFFS DEBUG(&STR(&DEBUG))    /* calculate offset to GMT */
 SET &OFFRET = &RET               /* get offset */
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 IF &OFFRET NE &EVAL(2**24-1) THEN DO
  SET &Z = &STR(+)
  IF &OFFRET >= 1ØØØØ THEN DO
   SET &OFFRET = &OFFRET - 1ØØØØ
   SET &Z = &STR(-)
  END
  SET &TIMEOFF = &OFFRET
  SET &FOURZERO = &STR(ØØØØ)
  SET &LENOFF = &LENGTH(&TIMEOFF)
  SET &TIMEOFF = &SUBSTR(1:4-&LENOFF,&STR(&FOURZERO))&TIMEOFF
  SET &TIMEOFF = &STR(&Z&TIMEOFF)
 END
END
IF &STR(&REMAIL) = REPLY OR &STR(&REMAIL) = FORWARD THEN DO
 ISPEXEC VGET (SU)
 IF &STR(&SYSNSUB(1,&SU)) NE &STR() THEN DO
  SET &ID = &STR(&SYSNSUB(1,&SU))   /* keep case */
 END
END
IF &STR(&SYSNSUB(1,&RC)) = &STR(?) AND &SYSISPF = ACTIVE +
AND &SYSENV = FORE THEN DO
 SET &RET = Ø
 %MAILDIRS CALLER(&SYSICMD) DEBUG(&STR(&DEBUG))   /* SEARCH DIRECTORY */
 SET &DIRSRET = &RET
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 IF &DIRSRET = Ø THEN DO
  ISPEXEC VGET (DIRRC,DIRDM)
  /*                                                                */
  SET &STRING = &STR(&SYSNSUB(1,&DIRRC))
  SET &RSTRING = &STR()     /* CLEAR FOR CALL TO SUBROUTINE RETURN */
  SET &RET = Ø
  SYSCALL VARSTRNG RSTRING STRING(&STR('&SYSNSUB(1,&STRING)')) +
  DEBUG(&DEBUG)
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &RC = &STR(&SYSNSUB(1,&RSTRING))
  /*                                                                */
  SET &STRING = &STR(&SYSNSUB(1,&DIRDM))
  SET &RSTRING = &STR()     /* CLEAR FOR CALL TO SUBROUTINE RETURN */
```

```
  SET &RET = Ø
  SYSCALL VARSTRNG RSTRING STRING(&STR('&SYSNSUB(1,&STRING)')) +
  DEBUG(&DEBUG)
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &DOMAIN = &STR(&SYSNSUB(1,&RSTRING))
  /*                                                                */
  SET &DIRRC = &STR()
  SET &DIRDM = &STR()
  ISPEXEC VPUT (DIRRC,DIRDM)
 END
END
SET &TSTAMP = +
&STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
IF &STR(&SYSNSUB(1,&RC)) = &STR() OR +
&STR(&SYSNSUB(1,&RC)) = &STR(?) THEN DO
 ISPEXEC CONTROL DISPLAY LINE START(14)
 SLEEP 1
 SET &RC = &STR()
 WRITE ====> No Receiver indicated, enter Receiver or Receiver@domain.
 SET &RET = Ø
 SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
 /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
 SET &RET = &LASTCC
 SET &ZISPFRC = 8
 ISPEXEC VPUT (ZISPFRC) SHARED
 EXIT CODE(&ZISPFRC)
END
IF &STR(&SYSNSUB(1,&RC)) = &STR(Z) OR +
&STR(&SYSNSUB(1,&RC)) = &STR(DUMMY) THEN DO
 WRITE &STR(Receiver          ====> +
 &SYSNSUB(1,&RC) ignored, considered Dummy)
 SET &RET = Ø
 SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
 /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
 SET &RET = &LASTCC
 SET &ZISPFRC = Ø
 ISPEXEC VPUT (ZISPFRC) SHARED
 EXIT CODE(&ZISPFRC)
END
IF &STR(&CCREQ) NE YES THEN DO
 WRITE &STR(Mail System       ====> &TRP)
 WRITE &STR(Mail ID           ====> &SYSNSUB(1,&ID))
 WRITE &STR(Receiver          ====> &SYSLC(&SYSNSUB(1,&RC)))
END
ELSE DO
 WRITE &STR(CC-Receiver          ====> &SYSLC(&SYSNSUB(1,&RC)))
END
IF &STR(&CCREQ) NE YES THEN DO
 IF &STR(&DS) NE &STR() THEN DO
  WRITE &STR(MAIL from DATASET ====> &DS)
```

```
 END
 ELSE DO
  IF &SYSENV NE FORE THEN DO
   WRITE &STR(MAIL will be taken from DD-name SYSUT1)
  END
 END
END
SET &SRCHSPC = &STR(&ATSIGN) /* look for at sign */
SET &DM = &STR()
SET &LENRC = &LENGTH(&STR(&SYSNSUB(1,&RC)))
SET &STARTSPC = 1
SET &LOCSPC = +
&SYSINDEX(&STR(&SRCHSPC),&STR(&SYSNSUB(1,&RC)),&STARTSPC)
IF &LOCSPC > Ø THEN DO
 IF &STR(&SYSNSUB(1,&RCDOMAIN)) = YES THEN DO
  SET &DM = &SUBSTR(&LOCSPC+1:&LENRC,&STR(&SYSNSUB(1,&RC)))
  IF &STR(&SYSNSUB(1,&DM)) NE &STR() THEN DO
   SET &RC = &SUBSTR(1:&LOCSPC-1,&STR(&SYSNSUB(1,&RC)))
   SET &DOMAIN = &STR(&SYSNSUB(1,&DM))
   WRITE &STR(Domain reset to implied from Receiver: &SYSNSUB(1,&DM))
  END
 END
 ELSE DO
  WRITE ====> Warning: Receiver already contains AT-sign +
  without RCDOMAIN requested (&SYSICMD).
 END
END
SET &OWNNET = &STR(&OWNNODE..&SMTPNODE)
SET &OWNALT = &STR(&OWNNODE..&NICKOWN)
SET &NETDLM = &STR(%)
SET &LENRC = &LENGTH(&STR(&SYSNSUB(1,&RC)))
SET &SRCHSPC = &STR(&NETDLM) /* look for %-sign */
SET &STARTSPC = 1
SET &LOCSPC = +
&SYSINDEX(&STR(&SRCHSPC),&STR(&SYSNSUB(1,&RC)),&STARTSPC)
IF &LOCSPC > Ø THEN DO     /* nje-network address */
 SET &US = &SUBSTR(1:&LOCSPC-1,&STR(&SYSNSUB(1,&RC)))
 SET &NET = &STR()
 SET &NET = &SUBSTR(&LOCSPC+1:&LENRC,&STR(&SYSNSUB(1,&RC)))
 IF &STR(&SYSLC(&OWNNET)) = &STR(&SYSLC(&SYSNSUB(1,&NET))) THEN DO
  SET &RC = &STR(&SYSNSUB(1,&US))
  SET &DOMAIN = &STR(&SYSNSUB(1,&OWNNODE))
  WRITE &STR(Receiver and Domain reset to NJE-Network address)
 END
 IF &STR(&SYSLC(&OWNALT)) = &STR(&SYSLC(&SYSNSUB(1,&NET))) THEN DO
  SET &RC = &STR(&SYSNSUB(1,&US))
  SET &DOMAIN = &STR(&SYSNSUB(1,&OWNNODE))
  WRITE &STR(Receiver and Domain reset to NJE-Network address)
 END
 SET &GWYDLM = &STR(.)
 SET &LENNET = &LENGTH(&STR(&SYSNSUB(1,&NET)))
```

```
 SET &SRCHSPC = &STR(&GWYDLM) /* look for dot */
 SET &STARTSPC = 1
 SET &LOCSPC = +
 &SYSINDEX(&STR(&SRCHSPC),&STR(&SYSNSUB(1,&NET)),&STARTSPC)
 IF &LOCSPC > Ø THEN DO     /* nje-network address */
  SET &NODE = &SUBSTR(1:&LOCSPC-1,&STR(&SYSNSUB(1,&NET)))
  SET &GWY = &STR()
  SET &GWY = &SUBSTR(&LOCSPC+1:&LENNET,&STR(&SYSNSUB(1,&NET)))
  IF &STR(&SYSLC(&SMTPNODE)) = &STR(&SYSLC(&SYSNSUB(1,&GWY))) THEN DO
   SET &RC = &STR(&SYSNSUB(1,&US))
   SET &DOMAIN = &STR(&SYSNSUB(1,&NODE))
   WRITE &STR(Receiver and Domain reset to NJE-Network address)
  END
  IF &STR(&SYSLC(&NICKOWN)) = &STR(&SYSLC(&SYSNSUB(1,&GWY))) THEN DO
   SET &RC = &STR(&SYSNSUB(1,&US))
   SET &DOMAIN = &STR(&SYSNSUB(1,&NODE))
   WRITE &STR(Receiver and Domain reset to NJE-Network address)
  END
 END
END
IF &STR(&SYSLC(&SYSNSUB(1,&DOMAIN))) = &STR(&SYSLC(&SMTPNODE)) THEN DO
 SET &DOMAIN = &STR(&SYSLC(&HOSTNAME))
END
IF &STR(&SYSLC(&SYSNSUB(1,&DOMAIN))) = &STR(&SYSLC(&NICKOWN)) THEN DO
 SET &DOMAIN = &STR(&SYSLC(&HOSTNAME))
END
IF &STR(&SYSLC(&SMTPNODE)) NE &STR(&SYSLC(&OWNNODE)) THEN DO
 SET &HOSTNAME = &STR(&SYSLC(&OWNNODE))   /* set network node sender */
END
IF &STR(&SYSLC(&SYSNSUB(1,&DOMAIN))) = &STR() THEN DO
 SET &DOMAIN = &STR(&SYSLC(&STO))
END
IF &STR(&SYSLC(&SYSNSUB(1,&DOMAIN))) = &STR(&SYSLC(&NICKBJO)) THEN DO
 SET &DOMAIN = &STR(&SYSLC(&BJO))
END
IF &STR(&SYSLC(&SYSNSUB(1,&DOMAIN))) = &STR(&SYSLC(&NICKUS)) THEN DO
 SET &DOMAIN = &STR(&SYSLC(&BJO))
END
IF &STR(&CCREQ) NE YES THEN DO
 WRITE &STR(Senders Domain    ====> &SYSLC(&HOSTNAME))
 WRITE &STR(Receivers Domain  ====> &SYSLC(&SYSNSUB(1,&DOMAIN)))
END
ELSE DO
 WRITE &STR(CC-Receivers Domain  ====> &SYSLC(&SYSNSUB(1,&DOMAIN)))
END
IF &SYSENV NE FORE THEN DO
 WRITE &STR(MEMOFILE          ====> &MEMOFILE)
 WRITE &STR(REMAIL            ====> &REMAIL)
END
IF &STR(&REMAIL) = YES OR &STR(&REMAIL) = REPLY OR +
&STR(&REMAIL) = FORWARD THEN DO
```

```
 SET &TRP  = SMTP
END
SET &MAXCNT = 12Ø
SET &APIRET = Ø
SET &VIO = VIO                              /* VIO unit name */
SET &MAXSMTPLR = 251
SET &BLK8Ø = 2792Ø
SET &BLKSMTPLR = 27861
IF &LENGTH(&STR(&SYSNSUB(1,&RC))) > 3 THEN DO
 SET &RCPF4 = &SUBSTR(1:4,&STR(&SYSNSUB(1,&RC)))
END
IF &LENGTH(&STR(&SYSNSUB(1,&RC))) > 2 THEN DO
  /* support for compatibility reasons addresses in X4ØØ format     */
  /* and translate them to cc:mail format                           */
 SET &RCPF3 = &SUBSTR(1:3,&STR(&SYSNSUB(1,&RC)))
 SET &RCPF2 = &SUBSTR(1:2,&STR(&SYSNSUB(1,&RC)))
 IF &STR(&RCPF3) = &STR(FX=) OR &STR(&RCPF2) = &STR(G=) OR +
 &STR(&RCPF2) = &STR(S=) OR &STR(&RCPF2) = &STR(O=) OR +
 &STR(&RCPF2) = &STR(P=) OR &STR(&RCPF2) = &STR(A=) OR +
 &STR(&RCPF2) = &STR(C=) OR &STR(&RCPF3) = &STR(OU=) OR +
 &STR(&RCPF4) = &STR(OU1=) THEN DO
  SET &LENADDR = &LENGTH(&STR(&SYSNSUB(1,&RC)))
  IF &LENADDR > Ø THEN DO
   SET &NQ = Ø
   SET &QUAL = QUAL
   DO WHILE (&NQ < 16)
    SET &NQ = &NQ + 1
    SET &&QUAL&NQ = &STR()
   END
   SET &SRCHCHAR = &STR(;)
   SET &P = &LENADDR
   SET &STARTLOC = 1
   SET &QUAL = QUAL
   DO WHILE &P > Ø
    SET &LOC = +
    &SYSINDEX(&STR(&SRCHCHAR),&STR(&SYSNSUB(1,&RC)),&STARTLOC)
    IF &LOC = Ø THEN DO
     SET &QUALIFIER = +
     &SUBSTR(&STARTLOC:&LENGTH(&STR(&SYSNSUB(1,&RC))),+
     &STR(&SYSNSUB(1,&RC)))
     SET &LQ = &LENGTH(&STR(&QUALIFIER))
     DO WHILE &SUBSTR(1:1,&STR(&QUALIFIER)) = &STR() AND &LQ > 1
      SET &QUALIFIER = &SUBSTR(2:&LQ,&STR(&QUALIFIER))
     END
     SET &P = Ø
    END
    ELSE DO
     SET &PERIOD = &PERIOD + 1
     SET &QUALLEN = &LOC - &STARTLOC
     SET &ENDLOC = &LOC - &LENGTH(&STR(&SRCHCHAR))
     SET &QUALIFIER = &SUBSTR(&STARTLOC:&ENDLOC,&STR(&SYSNSUB(1,&RC)))
```

```
 SET &LQ = &LENGTH(&STR(&QUALIFIER))
 DO WHILE &SUBSTR(1:1,&STR(&QUALIFIER)) = &STR() AND &LQ > 1
  SET &QUALIFIER = &SUBSTR(2:&LQ,&STR(&QUALIFIER))
 END
 SET &PREVQUAL = &STR(&QUALIFIER)
 SET &STARTLOC = &LOC + &LENGTH(&STR(&SRCHCHAR))
 SET &P = &P - &QUALLEN - &LENGTH(&STR(&SRCHCHAR))
 SET &&QUAL&PERIOD = &STR(&QUALIFIER)
END
WRITE &STR(====> &QUALIFIER)
SET &B = &STR(&QUALIFIER)
SET &CCCOUNTRY = &STR(SE)
IF &STR(&B) NE &STR() THEN DO
 IF &LENGTH(&STR(&B)) > 2 THEN DO
  SET &BPF2 = &SUBSTR(1:2,&STR(&B))
  IF &STR(&BPF2) = &STR(S=) THEN DO
   SET &SURNAME = &SUBSTR(3:&LENGTH(&STR(&B)),&STR(&B))
   SET &LSN = &LENGTH(&STR(&SURNAME))
   IF &SUBSTR(&LSN:&LSN,&STR(&SURNAME)) = &STR(;) THEN DO
    SET &SURNAME = &SUBSTR(1:&LSN-1,&STR(&SURNAME))
   END
  END
  IF &STR(&BPF2) = &STR(G=) THEN DO
   SET &GIVENNAME = &SUBSTR(3:&LENGTH(&STR(&B)),&STR(&B))
   SET &GSN = &LENGTH(&STR(&GIVENNAME))
   IF &SUBSTR(&GSN:&GSN,&STR(&GIVENNAME)) = &STR(;) THEN DO
    SET &GIVENNAME = &SUBSTR(1:&GSN-1,&STR(&GIVENNAME))
   END
  END
 END
 IF &LENGTH(&STR(&B)) > 3 THEN DO
  SET &BPF3 = &SUBSTR(1:3,&STR(&B))
  IF &STR(&BPF3) = &STR(OU=) THEN DO
   SET &CCCOUNTRY = &SUBSTR(4:&LENGTH(&STR(&B)),&STR(&B))
   SET &CSN = &LENGTH(&STR(&CCCOUNTRY))
   IF &SUBSTR(&CSN:&CSN,&STR(&CCCOUNTRY)) = &STR(;) THEN DO
    SET &CCCOUNTRY = &SUBSTR(1:&CSN-1,&STR(&CCCOUNTRY))
   END
  END
 END
 IF &LENGTH(&STR(&B)) > 4 THEN DO
  SET &BPF4 = &SUBSTR(1:4,&STR(&B))
  IF &STR(&BPF4) = &STR(OU1=) THEN DO
   SET &CCCOUNTRY = &SUBSTR(5:&LENGTH(&STR(&B)),&STR(&B))
   SET &CSN = &LENGTH(&STR(&CCCOUNTRY))
   IF &SUBSTR(&CSN:&CSN,&STR(&CCCOUNTRY)) = &STR(;) THEN DO
    SET &CCCOUNTRY = &SUBSTR(1:&CSN-1,&STR(&CCCOUNTRY))
   END
  END
 END
 SET &NOQUAL = &PERIOD + 1
```

```
     SET &&QUAL&NOQUAL = &STR(&QUALIFIER)
    END
   END
   IF &STR(&SURNAME) NE &STR() AND &STR(&GIVENNAME) NE &STR() AND +
   &STR(&CCCOUNTRY) NE &STR() THEN DO
    IF &LENGTH(&STR(&CCCOUNTRY)) > 1 THEN DO
     SET &CCCOUNTRY = &SUBSTR(1:2,&STR(&CCCOUNTRY))
    END
    IF &STR(&CCCOUNTRY) = &STR(PW) THEN DO
     SET &CCCOUNTRY = &STR(AM)
    END
    SET &X400CONV = YES
   END
  END
  IF &SYSENV NE FORE THEN DO
   WRITE SURNAME=&STR(&SURNAME)
   WRITE GIVENNAME=&STR(&GIVENNAME)
   WRITE COUNTRY=&STR(&CCCOUNTRY)
  END
  IF &X400CONV NE YES THEN DO
   WRITE &STR(===> Error: X400-address could not be converted.)
   SET &RET = 0
   SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
   /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
   SET &RET = &LASTCC
   SET &ZISPFRC = 4
   ISPEXEC VPUT (ZISPFRC) SHARED
   EXIT CODE(&ZISPFRC)
  END
  SET &X400 = YES
 END
END
IF &LENGTH(&STR(&SYSNSUB(1,&RC))) < 8 THEN DO
 SET &RCPF2 = &SUBSTR(1:2,&STR(&SYSNSUB(1,&RC)))
 SET &SRCHSPC = &STR(.)
 SET &STARTSPC = 1
 SET &LOCSPC = &SYSINDEX(&STR(&SRCHSPC),&STR(&SYSNSUB(1,&RC)),&STARTSPC)
 IF &LOCSPC = 0 THEN DO
  IF &STR(&SYSLC(&SYSNSUB(1,&DOMAIN))) = +
  &STR(&SYSLC(&STO)) THEN DO   /* CC:mail domain and MVS user-id */
   /* test for user-id available as TSO user instead, length < 8 */
   SET &SPEC  = &SYSXSPECIAL               /* CLIST built-in function */
   NOBREAK
   &SYSXSPECIAL(ON)                        /* CLIST built-in function */
   SET &RET = 0
   SET &SYSOUTTRAP = 999999
   CONTROL MSG
   TSOEXEC &STR(LU &SYSNSUB(1,&RC) NORACF TSO)
   SET &LURET = &RET
   SET &SYSOUTTRAP = 0
   SET &MAXTSO = &SYSOUTLINE
```

```
   IF &SYSCAPS(&STR(&DEBUG)) NE DEBUG THEN DO
    CONTROL NOMSG
   END
   &SYSXSPECIAL(&SPEC)                    /* CLIST built-in function */
   BREAK
   IF &LURET = Ø THEN DO
    SET &N = Ø
    SET &RET = Ø
    DO WHILE &N < &MAXTSO
     SET &N = &N + 1
     SET &SYSDVAL = &STR(&SYSNSUB(2,&&SYSOUTLINE&N))
     SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
     READDVAL &B1 &B2 &B3 &B4 &B5 &B6 &B7 &B8
     IF &STR(&B1) = &STR(NO) AND &STR(&B2) = &STR(TSO) THEN DO
      SET &LURET = 4
     END
    END
   END
   IF &LURET = Ø THEN DO
    SET &TSOEXIST = YES
   END
  END
 END
END
IF &STR(&CCREQ) NE YES THEN DO
 SET &RET = Ø
 ALLOC FI(&STR(&MFILEID)) DUMMY +
 RECFM(F B) LRECL(8Ø) /* No Reuse could be preallocatd */
 IF &RET > Ø THEN DO        /* does mailfile exist */
  SET &MEMOFI = YES
 END
 ALLOC FI(SYSPRINT) DUMMY REUSE
END
IF &TSOEXIST = YES THEN DO
 SET &DOMAIN = &STR(&SYSLC(&SYSNSUB(1,&HOSTNAME)))
 WRITE &STR(DOMAIN reset to &HOSTNAME due to receiver is TSO-user.)
END
IF &STR(&CCREQ) = YES THEN DO
 IF &SYSLC(&STR(&SYSNSUB(1,&DOMAIN))) = &SYSLC(&STR(&HOSTNAME)) AND +
 &SYSLC(&STR(&SYSNSUB(1,&RC))) = &SYSLC(&STR(&SYSUID)) THEN DO
  WRITE ====> Warning: You are creating mail which will be sent to +
  yourself as CC (&SYSICMD).
  SLEEP 4
 END
 ISPEXEC VPUT (HOSTNAME,SMTPNODE,OWNNODE,NICKOWN)
 ISPEXEC VPUT (DOMAIN)
 SET &RET = Ø
 /* Dont call FRSC for CC-request, no allocations need to be freed */
 SET &ZISPFRC = Ø
 ISPEXEC VPUT (ZISPFRC) SHARED
 EXIT CODE(&ZISPFRC)
```

```
END
IF &STR(&DS) = &STR() THEN DO
 IF &STR(&MFILEID) = SYSUT1 THEN DO
  SET &MEMOFI = &STR()
 END
 IF &MEMOFI = YES THEN DO   /* does mailfile exist */
  LISTDSI &STR(&MFILEID) FILE
  SET &DSIRET = &LASTCC
  IF &DSIRET > 4 THEN DO      /* SYSIN */
   SET &DSTYPE = &STR(RECFM(F B) LRECL(8Ø))
  END
  ELSE DO
   SET &DSTYPE = &STR(LIKE('&SYSDSNAME'))
  END
  DEL '&DSPREF..&JOBNM..&TSTAMP..SYSUT1.OUTLIST'
  SET &CNT = Ø
  SET &RET = Ø
  /* coordinate space with maxmailbytes-parm (32m in this case) */
  ALLOC FI(SYSUT1) NEW SPACE(Ø &SECALC) CYLINDERS +
  &STR(&DSTYPE) DSORG(PS) +
  DA('&DSPREF..&JOBNM..&TSTAMP..SYSUT1.OUTLIST') +
  MGMTCLAS(TEMP) STORCLAS(TEMP) REUSE
  SET &ALCRET = &RET
  DO WHILE &ALCRET NE Ø AND &CNT < &MAXCNT
   SET &TSTAMP = +
   &STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)+
   &SUBSTR(7:8,&SYSTIME)
   SET &RET = Ø
  /* coordinate space with maxmailbytes-parm (32m in this case) */
   ALLOC FI(SYSUT1) NEW SPACE(Ø &SECALC) CYLINDERS +
   &STR(&DSTYPE) DSORG(PS) +
   DA('&DSPREF..&JOBNM..&TSTAMP..SYSUT1.OUTLIST') +
   MGMTCLAS(TEMP) STORCLAS(TEMP) REUSE
   SET &ALCRET = &RET
   IF &CNT > Ø THEN DO
    WRITE Not able to allocate +
    &DSPREF..&JOBNM..&TSTAMP..SYSUT1.OUTLIST
    SET &RET = Ø
    SYSCALL FRSC &JOBNM &TSTAMP &MFILEID +
    &VIO &DSPREF DEBUG(&STR(&DEBUG))
    /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
    SET &RET = &LASTCC
    SET &ZISPFRC = 8
    ISPEXEC VPUT (ZISPFRC) SHARED
    EXIT CODE(&ZISPFRC)
   END
   SET &CNT = &CNT + 1
   SLEEP 2
  END
  SET &MAXREC = (&MAXBYTES+&MAXSMTPLR-1)/&MAXSMTPLR  /*MAX NO OF LOOPS*/
  SET &CNT = Ø
```

```
  SET &RET = Ø
  OPENFILE &STR(&MFILEID) INPUT
  OPENFILE SYSUT1 OUTPUT
  IF &RET = Ø THEN DO
   SET &RET = Ø
   DO WHILE &RET NE 4ØØ AND &CNT < &MAXREC
    SET &CNT = &CNT + 1
    IF &CNT = &MAXREC THEN DO
     WRITE ====> Error; dataset too large, more than &MAXREC records,
     WRITE ====> dataset will be truncated (&SYSICMD).
    END
    SET &RET = Ø
    /* GETFILE CAN NOT USE VARIABLE &MEMOFILE, WILL GIVE RC332 */
    GETFILE &STR(&MFILEID)
    IF &RET = Ø THEN DO
     SET &SYSUT1 = &STR(&SYSNSUB(2,&&&MFILEID))
     PUTFILE SYSUT1
    END
   END
  END
  CLOSFILE &STR(&MFILEID)
  CLOSFILE SYSUT1
 END
END
SET &RET = Ø
%WAITUNTJ &STR(&TRP) WAIT(1)      /* test for SMTP-stc existing */
SET &WAITRET = &RET
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
 WRITE ======> Reentering &SYSICMD <======
END
IF &WAITRET NE Ø AND &SYSENV = FORE THEN DO
  /* if smtp, deliver mail it will wait on the spool. */
 IF &STR(&SYSLC(&SMTPNODE)) = &STR(&SYSLC(&OWNNODE)) THEN DO
  WRITE &STR(====> &TRP is not running for the time being.)
 END
END
IF &STR(&DS) NE &STR() THEN DO
 IF &SYSDSN(&STR('&DS')) NE OK THEN DO
  WRITE ====> Error: &STR(&DS) &SYSDSN(&STR('&DS') (&SYSICMD).)
  SET &RET = Ø
  SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &ZISPFRC = 8
  ISPEXEC VPUT (ZISPFRC) SHARED
  EXIT CODE(&ZISPFRC)
 END
 ALLOC FI(&STR(&MFILEID)) DA('&DS') SHR REUSE
END
SET &RET = Ø
LISTDSI '&DS'
```

```
SET &LRET = &LASTCC
SET &LAER = &MAXSMTPLR
SET &RF = F
IF &STR(&DS) NE &STR() THEN DO
 IF &STR(&SYSDSORG) NE PS AND &STR(&SYSDSORG) NE PO THEN DO
  WRITE ====> Error: Dataset organization not sequential nor +
  partitioned (&SYSICMD).
  SET &RET = Ø
  SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &ZISPFRC = 8
  ISPEXEC VPUT (ZISPFRC) SHARED
  EXIT CODE(&ZISPFRC)
 END
 IF &LRET < 8 AND &SYSREASON NE 5 THEN DO
  SET &LDS = &LENGTH(&STR(&DS))
  IF &STR(&SYSDSORG) = PO AND &SUBSTR(&LDS:&LDS,&STR(&DS)) NE &STR()) +
  THEN DO
   WRITE ====> Error: Dataset partitioned, but member not specified +
   (&SYSICMD).
   SET &RET = Ø
   SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
   /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
   SET &RET = &LASTCC
   SET &ZISPFRC = 8
   ISPEXEC VPUT (ZISPFRC) SHARED
   EXIT CODE(&ZISPFRC)
  END
  SET &LAER = &SYSLRECL
  SET &MAXRECL = &MAXSMTPLR
  IF &SYSLRECL > &MAXRECL THEN DO
   WRITE ====> Error: Record length &SYSLRECL higher than &MAXRECL +
   (&SYSICMD).
   SET &RET = Ø
   SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
   /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
   SET &RET = &LASTCC
   SET &ZISPFRC = 8
   ISPEXEC VPUT (ZISPFRC) SHARED
   EXIT CODE(&ZISPFRC)
  END
  IF &SYSLRECL = Ø THEN DO
   WRITE ====> Error: Record length is &SYSLRECL (&SYSICMD).
   SET &RET = Ø
   SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
   /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
   SET &RET = &LASTCC
   SET &ZISPFRC = 8
   ISPEXEC VPUT (ZISPFRC) SHARED
   EXIT CODE(&ZISPFRC)
```

```
  END
 END
 ELSE DO
  WRITE ====> Error: Dataset error &LRET &SYSREASON from Listdsi +
  (&SYSICMD).
  SET &RET = Ø
  SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &ZISPFRC = 8
  ISPEXEC VPUT (ZISPFRC) SHARED
  EXIT CODE(&ZISPFRC)
 END
END
DEL '&DSPREF..&JOBNM..&TSTAMP..MAILFILE.OUTLIST'
SET &CNT = Ø
SET &NOBLK = 27998 / &LAER             /* use half trk 339Ø blocking */
SET &BLKLAER = &NOBLK * &LAER
  /* coordinate space with maxmailbytes-parm (32m in this case) */
SET &RET = Ø
ALLOC FI(MAILFILE) NEW SPACE(1 &SECALC) CYLINDERS RECFM(&RF B) +
LRECL(&LAER) BLKSIZE(&BLKLAER) +
DA('&DSPREF..&JOBNM..&TSTAMP..MAILFILE.OUTLIST') +
MGMTCLAS(TEMP) STORCLAS(TEMP) REUSE
SET &ALCRET = &RET
DO WHILE &ALCRET NE Ø AND &CNT < &MAXCNT
 SET &TSTAMP = +
 &STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
  /* coordinate space with maxmailbytes-parm (32m in this case) */
 SET &RET = Ø
 ALLOC FI(MAILFILE) NEW SPACE(1 &SECALC) CYLINDERS RECFM(&RF B) +
 LRECL(&LAER) BLKSIZE(&BLKLAER) +
 DA('&DSPREF..&JOBNM..&TSTAMP..MAILFILE.OUTLIST') +
 MGMTCLAS(TEMP) STORCLAS(TEMP) REUSE
 SET &ALCRET = &RET
 IF &CNT > Ø THEN DO
  WRITE Not able to allocate &DSPREF..&JOBNM..&TSTAMP..MAILFILE.OUTLIST
  SET &RET = Ø
  SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &ZISPFRC = 8
  ISPEXEC VPUT (ZISPFRC) SHARED
  EXIT CODE(&ZISPFRC)
 END
 SET &CNT = &CNT + 1
 SLEEP 2
END
SET &DSNM = &DSPREF..&JOBNM..&TSTAMP..MAILFILE.OUTLIST
SET &CNT = Ø
SET &RET = Ø
```

```
SET &SEND = YES
SET &SENDSUP = NO
SET &FUNCTION = &STR()
SET &REPLY = &STR()
ISPEXEC VPUT (SENDSUP,FUNCTION,REPLY)
IF &SYSLC(&STR(&SYSNSUB(1,&DOMAIN))) = &SYSLC(&STR(&HOSTNAME)) AND +
&SYSLC(&STR(&SYSNSUB(1,&RC))) = &SYSLC(&STR(&SYSUID)) THEN DO
 WRITE ====> Warning: You are creating mail which will be send to +
 yourself (&SYSICMD).
 SLEEP 1
END
IF &STR(&REMAIL) NE YES THEN DO
 IF &X400CONV = YES THEN DO
  SET &RC = &STR(&GIVENNAME..&SURNAME)
 END
 SET &YY = &SUBSTR(1:2,&STR(&SYSSDATE))
 SET &MM = &SUBSTR(4:5,&STR(&SYSSDATE))
 SET &DD = &SUBSTR(7:8,&STR(&SYSSDATE))
 SET &CTY =  &STR(19)
 IF &YY <= 96 THEN DO
  SET &CTY = &CTY + 1
  SET &CTY = &SUBSTR(1:2-&LENGTH(&CTY),&STR(00))&CTY /* pad with zero */
 END
 SET &A01 = Jan            /* build table of months */
 SET &A02 = Feb            /* build table of months */
 SET &A03 = Mar            /* build table of months */
 SET &A04 = Apr            /* build table of months */
 SET &A05 = May            /* build table of months */
 SET &A06 = Jun            /* build table of months */
 SET &A07 = Jul            /* build table of months */
 SET &A08 = Aug            /* build table of months */
 SET &A09 = Sep            /* build table of months */
 SET &A10 = Oct            /* build table of months */
 SET &A11 = Nov            /* build table of months */
 SET &A12 = Dec            /* build table of months */
 SET &A = &STR(A&MM)
 SET &MONTH = &&&A         /* get actual month name from table */
 SET &RET = 0
 OPENFILE MAILFILE OUTPUT
 SET &MAILFILE = &STR(helo &SYSLC(&HOSTNAME))
 PUTFILE MAILFILE
 SET &MAILFILE = &STR(mail from:<&MAILUS&ATSIGN&SYSLC(&HOSTNAME)>)
 PUTFILE MAILFILE
 SET &MAILFILE = &STR(rcpt to:<&SYSLC(&SYSNSUB(1,&RC))+
 &ATSIGN&SYSLC(&SYSNSUB(1,&DOMAIN))>)
 PUTFILE MAILFILE
 SET &MAILFILE = &STR(data)
 PUTFILE MAILFILE
 SET &MAILFILE = &STR(Date:     &DD &MONTH &CTY&YY &SYSTIME +
 &STR(&TIMEOFF))
 PUTFILE MAILFILE
```

```
SET &MAILFILE = &STR(From:     &MAILUS&ATSIGN&SYSLC(&HOSTNAME)) +
&STR(&SYSNSUB(1,&BATCHUS))
PUTFILE MAILFILE
SET &MAILFILE = &STR(To:       &SYSLC(&SYSNSUB(1,&RC))+
&ATSIGN&SYSLC(&SYSNSUB(1,&DOMAIN)))
PUTFILE MAILFILE
SET &MAILFILE = &STR(Subject:  &SYSNSUB(1,&ID))
PUTFILE MAILFILE
SET &MAILFILE = &STR()
PUTFILE MAILFILE
IF &STR(&REMAIL) = REPLY AND &SYSISPF = &STR(ACTIVE) AND +
&SYSENV = FORE AND &STR(&DS) NE &STR() THEN DO
 SET &MAILFILE = +
 &STR(_____ Reply Separator _____+
 _____)
 PUTFILE MAILFILE
END
IF &STR(&REMAIL) = FORWARD AND &SYSISPF = &STR(ACTIVE) AND +
&SYSENV = FORE AND &STR(&DS) NE &STR() THEN DO
 SET &MAILFILE = +
 &STR(_____ Forward header _____+
 _____)
 PUTFILE MAILFILE
END
CLOSFILE MAILFILE
IF &SYSISPF = &STR(ACTIVE) AND &SYSENV = FORE AND +
&STR(&DS) = &STR() THEN DO
 SET &FUNCTION = MAIL
 ISPEXEC VPUT (FUNCTION,SEND) /*info to evt call of ZNOMAIL*/
 SET &REPLY = &STR()
 ISPEXEC VPUT (REPLY)           /* info to ZSMTPN */
 TSOLINE1                       /* Clear screen */
 %EDITRECV                      /* take edit recovery */
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 ISPEXEC EDIT DATASET('&DSNM') MACRO(%ZSMTPN)
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 ISPEXEC VGET (SEND)
 IF &STR(&SEND) = NO THEN DO
  SET &APIRET = 4
  SET &SENDSUP = YES
  ISPEXEC VPUT (SENDSUP)
 END
 IF &STR(&SEND) = CANCEL THEN DO
  SET &SENDSUP = YES
  ISPEXEC VPUT (SENDSUP)
  SET &SEND = NO
  ISPEXEC VPUT (SEND)
```

```
   SLEEP 2
   TSOLINE1              /* clear screen */
   SET &RET = Ø
   SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
   /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
   SET &RET = &LASTCC
   SET &ZISPFRC = 8
   ISPEXEC VPUT (ZISPFRC) SHARED
   EXIT CODE(&ZISPFRC)
  END
 END
END
IF &STR(&DS) NE &STR() OR (&STR(&DS) = &STR() AND &SYSENV NE FORE) +
THEN DO
 IF &STR(&DS) NE &STR() THEN DO
  ALLOC FI(SYSUT1) DA('&DS') SHR REUSE
 END
 ELSE DO
  ALLOC FI(SYSUT1) DUMMY +
  RECFM(F B) LRECL(&LAER) /* NO REUSE, EVT PREALLOC */
 END
 ALLOC FI(SYSUT2) DA('&DSNM') MOD REUSE
 ALLOC FI(SYSIN) DUMMY REUSE
 IF &SYSENV NE FORE THEN DO
  ALLOC FI(SYSPRINT) DA(*) REUSE
 END
 ELSE DO
  ALLOC FI(SYSPRINT) DUMMY REUSE
 END
 PROFILE NOWTPMSG
 SET &SYSOUTTRAP = 999999
 SET &RET = Ø
 /*  CALL 'SYS1.LINKLIB(IEBGENER)'                                    */
 /*  Use ICEGENER instead of IEBGENER due to evt difference in        */
 /*  Lrecl on sysut1 and sysut2.                                      */
 /*  Icegener sets cc=4 when record lengths are different.            */
 CALL 'SYS1.LPALIB(ICEGENER)'
 SET &GENRET = &RET
 SET &SYSOUTTRAP = Ø
 PROFILE WTPMSG
 IF &GENRET > 4 THEN DO
  IF &GENRET = SD37 THEN DO
   WRITE ====> Error: Mail is too large, can max be &MAXBYTES bytes +
   (&SYSICMD).
  END
  ELSE DO
   /* in case of conflicting DCB info (recfm v or u) use INSØ25M */
   SET &GENRET = Ø
   PROFILE NOWTPMSG
   SET &SYSOUTTRAP = 999999
   SET &RET = Ø
```

```
   CALL 'YOUR.LOAD.LIBRARY(INSØ25M)'
   SET &GENRET = &RET
   SET &SYSOUTTRAP = Ø
   PROFILE WTPMSG
   IF &GENRET > 4 THEN DO
    WRITE ====> Error: IEBGENER copying mail gave rc &GENRET (&SYSICMD).
    SET &APIRET = 2Ø
    SET &SEND = NO
   END
  END
 END
END
IF &STR(&REMAIL) = REPLY AND &SYSISPF = &STR(ACTIVE) AND +
&SYSENV = FORE AND &STR(&DS) NE &STR() THEN DO
 SET &FUNCTION = MAIL
 ISPEXEC VPUT (FUNCTION,SEND) /* info to evt call of ZNOMAIL         */
 SET &REPLY = YES
 /* Info to ZSMTPN */
 ISPEXEC VPUT (REPLY,HOSTNAME,SMTPNODE,OWNNODE,NICKOWN)
 /* dont clear screen, keep evt cc-receivers on same screen */
 %EDITRECV                   /* take edit recovery */
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 ISPEXEC EDIT DATASET('&DSNM') MACRO(%ZSMTPN)
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 ISPEXEC VGET (SEND)
 IF &STR(&SEND) = NO THEN DO
  SET &APIRET = 4
  SET &SENDSUP = YES
  ISPEXEC VPUT (SENDSUP)
 END
 IF &STR(&SEND) = CANCEL THEN DO     /* cancel from Zreply */
  SET &SENDSUP = YES
  ISPEXEC VPUT (SENDSUP)
  SET &SEND = NO
  ISPEXEC VPUT (SEND)
  SLEEP 2
  TSOLINE1              /* clear screen */
  SET &RET = Ø
  SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &ZISPFRC = 8
  ISPEXEC VPUT (ZISPFRC) SHARED
  EXIT CODE(&ZISPFRC)
 END
END
IF &SYSENV NE FORE THEN DO
```

```
 ALLOC FI(SYSPRINT) DA(*) REUSE
 ALLOC FI(SYSIN) DUMMY REUSE
 ALLOC FI(SYSUT1) DA('&DSNM') SHR REUSE
 ALLOC FI(SYSUT2) DA(*) REUSE
 CALL 'SYS1.LINKLIB(IEBGENER)'
END
ELSE DO
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG OR +
 &SYSCAPS(&STR(&DEBUG)) = SHOWS THEN DO
  ALLOC FI(SYSPRINT) DA(*) REUSE
  ALLOC FI(SYSIN) DUMMY REUSE
  ALLOC FI(SYSUT1) DA('&DSNM') SHR REUSE
  ALLOC FI(SYSUT2) DA(*) REUSE
  CALL 'SYS1.LINKLIB(IEBGENER)'
 END
END
IF &SYSISPF = &STR(ACTIVE) THEN DO
 SET &QUIET = YES
 ISPEXEC VPUT (QUIET)
 %EDITRECV                    /* take edit recovery */
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 SET &RET = Ø
 ISPEXEC EDIT DATASET('&DSNM') MACRO(%ZSMTPC)
 SET &EDITRET = &RET
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 SET &QUIET = &STR()
 ISPEXEC VPUT (QUIET)
 IF &EDITRET < 8 THEN DO
  ISPEXEC VGET (BHITS,NBHITS,SYNTAX)
  SET &BYTES = &BHITS + &NBHITS
  IF &BYTES > &MAXBYTES OR &BYTES = Ø OR &SYNTAX = ERROR THEN DO
   SET &APIRET = 2Ø
   SET &SEND = NO
   IF &BYTES > &MAXBYTES THEN DO
    WRITE ====> Error: Mail is &BYTES, can max be &MAXBYTES bytes +
    (&SYSICMD).
   END
   IF &BYTES = Ø THEN DO
    WRITE ====> Error: Mail is &BYTES characters, mail is not sent +
    (&SYSICMD).
   END
  END
  ELSE DO
   IF &SYSENV NE FORE THEN DO
    WRITE Mail contains Blanks     = &BHITS Bytes
    WRITE Mail contains Non-Blanks = &NBHITS Bytes
    WRITE Mail contains Total      = &EVAL(&NBHITS + &BHITS) Bytes
```

```
   END
  END
 END
END
ISPEXEC CONTROL DISPLAY LINE START(14)
IF &SEND = YES THEN DO
 SET &SYSOUTTRAP = 999999
 CONTROL MSG
 SET &RET = Ø
 TRANSMIT &STR(&SMTPNODE..&TRP) +
 DATASET('&DSNM') NOEPILOG NOLOG NOPROLOG
 SET &APIRET = &RET
 SET &SYSOUTTRAP = Ø
 SET &MAXLNE = &SYSOUTLINE
 IF &SYSCAPS(&STR(&DEBUG)) NE DEBUG THEN DO
  CONTROL NOMSG
 END
 SET &XMITUSE = &STR()
 SET &N = Ø
 SET &RET = Ø
 DO WHILE &N < &MAXLNE
  SET &N = &N + 1
  SET &SYSDVAL = &STR(&SYSNSUB(2,&&SYSOUTLINE&N))
  SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
  READDVAL &C1 &C2 &C3 &C4 &C5 &C6 &C7 &C8
  WRITE &STR(&C1) &STR(&C2) &STR(&C3) &STR(&C4) +
  &STR(&C5) &STR(&C6) &STR(&C7) &STR(&C8)
  IF &LENGTH(&STR(&C1)) >= 8 THEN DO
   SET &MSGPFX = &SUBSTR(1:8,&STR(&C1))
   IF &STR(&MSGPFX) = INMXØ7ØI OR &STR(&MSGPFX) = INMXØ72I THEN DO
    /* permanent io-error or null dataset */
    SET &XMITUSE = UNUSABLE
    SET &N = &MAXLNE
   END
  END
 END
END
IF &APIRET = Ø THEN DO
 WRITE &STR(====> Mail successfully distributed to &TRP)
 IF &STR(&SYSNSUB(1,&HOSTNAME)) = &STR(&SYSNSUB(1,&DOMAIN)) THEN DO
  IF &SYSENV = FORE THEN DO
   SE 'SMTPØØ4I Mail sent to you from' USER(&STR(&SYSNSUB(1,&RC))) LOGON
   /* Sending user-id will automatically be appended by send */
  END
  ELSE DO
   SE 'SMTPØØ4I Mail sent to you from &JOBNM' +
   USER(&STR(&SYSNSUB(1,&RC))) LOGON
  END
 END
END
IF &APIRET NE Ø THEN DO
```

```
 IF &SENDSUP NE YES THEN DO
  WRITE ====> Error: transmit to &trp failed with rc &apiret +
  (&SYSICMD). <====
 END
 SET &SEND = NO
 IF &XMITUSE = UNUSABLE OR &BYTES = Ø THEN DO
  SET &RET = Ø
  SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &ZISPFRC = 8
  ISPEXEC VPUT (ZISPFRC) SHARED
  EXIT CODE(&ZISPFRC)
 END
END
  /* save a copy if TSO-online or TSO user submitted batch */
IF &STR(&SYSISPF) = ACTIVE AND &SYSUID NE &STR() AND +
&STR(&REROUTE) NE YES THEN DO
 IF &SYSDSN('&DSPREF..&TRP') = OK THEN DO
  LISTDSI '&DSPREF..&TRP'
  /* SMTP DOES NOT SUPPORT VARIABLE RECORDS */
  IF &STR(&SYSDSORG) NE PO AND &STR(&SYSLRECL) NE &MAXSMTPLR AND +
  &STR(&SYSRECFM) NE FB THEN DO
   DEL '&DSPREF..&TRP'
  END
 END
 IF &SYSDSN('&DSPREF..&TRP') NE OK THEN DO
  ALLOC FI(ZXCVZXCV) NEW CATALOG SPACE(1 3) CYLINDERS DIR(45) +
  RECFM(F B) LRECL(&MAXSMTPLR) BLKSIZE(&BLKSMTPLR) +
  MGMTCLAS(STANDARD) STORCLAS(STANDARD) +
  DA('&DSPREF..&TRP') REUSE
 END
 SET &PFX = O
 SET &SUF = Ø
 SET &CNT = Ø
 SET &MAXNRT = &EVAL(29*1Ø)
 SET &CPYRET = 4
 DO WHILE &CPYRET > Ø AND &CNT < &MAXNRT
  SET &CNT = &CNT + 1
  SET &NAME = &STR(&PFX)+
  &SUBSTR(1:2,&SYSSDATE)&SUBSTR(4:5,&SYSSDATE)&SUBSTR(7:8,&SYSSDATE)+
  &SUF
  SET &RET = Ø
  %ISPFCPY &NAME &DSNM &DSPREF..&TRP TYPE(DSNAME) +
  REPLACE(NOREPLACE) DEBUG(&STR(&DEBUG))
  SET &CPYRET = &RET
  IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
   WRITE ======> Reentering &SYSICMD <======
  END
  IF &CPYRET = Ø THEN DO
   SET &LOGDS = &TRP
```

```
 ISPEXEC VPUT (LOGDS)
 SET &FUNCTION = MAIL
 ISPEXEC VPUT (FUNCTION)
 %EDITRECV                    /* take edit recovery */
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 ISPEXEC EDIT DATASET('&DSPREF..&TRP(&NAME)') MACRO(%ZSMTPR)
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
 ISPEXEC VGET (ZSMTPR)
 IF &STR(&ZSMTPR) NE ERROR AND &STR(&ZSNTPR) NE SEVERROR THEN DO
  ISPEXEC LMINIT DATAID(DID) DATASET('&DSPREF..&TRP') ENQ(SHRW)
  ISPEXEC LMOPEN DATAID(&DID) OPTION(OUTPUT)
  SET &RET = Ø
  ISPEXEC LMMDEL DATAID(&DID) MEMBER(&NAME) NOENQ
  SET &LMMDELRET = &RET
  IF &LMMDELRET > Ø THEN DO
   SET &MAXCC = Ø
   SET &RET = Ø
   ISPEXEC LMMDEL DATAID(&DID) MEMBER(&NAME)
   SET &LMMDELRET = &RET
  END
  ISPEXEC LMCLOSE DATAID(&DID)
  ISPEXEC LMFREE DATAID(&DID)
  IF &LMMDELRET > Ø THEN DO
   ALLOC FI(DELZXCVB) DA('&DSPREF..&TRP') SHR REUSE
   DEL '&DSPREF..&TRP(&NAME)' FILE(DELZXCVB)
   FREE FI(DELZXCVB)
  END
 END
 ELSE DO
  IF &STR(&ZSMTPR) NE SEVERROR THEN DO
   SET &ZEDLMSG = +
   &STR(====> Mail saved to &DSPREF..&TRP(&NAME) <====)
   WRITE &STR(&ZEDLMSG)
   ISPEXEC LMINIT DATAID(DID) +
   DATASET(&STR('&DSPREF..&TRP')) ENQ(SHRW)
   ISPEXEC LMMSTATS DATAID(&DID) MEMBER(&NRSTR(&NAME)) +
   USER(MAILED)
   ISPEXEC LMFREE DATAID(&DID)
  END
  ELSE DO
   SET &CPYRET = 16    /* SIMULATE COPY ERROR */
  END
 END
 SET &ZSMTPR = &STR()
 ISPEXEC VPUT (ZSMTPR)
END
IF &CPYRET > 12 THEN DO
```

```
   SET &ZEDLMSG = +
   &STR(====> Error saving mail to &DSPREF..&TRP rc &CPYRET +
   (&SYSICMD). <====)
   ISPEXEC SETMSG MSG(ISRZØØ1)
   SET &CPYRET = Ø    /* TERMINATE LOOP */
  END
  SET &SUF = &SUF + 1
  IF &SUF > 9 THEN DO
   SET &SUF = Ø
   SET &STARTSPC = 1
   SET &RC = &STR(ABCDEFGHIJKLMNOPQRSTUVWXYZ$#@)
   SET &LOCSPC = &SYSINDEX(&STR(&PFX),&STR(&SYSNSUB(1,&RC)),&STARTSPC)
   IF &LOCSPC = Ø THEN DO
    SET &ZEDLMSG = +
    &STR(====> Error saving mail to &DSPREF..&TRP rc &CPYRET +
    (&SYSICMD). <====)
    ISPEXEC SETMSG MSG(ISRZØØ1)
    SET &CPYRET = Ø    /* TERMINATE LOOP */
   END
   SET &RET = Ø
   SET PFX = &SUBSTR(&LOCSPC+1:&LOCSPC+1,&STR(&SYSNSUB(1,&RC)))
   IF &RET > Ø THEN DO
    SET &ZEDLMSG = +
    &STR(====> Error saving mail to &DSPREF..&TRP rc &CPYRET +
    (&SYSICMD). <====)
    ISPEXEC SETMSG MSG(ISRZØØ1)
    SET &CPYRET = Ø    /* TERMINATE LOOP */
   END
  END
 END
END
IF &APIRET NE Ø THEN DO
 IF &SENDSUP NE YES THEN DO
  WRITE ====> Error: Transmit to &TRP failed with rc &APIRET +
  (&SYSICMD). <====)
 END
END
SET &RET = Ø
SYSCALL FRSC &JOBNM &TSTAMP &MFILEID &VIO &DSPREF DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
SET &RET = &LASTCC
SET &ZISPFRC = &APIRET
ISPEXEC VPUT (ZISPFRC) SHARED
EXIT CODE(&ZISPFRC)
 /*                                                                  */
 /*  INLINE SUBROUTINES                                              */
 /*                                                                  */
VARSTRNG: +
PROC 1 RSTRING STRING() DEBUG(NEBUG)
 /*                                                                  */
 /* INLINE VARSTRNG ROUTINE; TRUNCATE TRAILING BLANKS                */
```

```
/*                                                                    */
CONTROL NOMSG NOFLUSH NOLIST NOCONLIST NOSYMLIST
ATTN DO
 SET &FLUSH = FLUSH         /* NEXT STATEMENT MUST BE NULL LINE     */

END
ERROR DO
 SET &RET = &LASTCC
 RETURN
END
IF &SYSCAPS(&STR(&DEBUG)) NE DEBUG THEN DO
 ISPEXEC VGET (DEBUG)
END
IF &STR(&DEBUG) = &STR() THEN DO
 SET &DEBUG = NEBUG
END
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
END
IF &FLUSH = FLUSH THEN DO
 SET &DEBUG = NEBUG
 ISPEXEC VPUT (DEBUG) SHARED
 SET &ZEDSMSG = &str(Function interrupted)
 ISPEXEC SETMSG MSG(ISRZØØ1)
 RETURN CODE(Ø)
END
ISPEXEC CONTROL ERRORS RETURN
SYSREF &RSTRING
SET &RSTRING = &STR(&SYSNSUB(1,&STRING))
IF &STR(&SYSNSUB(1,&STRING)) = &STR() THEN DO
 RETURN CODE(Ø)
END
SET &RET = Ø
SET &ALLBLANKS = +
&STR(                                                                  -
                                                                       -
                                                                       -
                                                                       )
IF &LENGTH(&SYSNSUB(1,&STRING)) > Ø THEN DO
 SET &SRCHSPC = &STR( )    /* look for last blank */
 SET &STARTSPC = 1
 SET &LOCSPC = 1
 SET &LASTSPC = Ø
 SET &LENRCP = &LENGTH(&STR(&SYSNSUB(1,&STRING)))
 SET &MAXT = &LENRCP
 SET &R = Ø
 DO WHILE &LOCSPC > Ø AND &STARTSPC <= &LENRCP AND &R < &MAXT
  SET &R = &R + 1
  SET &LOCSPC = +
  &SYSINDEX(&STR(&SRCHSPC),&STR(&SYSNSUB(1,&STRING)),&STARTSPC)
  IF &LOCSPC > Ø THEN DO
```

```
    SET &STARTSPC = &LOCSPC + 1
    SET &LASTSPC = &LOCSPC
    IF &LASTSPC > Ø AND &LENRCP > &EVAL(&LASTSPC + 1) +
    THEN DO /* REST BLANKS */
     IF &SUBSTR(&LASTSPC:&LENRCP,&STR(&SYSNSUB(1,&STRING))) = +
     &SUBSTR(1:&LENRCP-&LASTSPC-1,&STR(&ALLBLANKS)) THEN DO
      SET &STRING = +
      &SUBSTR(1:&LASTSPC-1,&STR(&SYSNSUB(1,&STRING)))
      SET &R = &MAXT
     END
    END
   END
  END
 END
 SET &RSTRING = &STR(&SYSNSUB(1,&STRING))
 RETURN CODE(Ø)
END
 /*                                                                  */
FRSC: +
PROC 5 JOBNM TSTAMP MFILEID VIO DSPREF DEBUG(NEBUG)
 /*                                                                  */
 /* INLINE RETURN ROUTINE TO FREE EVT ALLOCATED RESOURCES            */
 /*                                                                  */
 CONTROL NOMSG NOFLUSH NOLIST NOCONLIST NOSYMLIST NOCAPS
 ATTN DO
  SET &FLUSH = FLUSH         /* NEXT STATEMENT MUST BE NULL LINE     */

 END
 ERROR DO
  SET &RET = &LASTCC
  RETURN
 END
 IF &SYSCAPS(&STR(&DEBUG)) NE DEBUG THEN DO
  ISPEXEC VGET (DEBUG)
 END
 IF &STR(&DEBUG) = &STR() THEN DO
  SET &DEBUG = NEBUG
 END
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
 END
 IF &FLUSH = FLUSH THEN DO
  SET &ZEDSMSG = &str(Function interrupted)
  ISPEXEC SETMSG MSG(ISRZØØ1)
  RETURN CODE(Ø)
 END
 ISPEXEC CONTROL ERRORS RETURN
 DEL '&DSPREF..&JOBNM..&TSTAMP..SYSUT1.OUTLIST'
 DEL '&DSPREF..&JOBNM..&TSTAMP..MAILFILE.OUTLIST'
 IF &SYSENV = FORE THEN DO
  FREE FI(&STR(&MFILEID))
```

```
  ALLOC FI(SYSPRINT) DA(*) REUSE
  ALLOC FI(SYSIN) DA(*) REUSE
  ALLOC FI(SYSUT1) SPACE(Ø 5) TRACKS UNIT(&VIO) REUSE
  ALLOC FI(SYSUT2) SPACE(Ø 5) TRACKS UNIT(&VIO) REUSE
 END
 SET &SEND = YES
 ISPEXEC VPUT (SEND)
 RETURN CODE(Ø)
END
 /*                                                                  */
```

EXAMPLE JCL

The following are examples of JCL to send mail in batch.

```
//STEPØ3   EXEC ISPFBAT,REGION=6M,       ISPF BATCH
//    PARM.ISPFBAT='ISPSTART CMD(%MAILSENS RC(userid@MULP)RCDOMAIN(YES)
//             DEBUG(DEBUG)DS(USERID.JCL.CNTL(XXXX)))'
//*


//STEPØ1   EXEC ISPFBAT,REGION=6M,       ISPF BATCH
//         PARM.ISPFBAT='ISPSTART CMD(%MAILSENS RC(userid) DEBUG(nEBUG)
//             DS(USERID.JCL.CNTL(YYYYY)))'
//*


//STEP1    EXEC PGM=IKJEFTØ1,REGION=6M,DYNAMNBR=1Ø24   TSO BATCH
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
%MAILSENS RC(userid) DEBUG(nEBUG) +
DS(USERID.JCL.CNTL(XXXX))
//*


//STEPØ7   EXEC PGM=IKJEFTØ1,REGION=6M,DYNAMNBR=1Ø24
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
%MAILSENS RC(userid) MEMOFILE(MAILDATA)
//MAILDATA DD   *
mail data
mail data
//*


//STEP11   EXEC PGM=IKJEFTØ1,REGION=6M,DYNAMNBR=1Ø24
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
```

```
//SYSTSIN  DD   *
%MAILSENS RC(userid)
//MEMOFILE DD   *
mail data
mail data
//*


//STEP14   EXEC PGM=IKJEFT01,REGION=6M,DYNAMNBR=1024
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *           X400 address to be converted
%MAILSENS +
RC('G=firstn; S=lastn; O=CC; OU=server; P=installat; A=400NET; C=DK') +
MEMOFILE(SYSUT1)
//SYSUT1   DD   *
mail data
mail data
//*
```

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

Nils Plum
Systems Programmer (Denmark)

A Web server under MVS or OS/390

In the past few years there has been an explosion in the number of Web servers and sites implemented on the Internet. These Web servers are used to present information in words, pictures, sound, and animation, to the public, and to communicate information throughout an individual organization over a proprietary intranet. Web servers are also used to speed communication between businesses and their suppliers.

The World Wide Web is growing in use and many companies are considering it for in-house surfing as well, by setting up intranets. Intranets are internal Web sites, and companies are looking to use them for their client/server applications.

WHAT IS A WEB SERVER?

A Web server is software running on a host operating system such as

Unix or Windows NT, or OS/390 with TCP/IP as implemented in Open Edition, that enables a business or organization to publish information on the Internet or on a corporate intranet.

A Web server supports a system of requests and responses in which a single URL request generates additional requests for embedded objects.

Under OS/390, the TCP/IP task, TCPICSS, should be started because it's the OS/390 Web server.

The Web server PROC can be:

```
//WEBSRV PROC
//WEBSRV   EXEC PGM=IMWHTTPD,REGION=ØK,TIME=NOLIMIT,
//  PARM='RPTSTG(ON),RPTOPTS(ON)/'
//STEPLIB  DD DSN=SYS1.SIMWMOD1,DISP=SHR
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
//SYSTCPD  DD DSN=TCPIP.PARMLIB(TCPDATA),DISP=SHR
```

The TCPDATA member of TCPIP PARMLIB is:

```
 ;
 A691PWER:    TCPIPJOBNAME TCPIP
 TCPIPUSERID  TCPIP
 ;
 A691PWER:    HOSTNAME  A691PWER
 ;
 DOMAINORIGIN  rdv.dcp
 ;
 ;NSINTERADDR  1Ø.4.26.2
 ;NSPORTADDR 53
 RESOLVEVIA UDP
 RESOLVERTIMEOUT 3Ø
 RESOLVERUDPRETRIES 1
;
;TRACE RESOLVER
;
DATASETPREFIX TCPIP
;
; MESSAGECASE MIXED
; MESSAGECASE UPPER
;
```

```
; LOADDBCSTABLES JIS78KJ JIS83KJ SJISKANJI EUCKANJI HANGEUL KSC56Ø1
;                TCHINESE
;
; End of file.
```

Users utilize a Web browser to sort through a data warehouse.

A Web browser is used to access the server:

- Netscape Navigator or Internet Explorer are usually used on a PC workstation.
- Under OS/390 the TEXPLOR CLIST can be used (seeTEXPLOR.REX, TEXPLOR.PNL, TEXPLOR0.PNL, TEXPLOR1.PNL, and TEXPLOR2.PNL).

TEXPLOR*x*.PNL are ISPF panels and TEXPLOR.REX is a REXX CLIST.

Prerequisites are OS/390 Open Edition and TCP/IP.

TEXPLOR.PNL

```
)ATTR DEFAULT(%+_)
*  AREA(DYNAMIC) EXTEND(ON) SCROLL(ON)
1  TYPE(CHAR) COLOR(RED)
2  TYPE(CHAR) COLOR(GREEN)
3  TYPE(CHAR) COLOR(BLUE)
4  TYPE(CHAR) COLOR(RED)
5  TYPE(CHAR) COLOR(GREEN)
6  TYPE(CHAR) COLOR(BLUE)
7  TYPE(CHAR) COLOR(TURQ) HILITE(BLINK)
8  TYPE(CHAR) COLOR(PINK) HILITE(USCORE)
9  TYPE(CHAR) COLOR(YELLOW) HILITE(REVERSE)
#  TYPE(INPUT) CAPS(OFF)
)BODY
%TSO Explorer: &TITRE
+N.&NUMERO+Long:&LONGUEUR+
%S{lection:#choix
+ URL:%&URL
%
%PFØ7/Ø8+d{filement dans document%PF1Ø/11+document pr{c{dent/suivant
%?+liste  +
*L1,S1
*
)PROC
```

```
  &BAS=LVLINE(L1)
  &PF=.PFKEY
  &CSRC=.CURSOR
  &CSRP=.CSRPOS
)END
```

TEXPLOR.REX

```
/* rexx */

/* Visualize HTML document */

debug="";

signal on error name arret;
signal on syntax name arret;
signal on halt name arret;

call tcpcall 'initialize','webtcpip'

address ISPEXEC "DISPLAY PANEL(TEXPLORØ)";

URL=$CPURL; /* initial document        */
?doc=Ø;  /* number of current document */
?Ndoc=Ø; /* number of documents        */

/* Total number of dynamic zones attached */
address ispexec "PQUERY PANEL(TEXPLOR) AREANAME(L1) DEPTH(taille)";

do forever;
   Up_URL = translate(URL);
   select;
      when left(Up_URL,7) = "MAILTO:" then
      do;
         call Messagerie;
         URL=URL.?doc;
      end;
      when left(Up_URL,7) = "HTTP://" then
      do;
         code_retour = Doc_HTML();
         if code_retour >Ø then leave;
      end;
      otherwise
         address ispexec "DISPLAY PANEL(TEXPLOR1)";
         if RC>Ø then URL=URL.?doc;
   end;
end;

call tcpcall 'terminate';
```

```
exit Ø;

arret:
   call tcpcall 'terminate';
   exit Ø;

/*********************************************************************/

Messagerie:

   if substr(URL,8,1)="/" then adresse_email=substr(URL,9);
                          else adresse_email=substr(URL,8);
   call SMTPISPF adresse_email;

return;

/*********************************************************************/

Doc_HTML:

   parse var URL . "://" adresse_tcp "/" document "£" ref_locale;

   do ?idoc=1 to ?Ndoc
      if URL.?idoc = URL then
      do;
         ?doc=?idoc;
         code_doc = Affiche_Document();
         return code_doc
      end;
   end;

   ?gdoc = ?Ndoc+1
   URL.?gdoc = URL
   code_doc = Get_document();
   if code_doc =Ø then
   do;
      ?Ndoc=?gdoc;
      ?doc=?gdoc;
      code_doc = Affiche_Document();
   end;

return code_doc;

/*********************************************************************/

Get_document:
   destination=adresse_tcp;
   objet="/"document;
   if Get_TCPIP()¬=Ø then return 12;
```

```
   call Traitement_Document;

return Ø;

/*********************************************************/
/* acc}s object via TCP/IP                               */
/*********************************************************/
Get_TCPIP:
if debug¬="" then Say "Address of local TCP/IP" TCPCall('gethostid');
dest.addr=tcpcall('gethostbyname',destination);
dest.name=destination;
if src¬=Ø then return 12;
/*say "destination" destination "==>" dest.addr;
  say "                          >" dest.name; */
s=tcpcall('Socket');
if debug¬="" then say "Cr{ation d'une prise" s;
rc=tcpcall('setsockopt',s,'sol_socket','so_ascii','on');
dest.family="AF_INET";
dest.port=8Ø;
rc=tcpcall('connect',s,dest.family dest.port dest.addr);
if debug¬="" then Say "Connexion" rc
/* say "Correspondant" sockgetsockname(s,c.);
say c.addr c.port c.family;  */
rc=tcpcall('send',s,"GET "objet "HTTP/1.Ø "||"Ød25Ød25"x)
if debug¬="" then say "Emission: GET "objet "HTTP/1.Ø " rc
drop x buffer;
lrecu=-1;
buffer="";
do until lrecu=Ø
   x=tcpcall('recv',s,256);
   if debug¬="" then say "Chaine recue" x;
   parse var x lrecu recu
   buffer = buffer || recu
end;
rc=tcpcall('close',s)
if debug¬="" then say "Close" rc
return Ø;

/*********************************************************/
/* Treatment document                                    */
/*********************************************************/
Traitement_document:
Texte="";
mot="";
n_ctrl=Ø;
mode="T";
centrer=Ø;
Long_Ligne=8Ø;
ligne.?gdoc.="";
attribut.?gdoc="";
```

```
ligne.?gdoc.Ø=Ø;
tab=Ø;
sel=Ø;
HRef.?gdoc.="";
Href.?gdoc.Ø=Ø;
zz = lastpos("/",objet);
URL_Base = "http://" || adresse_tcp || substr(objet,1,zz);
if debug¬="" then say URL_Base;
corps=Ø;
attribut=Ø;
forme="N";
/* attributs:  Ø normal, 1 TTY,
              +2 Italique
              +4 Gras          */
/* saut entete */
parse var buffer entete "Ød25Ød25"x buffer;
if buffer="" then buffer=entete

l=enregistrement();
do while buffer¬="" | l¬="";
   do while l¬="";
      parse var l t "<" markup ">" l;
      if t¬="" then
      do
         t=accent(t);
         if forme="N" then
         do
            do z=1 to words(t);
               w=word(t,z);
               call ajout_mot w;
            end /* do */
         end /* forme N */
         else
         do
            texte=texte||t;
            l_att=l_att||substr(attribut,1,length(t),attribut);
            call affiche(Ø);
         end /* forme PRE */
      end;
      if markup¬="" then
      do;
         /* actions */
         parse var markup ctrl options;
         ctrl=translate(ctrl);
         select
            when ctrl="BODY" | ctrl="HTML" then
            do
               call affiche(Ø);
               corps=1;
            end /* when BODY */
```

```
when ctrl="/BODY" then
do
   call affiche(Ø);
   corps=Ø;
end /* when /BODY */
when ctrl="HEAD" then
do
   call affiche(Ø);
   corps=Ø;
end /* when HEAD  */
when ctrl="/HEAD" then
do
   call affiche(Ø);
   corps=1;
end /* when /HEAD  */
when ctrl="TITLE" then
do
   call affiche(Ø);
   corps=2;
end /* when TITLE */
when ctrl="/TITLE" then
do
   call affiche(Ø);
   corps=1;
end /* when /TITLE */
when ctrl="P" then call affiche(1);
when ctrl="/P" then nop;
when ctrl="PRE" then forme="PRE";
when ctrl="/PRE" then forme="N";
when ctrl="BR" then call affiche(Ø);
when ctrl="HR" then
do
   call affiche(Ø)
   texte=substr(" ",2,long_ligne-2,"_");
   call affiche(Ø)
end /* when HR */
when ctrl="H1" then call affiche(3);
when ctrl="/H1" then call affiche(1);
when ctrl="H2" then call affiche(2);
when ctrl="/H2" then call affiche(1);
when ctrl="H3" then call affiche(1);
when ctrl="/H3" then call affiche(1);
when ctrl="H4" then call affiche(1);
when ctrl="/H4" then call affiche(1);
when ctrl="H5" then call affiche(1);
when ctrl="/H5" then call affiche(Ø);
when ctrl="H6" then call affiche(Ø);
when ctrl="/H6" then call affiche(Ø);
when ctrl="CENTER" then
do
```

```
   call affiche(Ø)
   Centrer=1
end /* when CENTER */
when ctrl="/CENTER" then
do
   call affiche(Ø);
   centrer=Ø;
end /* when /CENTER */
when ctrl="LI" then call affiche(Ø);
when ctrl="UL" then
do
   call affiche(Ø)
   tab=tab+3;
end /* when UL */
when ctrl="/UL" then
do
   call affiche(Ø);
   tab=tab-3;
end /* when /UL */
when ctrl="OL" then
do
   call affiche(Ø)
   tab=tab+3;
end /* when OL */
when ctrl="/OL" then
do
   call affiche(Ø);
   tab=tab-3;
end /* when /OL */
when ctrl="MENU" then
do
   call affiche(Ø)
   tab=tab+3;
end /* when MENU */
when ctrl="/MENU" then
do
   call affiche(Ø);
   tab=tab-3;
end /* when /MENU */
when ctrl="A" then
do
   x_href=pos('HREF=',translate(options));
   if x_href¬=Ø then
   do
      x_href=x_href+5;
      if substr(options,x_href,1)='"' then
      do;
        x_href=x_href+1;
        y_href=pos('"',options,x_href);
      end
```

```
                  else
                    y_href=pos(' ',options||" ",x_href);
                  obj_nouv=substr(options,x_href,y_href-x_href);
                  sel=sel+1;
                  if translate(left(obj_nouv,7))="HTTP://" | ,
                     translate(left(obj_nouv,7))="MAILTO:" then
                     nop;
                  else
                     obj_nouv = URL_Base || obj_nouv;
                  HRef.?gdoc.sel=obj_nouv;
                  HREF.?gdoc.Ø=sel
                  call ajout_mot "<"sel">", "9";
               end
            end /* when A */
            when ctrl="IMG" then
            do
               x_alt=pos('ALT=',translate(options));
               if x_alt=Ø then txt_alt="<<Image>>";
               else
               do
                  x_alt=x_alt+5;
                  if substr(options,x_alt,1)='"' then
                  do;
                    x_alt=x_alt+1;
                    y_alt=pos('"',options,x_href);
                  end
                  else
                     y_alt=pos(' ',options,x_alt);
                  txt_alt=substr(options,x_alt,y_alt-x_alt);
               end
               call ajout_mot txt_alt;
            end;
            when ctrl="B" then attribut=bitor(attribut,"4");
            when ctrl="/B" then attribut=bitand(attribut,"fa"x);
            when ctrl="I" then attribut=bitor(attribut,"2");
            when ctrl="/I" then attribut=bitand(attribut,"fd"x);
         otherwise
            nop;
         end  /* select */
      end /* if markup */
      else
         nop;
   end /* do while line is not empty */
   if forme="P" then call affiche(Ø)
   l=enregistrement();
end /* do while donn{es pr{sentes */

call affiche(Ø);
return;
/*********************************************************/
/* Recording following                                   */
```

```
/*********************************************************/
enregistrement: procedure expose buffer;
   parse var buffer x "25"X buffer;
return translate(x," ","ØD"x);

/*********************************************************/
/* Procedures for the treatement of a document           */
/*********************************************************/
affiche:
if corps=Ø then nop
else
do;
  if corps=2 then
  do
     titre.?gdoc = texte;
  end /* do corps 2 */
  else
  do
     nIspf = ligne.?gdoc.Ø + 1;
     ligne.?gdoc.Ø = nIspf;
     if centrer=Ø then
     do
        ligne.?gdoc.nIspf=texte;
        attribut.?gdoc.nIspf=l_att;
     end
     else
     do
        ligne.?gdoc.nIspf=center(texte,long_ligne);
        attribut.?gdoc.nIspf=center(l_att,long_ligne);
     end;
  end /* do */
end;
if tab<Ø then tab=Ø;
texte=substr(" ",1,tab);
l_att=substr(" ",1,tab);
return;

accent: procedure;
m=arg(1);
r="";
i=pos("&",m);
do while i>Ø;
   r=r||left(m,i-1);
   m=substr(m,i+1);
   j=pos(";",m);
   if j=Ø then r=r||"&";
   else
   do
      car=substr(m,1,j);
      m=substr(m,j+1);
      select
```

```
         when car="eacute;" then car="{";
         when car="egrave;" then car="}";
         when car="agrave;" then car="@";
         when car="ccedil;" then car="\";
         when car="ugrave;" then car="|";
         when car="lt;" then car="<";
         when car="gt;" then car=">";
         when car="Eacute;" then car="E";
         when car="Egrave;" then car="E";
         when car="Agrave;" then car="A";
         when car="Ugrave;" then car="U";
         when car="Ccedil;" then car="C";
         when car="acirc;" then car="É";
         when car="Acirc;" then car="A";
         when car="ecirc;" then car="à";
         when car="Ecirc;" then car="E";
         when car="Icirc;" then car="I";
         when car="icirc;" then car="å";
      otherwise
         car=".";
      end  /* select */
      r=r||car;
   end /* do */
   i=pos("&",m);
end /* do */
return r||m;

Ajout_mot:
w=arg(1); at=arg(2);
if at="" then at=attribut
lw=length(w);
a=substr(at,1,lw,at);
if length(texte)+lw>=long_ligne then
   call affiche Ø;
if length(texte)=Ø then
do
   texte=w;
   l_att=a;
end;
else;
do
   texte=texte w
   l_att=l_att a;
end
return;

/* procedure for calling the TCP/IP functions  */

tcpcall: procedure expose src debug;
  parse value  socket( ,
                       arg(1),arg(2),arg(3),arg(4),arg(5), ,
```

```
                      arg(6),arg(7),arg(8),arg(9),arg(10),
                    ) with src resultat;
  if debug="ALL" then
  do;
    say "TCPCALL: Fonction="arg(1) "Code retour="src;
    say "TCPCALL:                  Resultat="resultat;
  end;
return resultat;

/* Affichage sous ISPF */
Affiche_Document:
titre = titre.?doc;
numero = ?doc;
longueur = ligne.?doc.0;
debut=1;
do forever;
  fin=min(debut+taille-1,longueur);
  L1="";S1="";
  do i=debut to fin;
     L1=L1 || substr(ligne.?doc.i,1,80);
     S1=S1 || substr(attribut.?doc.i,1,80);
  end;
  choix="";
  address ispexec "DISPLAY PANEL(texplor)";
  code_Ispf=RC;
  if code_Ispf¬=0 then return code_Ispf;
  if PF="PF07" | PF="PF19" then
  do;
     debut=debut-taille-1;
     if debut<1 then debut=1;
  end;
  else
  if PF="PF08" | PF="PF20" then
  do;
     debut=debut+taille-1;
     if (debut+taille-1)>longueur then debut=longueur-taille+1;
     if debut<1 then debut=1;
  end;
  else
  if PF="PF10" | PF="PF22" then
  do;
     zz=?doc-1;
     if zz<1 then zz=1
     URL = URL.zz;
     return 0;
  end;
  else;
  if PF="PF11" | PF="PF23" then
  do;
     zz=?doc+1;
     if zz>?Ndoc then zz=?Ndoc
```

```
   URL = URL.zz;
   return Ø;
end;
else
if PF="PF12" | PF="PF24" then /* Home page */
do;
   zz=1;
   URL = URL.zz;
   return Ø;
end;
else
if choix="?" then /* liste des URL */
do;
   zz2=Ø;
   if ?Ndoc>6Ø then zz1=?Ndoc-59; else zz1=1
   do zz=1 to 6Ø;
      interpret "LISTED"zz "=' '";
   end;
   do zz=zz1 to ?Ndoc;
      zz2=zz2+1
      kkk= substr(zz,1,5) ":" substr(URL.zz,1,52);
      interpret "LISTED"zz2 "= kkk";
   end;
   address ISPEXEC "ADDPOP";
   address ISPEXEC "DISPLAY PANEL(TEXPLOR2)";
   zz3=RC
   address ISPEXEC "REMPOP";
   if zz3=Ø then
   do;
      if LISTESEL>Ø & LISTESEL<=?Ndoc then
      do;
         URL = URL.LISTESEL
         return Ø;
      end;
   end;
end;
else
if choix¬="" then
do;
   if datatype(choix)="NUM" then
   do;
      if choix>Ø & choix<= HRef.?doc.Ø then
      do;
         URL = HRef.?doc.choix;
         return Ø;
      end;
   end;
   else
   do;
      URL = choix;
      return Ø;
```

```
     end;
  end;
end;
```

TEXPLORO.PNL

```
)ATTR DEFAULT(%+_)
 # TYPE(INPUT) CAPS(OFF)
)BODY
%TSO Explorer
+
+Appel du document initial suivant:
%          #$CPURL
+
)INIT
IF (&$CPURL= ' ')
  &$CPURL='http://a691pwer.rdv.dcp'
)PROC
VPUT ($CPURL)
)END
```

TEXPLOR1.PNL

```
)ATTR DEFAULT(%+_)
 # TYPE(INPUT) CAPS(OFF)
)BODY
%TSO Explorer
+
+La r{f{rence suivante est invalide:
%          #URL
+
)END
```

TEXPLOR2.PNL

```
)ATTR DEFAULT(%+_)
 * AREA(SCRL) EXTEND(ON)
)BODY
%List of availible documents
%Your choice:_LISTESEL
+
*LISTE                                                    *
)AREA LISTE DEPTH(1Ø)
%&LISTED1                                                 +
%&LISTED2                                                 +
%&LISTED3                                                 +
%&LISTED4                                                 +
%&LISTED5                                                 +
```

```
%&LISTED6                                                    +
%&LISTED7                                                    +
%&LISTED8                                                    +
%&LISTED9                                                    +
%&LISTED1Ø                                                   +
%&LISTED11                                                   +
%&LISTED12                                                   +
%&LISTED13                                                   +
%&LISTED14                                                   +
%&LISTED15                                                   +
%&LISTED16                                                   +
%&LISTED17                                                   +
%&LISTED18                                                   +
%&LISTED19                                                   +
%&LISTED2Ø                                                   +
%&LISTED21                                                   +
%&LISTED22                                                   +
%&LISTED23                                                   +
%&LISTED24                                                   +
%&LISTED25                                                   +
%&LISTED26                                                   +
%&LISTED27                                                   +
%&LISTED28                                                   +
%&LISTED29                                                   +
%&LISTED3Ø                                                   +
%&LISTED41                                                   +
%&LISTED42                                                   +
%&LISTED43                                                   +
%&LISTED44                                                   +
%&LISTED45                                                   +
%&LISTED46                                                   +
%&LISTED47                                                   +
%&LISTED48                                                   +
%&LISTED49                                                   +
%&LISTED5Ø                                                   +
%&LISTED51                                                   +
%&LISTED52                                                   +
%&LISTED53                                                   +
%&LISTED54                                                   +
%&LISTED55                                                   +
%&LISTED56                                                   +
%&LISTED57                                                   +
%&LISTED58                                                   +
%&LISTED59                                                   +
%&LISTED6Ø                                                   +
)PROC
   VER(&LISTESEL,NUM)
)END
```

Claude Dunand (France)

TCP/SNA news

IBM has begun shipping Version 2.4 of Tivoli NetView Performance Monitor, which integrates with TME 10 NetView for OS/390 automation and management, and supports both SNA-based hardware and software and TCP/IP.

The integration with TME 10 NetView for OS/390 allows for a single view for managing system and network availability. Among the enhancements is a TCP/IP session collection capability which collects data for a complete VTAM through TCP/IP session. It's available for TN3270 and TN3270E servers running OS/390 Release 5 or higher.

Using the monitoring capability for Multinode Persistent Sessions (MNPS), VTAM can preserve sessions across application outages where hosts are connected through the OS/390 coupling facility. MNPS provides for the recovery of VTAM, MVS, or hardware failures.

There's also now a monitoring capability for Communications Storage Manager (CSM), part of high performance data transfer, which is designed to reduce performance overhead resulting from the movement or copying of large amounts of data.

Finally, the performance monitor works on RTP, which is used by high performance routing to perform end-to-end error recoveries and flow control. HPR allows SNA to make use of multi-megabit links with very low bit error rates on LANs and WANs, and to make use of switched virtual networking technologies like Frame Relay, SMDS, and ATM.

For further information contact your local IBM representative.

* * *

NetManage has begun shipping Version 8.0 of Chameleon HostLink, which includes ActiveX-based controls for 3270 and 5250 emulation, as well as NetManage's SupportNow real-time support technology.

SupportNow provides real-time, interactive support capabilities via the Internet, and is integrated into the applications provided within HostLink and Unix Link.

New features in HostLink 8.0 include SupportNow and ODBC access to AS/400 relational databases. The new ODBC drivers allow Windows users network access and information retrieval from the AS/400 databases. The ODBC inclusion provides a development foundation for Visual BASIC, Powerbuilder, and Visual C++ users as well as compatibility with ODBC-enabled desktop applications.

The drivers support APPC connectivity in addition to TCP/IP.

For further information contact:
NetManage, 10725 N De Anza Blvd, Cupertino, CA 95014, USA.
Tel: (408) 973 7171.
NetManage UK, Chameleon House, 22 Frederick Sanger Road, Guildford, Surrey, GU2 5YD, UK.
Tel: (01483) 302333.
URL: http://www.netmanage.com.

* * *

IBM has announced Communications Server for UnixWare 7, which puts SNA gateway, integrated TN3270E server, routing, and APPN support on Intel-based UnixWare boxes.

For further information contact your local IBM representative.

* * *