34

TCP/SNA update

June 1999

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 £130.00 in the UK; $190.00 in the USA and Canada; £136.00 in Europe; £142.00 in Australasia and Japan; and £140.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the March 1991 issue, are available separately to subscribers for £33.00 ($48.00) each including postage.

Printed in England.

SNA and TCP/IP integration

I remember people claiming that client/server was king and mainframes were dead as dinosaurs. If this were true, how come IBM is currently spending $100m (£63m) a year on SNA/IP integration development? And why are there still some 50,000 SNA licences worldwide?

IBM suggests that 40 percent of its mainframes will soon support IP natively. So what's available to those who want to integrate SNA with other network traffic?

There are multiple ways to integrate SNA traffic across IP networks via routers, gateways, and software-based emulation. In connectivity terms, SNA's big problem is that it is not natively routable. It is better over WANs, particularly over Frame Relay, because SNA is something of a WAN protocol in its own right.

IBM's current high-end routers are the 2216 Nways Multi-access Connector and 2210 Nways Multiprotocol router. The 2216 provides IP, APPN (Advanced Peer-to-Peer Networking), SNA, and HPR (High-Performance Routing) network connectivity to ESCON-attached System/390 Parallel Enterprise Servers and offers improved performance, including Fast Ethernet, FDDI, high-speed ATM, and User Diagram Protocol (UDP) support.

FRAME RELAY

Traditionally, IBM host networks connected users via SNA or BSC multidrop lines – low-speed analogue lines. This resulted in sites running a large number of leased lines. IBM introduced X.25 as a cost-effective alternative to complex matrices of private lines. Not every SNA site implemented it, however, because user-response times were much slower. This was because of the network overhead delays caused by every node in the X.25 network performing error detection/correction, message sequencing, and flow control. Frame Relay, however, is much better because it performs these functions only at the network access points using an end-to-end protocol.

Frame Relay replaced SDLC and BSC multidrop lines when SNA networks were upgraded to packet-based networks in the mid-1990s. Replacing costly and unwieldy leased line networks, Frame Relay has the advantage of providing network services as an integral part of IBM's overall wide-area strategy for integrating SNA.

ENCAPSULATION

With connection-oriented protocols such as SNA, a path is set up through the network (a session) before any data is sent. With connectionless-oriented protocols, such as TCP/IP or Novell IPX, packets progress node by node.

There are three main encapsulation technologies available:

- Gateways, which put SNA messages into packets of routable protocols (for example, TCP/IP, IPX).
- Data Link Switching (DLSw), which enables routers to put SNA messages into IP packets.
- Dependent Logical Unit Requester/Server (DLUR/S), which carries messages over APPN.

SNA routing is performed at Level 2 of the OSI model (the data link layer) rather than Level 3 (the network layer).

IBM has evolved SNA in two stages. The first was a belated attempt to overcome SNA's host-centric design by introducing peer-to-peer capabilities in the form of APPN. The second was Data Link Switching.

HPR

APPN transferred controlling intelligence from the host to within the network itself. Sessions and routes between devices are determined according to the status of the network, without reference to the host. APPN formed the base for the third phase of SNA development, HPR.

HPR is supported across a range of IBM communications products. It removed a previous disadvantage of SNA, which stems from the fact that it is a connection-oriented service. APPN initially only supported LU6.2 sessions and not terminal emulation. IBM's Dependent LU

Service Requester provided support for mainframe and 3270 applications. Most router vendors support APPN, and build DLUR/S into routing software. APPN also reduces administration and provides higher bandwidth availability by getting rid of scheduled downtime. It offers a single management view, if that is the direction the enterprise IT ethos is driving towards.

APPN lacked any traffic prioritization schemes. HPR software provides error checking and flow control (although at session endpoints only). It also ends session loss because of path failure, even when SNA traffic is co-resident on IP links.

With HPR, though, it's important to watch traffic patterns to ensure that SNA sessions get sufficient bandwidth if more than 40% of the traffic running across the same lines is IP based. Disconnection can result from IP's tendency to go to the front of the queue for each connection.

DLSw

DLSw allows SNA traffic to be routed over TCP/IP networks, using compliant routers. A process called encapsulation, effectively hiding the SNA protocol code in an envelope of TCP/IP code, does the routing and ensures that SNA sessions do not time out when congestion occurs.

DLSw tunnels both SNA and NetBIOS traffic and also offers quality of service via NetBIOS cacheing and flow control protocols. More router vendors are offering DLSw options for existing and new routers. But DLSw does reduce performance because of the encapsulation process, and is therefore best suited to sites with low volumes of SNA traffic. It avoids the need to mix IP and SNA across an underlying switched ATM or Frame Relay transmission network, but there are still two network layer protocol stacks to manage end-to-end between the network and the host.

SEPARATE TRAFFIC

Where networks do not include a high percentage of SNA or where integration strategies are still being determined. there may be a need to separate SNA traffic from IP and IPX traffic. Although this is possible, it can be costly. The SNA gateway market includes Internet-to-host gateways and TN3270 servers. The advent of browser-led access was one of the biggest upsets in the SNA market.

Web integrators, written in Java and running inside the browser, allow cheap PC desktop access to SNA applications running on the mainframe. Vendors such as Attachmate and IBM bundle gateway options into one package.

An advantage of multiprotocol gateways is that they can be custom-configured to support limited SNA mission-critical applications for further TCP/IP migration. Gateways also screen-off SNA resources from busy IP links. Here, SNA is kept in the data centre, not run through routers, enabling IP applications to access SNA databases or applications only on an as-needed basis.

ANYNET

Another SNA replacement from IBM, AnyNet, was designed to cut the number of protocols managed at a site to one or two. It uses some sophisticated masking technologies designed to support IP, IPX, NetBIOS, and SNA from point-to-point, anywhere on the network.

AnyNet is based on IBM's formerly proprietary Multiprotocol Transport Networking (MPTN) technology and designed to make all enterprise applications independent from their transport so that upper-layer protocols can use non-native transports. This would make it possible, for instance, to map SNA APPC applications to the TCP/IP transport.

It does have major drawbacks, however. For AnyNet to be effective, SNA networks engineered for low-traffic 3270 sessions will probably need increased WAN bandwidth to handle other protocols for LAN-to-LAN use, especially IP. Also, AnyNet software must reside at all nodes.

TN3270

There is another way to integrate SNA with IP without using encapsulation. The TN3270 protocol eliminates the SNA loop between the client and the router at the edge of the network. TN3270 still emulates 3270 or 5250 screens, but, unlike DLSw, these screens feed directly into TCP/IP sessions, with no SNA involved at the client end. A server or gateway translates the IP sessions into SNA ones, obviating encapsulation and the need to run SNA protocol stacks in clients.

J Phillips
Independent Consultant (UK)

Centralized network console – part 3

This month we conclude the code to centralize at a specific point the management of different systems.

CLIST NCCXC00

```
/* NCCXCØØ */
/*********************************************************************/
/* CLIST Main.                                                       */
/* Called by NetView operators.                                      */
/* It is the interface used to visualize in a centralized            */
/* way MVS events only.                                              */
/* It manages global variables.                                      */
/* It calls CLIST NCCXCØ1 for the visualization of the details of    */
/* the event selected.                                               */
/*********************************************************************/
trace ?e
fillb = copies(' ',7Ø)
SIGNAL ON HALT
SIGNAL ON ERROR
'UNIQUE PROMOTE'
$command = 'FI UY'
$nccvgØ1 = 'FA IH CT HR UN'
$nccvgØ2 = 'FA IH CT HR UN'
$nccvgØ3 = 'FA IH CT HR UN'
$nccvgØ4 = 'FA IH CT HR UN'
$nccvgØ5 = 'FA IH CT HR UN'
$nccvgØ6 = 'FA IH CT HR UN'
$nccvgØ7 = 'FA IH CT HR UN'
$nccvgØ8 = 'FA IH CT HR UN'
$nccvgØ9 = 'FA IH CT HR UN'
```

```
$nccvg1Ø = 'FA IH CT HR UN'
$nccvg11 = 'FA IH CT HR UN'
$nccvg12 = 'FA IH CT HR UN'
$nccvg13 = 'FA IH CT HR UN'
$nccvg14 = 'FA IH CT HR UN'
'GLOBALV GETT $nccvgØ1,$nccvgØ2,$nccvgØ3,$nccvgØ4,$nccvgØ5,$nccvgØ6,',
     '$nccvgØ7,$nccvgØ8,$nccvgØ9,$nccvg1Ø,$nccvg11,$nccvg12,$nccvg13'
'GLOBALV GETT nccvgØ1,nccvgØ2,nccvgØ3,nccvgØ4,nccvgØ5,nccvgØ6,nccvgØ7',
     'nccvgØ8,nccvgØ9,nccvg1Ø,nccvg11,nccvg12,nccvg13,opncc,ncctxt'
Entry_point:
DO FOREVER
  command = 'ØØ'X
  ncctxtb = copies(' ',7Ø)
  'globalv putt ncctxt'
  'VIEW NCCXØØØ NCCXPØØ INPUT MSG'
  UPPER command
  SELECT
    when viewaid = PF1 then view 8 NCCXPØØH
    when viewaid = PF2 then exit
    when viewaid = PF6 then 'CMD HIGH ROLL'
    when viewaid = PF8 then call init_menu
    when viewaid = ENTER then
        SELECT
          when command = NEXT | command = 'ØØ'X then call init_menu
          when command ¬= ' ' then
               DO
               'CMD HIGH' COMMAND
               END
          otherwise nop
        END
    otherwise nop
  END
END
init_menu:
$command = 'FI UY'
ncctxtb = copies(' ',7Ø)
$ncctxtb = 'FA IH CR HD UN'
$opncc = 'FI IH CG HR UN'
DO FOREVER
 command = 'ØØ'X
'GLOBALV GETT $nccvgØ1,$nccvgØ2,$nccvgØ3,$nccvgØ4,$nccvgØ5,$nccvgØ6,',
     '$nccvgØ7,$nccvgØ8,$nccvgØ9,$nccvg1Ø,$nccvg11,$nccvg12,$nccvg13'
'GLOBALV GETT nccvgØ1,nccvgØ2,nccvgØ3,nccvgØ4,nccvgØ5,nccvgØ6,nccvgØ7',
     'nccvgØ8,nccvgØ9,nccvg1Ø,nccvg11,nccvg12,nccvg13,opncc,ncctxt'
  'VIEW NCCXØØ1 NCCXPA1 INPUT MSG'
  UPPER command
  SELECT
    when opncc = Ø1 then call NCCXCØ1 fillb,opncc
    when opncc = Ø2 then call NCCXCØ1 fillb,opncc
    when opncc = Ø3 then call NCCXCØ1 fillb,opncc
    when opncc = Ø4 then call NCCXCØ1 fillb,opncc
    when opncc = Ø5 then call NCCXCØ1 fillb,opncc
    when opncc = Ø6 then call NCCXCØ1 fillb,opncc
    when opncc = Ø7 then call NCCXCØ1 fillb,opncc
    when opncc = Ø8 then call NCCXCØ1 fillb,opncc
    when opncc = Ø9 then call NCCXCØ1 fillb,opncc
    when opncc = 1Ø then call NCCXCØ1 fillb,opncc
    when opncc = 11 then call NCCXCØ1 fillb,opncc
    when opncc = 12 then call NCCXCØ1 fillb,opncc
    when opncc = 13 then call NCCXCØ1 fillb,opncc
    when viewaid = PF1 then do
                             view 8 NCCXPA1H
       if ncctxtb ¬= fillb then
                              do
                               ncctxtb = fillb
                               'globalv putt ncctxtb'
                              end
       end
    when viewaid = PF2 then exit
    when viewaid = PF7 then return
    when viewaid = PF6 then CMD HIGH ROLL
    when viewaid = PF11 then signal Entry_point
    when viewaid = ENTER then
        SELECT
          when command = BACK then return
          when command ¬= ' ' then
               DO
                ncctxtb = 'Command/option not foreseen or wrong.'
                'globalv putt ncctxtb'
               END
          otherwise nop
        END
    otherwise nop
  END
END
ERROR:
retc = rc
if retc = 4 then do
          say time() '*************************************************'
          say time() '***                                           ***'
          say time() '*** NCCXCØØ  - NetView Centralized Console.   ***'
          say time() '***         Procedure already active. Close.  ***'
          say time() '***                                           ***'
          say time() '*************************************************'
          end
          else do
          say time() '*************************************************'
          say time() '***                                           ***'
          say time() '*** NCCXCØØ  - NetView Centralized Console.   ***'
          say time() '***            Procedure in error.  RC 'retc' ***'
          say time() '***                                           ***'
          say time() '*************************************************'
          end
 EXIT -1
HALT:
retc = rc
say time() '*****************************************************'
say time() '***                                               ***'
say time() '*** NCCXCØØ  - NetView Centralized Console.       ***'
say time() '***            Procedure cancelled. RC 'retc'     ***'
say time() '***                                               ***'
say time() '***                                               ***'
say time() '*****************************************************'
EXIT
```

Panel NCCXP00

```
/*
*** AT2
+NCCXPØØ%                     System Support - Napoli
+                                 ? MVS - SYSS +
+
$
+                 +OPERATOR ID =$&OPID    +APPLICATION =$&APPLID
+
+
\          ======  ==   ==  ======   ======   ======  ======   ======
\          ==      ==  \     NETVIEW  CENTRALIZED CONSOLE      \ ==
\          ==      ==   ==  ==   ==  ==   ==  ==  ==  ==   ==    ==
\          ======  ==   ==  =====    =====    ==  ==  ======     ==
\              ==  ==   ==  ==       ==       ==  ==  == ==      ==
\              ==  ==   ==  ==       ==       ==  ==  ==  ==     ==
\          ======  =======  ==       ==       ======  ==   ==    ==
\'98$
$
$
$
$
$
$
\
%CMD==> &COMMAND
$            PF1=Help    PF2=Fine
$            PF6=Roll                        PF8/ENTER=Continua
```

Panel NCCXP00H

```
/*
***
```

```
+NCCXPØØH%
+                                \H E L P+
+             \ MANAGEMENT PROCEDURE FOR CENTRALIZED CONSOLE MVS.  +
+
+
+     This procedure manages in a centralized way the events of an MVS
+     system. It represents the operating interface towards all the
+     predefined events.
+     It is a focal point and can be considered as an on-line guide for
+     the management of the console events.
+
+
+
+
+
+
+
+
+
+
+
+
+
$             PF2/PF3=Exit Help
$    PF6=Roll  PF7=Previous
```

Panel NCCXPA1

```
HELP=NCCXPA1H
*** AT2
+NCCXPA1%          N E T V I E W  *@Procedure CENTRALIZED CONSOLE%*
+
+              } Events managed in MVS environment of development. +
+
+      %1? IEA911E         + %2? IEC512I         + %3? IEF233A       +
+            &NCCVGØ1   %           &NCCVGØ2   %           &NCCVGØ3   %
+      %4? IEF238D         + %5? IEF45ØI         + %6? IEF455D       +
+            &NCCVGØ4   %           &NCCVGØ5   %           &NCCVGØ6   %
+      %7? IOSØØØI         + %8? IOSØØ3A         + %9? IST663I       +
+            &NCCVGØ7   %           &NCCVGØ8   %           &NCCVGØ9   %
+     %1Ø? HASP19Ø         +%11? ESF499          +%12? SLSØ665E      +
+            &NCCVG1Ø   %           &NCCVG11   %           &NCCVG12   %
+     %13? IST1Ø5I         +
+            &NCCVG13   %
+
$ Choice option for verification of detail : &OPNCC%
+
+ Last Event:
+ &NCCTXT
$
```

```
\ &NCCTXTB
%
%Action==> &COMMAND                                               %
$         PF1=Help   PF2=Exit
$         PF6=Roll   PF7=Previous               PF11=Entry Point
```

Panel NCCXPA1H

```
/*
***
+NCCXPA1H%
+                                  \H E L P+
+               \ PROCEDURE CENTRALIZED CONSOLE MVS Development.  +
+
+
+
+
+ From this panel it is possible to select an event in order to consult
+ the details and to execute the suggested actions.
+ The panel is automatically updated when verifying one of the
+ predefined events.
+ The updates are carried out for all the NetView operators connected.
+
+
+
+
+
+
+
+
+
+
$              PF2/PF3=Exit Help
$    PF6=Roll  PF7=Previous
```

CLIST NCCXC01

```
/* NCCXCØ1 */
/**********************************************************************/
/* Called by select options from CLIST main NCCXCØØ.                  */
/* It manages global variables.                                       */
/* It visualizes the details for every event and supplies one line    */
/* guides for their management.                                       */
/**********************************************************************/
trace ?e
Parse Arg fillb,opncc
if opncc <= 9 then opncc = 'Ø'opncc
ncctxtb = fillb
'globalv putt ncctxtb'
```

```
/*******************************************************************/
/* Update one variable of the selected event in order to           */
/* show all the operators that the event has been                  */
/*  managed.                                                       */
/*******************************************************************/
rvgb = 'NCCVG'||opncc
c1= '$'
interpret rvgb "='Selected'"
interpret c1||rvgb "='IH CG HR'"
interpret "'GLOBALV PUTT 'rvgb',$'rvgb"
interpret "'GLOBALV GETT 'rvgb',$'rvgb"
env = 'S'
ncctxt = 'Selected event gives 'opid()
/*******************************************************************/
/* Alignment for all the operators.                                */
/*******************************************************************/
Call NCCSC3Ø rvgb env ncctxt
one_menu:
DO FOREVER
 command = 'ØØ'X
$command = 'FI UN'
  'VIEW NCCX'opncc' NCCXP'opncc' INPUT MSG'
  UPPER command
  SELECT
    when viewaid = PF2 | viewaid = PF7 then
                   DO
                    if ncctxtb ¬= fillb then
                              do
                               ncctxtb = fillb
                               'globalv putt ncctxtb'
                              end
                    return
                   END
    when viewaid = PF6 then CMD HIGH ROLL
    when viewaid = ENTER then do
        SELECT
/*******************************************************************/
/* From this point it is possible to connect to TSO and/or CICS    */
/* IMS.                                                            */
/*******************************************************************/
          when command = BACK then return
          when substr(command,1,7) = RAMMTSO Then do
                                     appltso = substr(command,9,8)
                                     BFSESS appltso
                                     say time() ' NCC - Logoff from TSO
...'
                                      End
          when substr(command,1,8) = RAMMCICS Then do
                                    applcix = substr(command,1Ø,8)
                                    BFSESS applcix
                                    say time() ' NCC - Logoff from CICS
...'
                                      End
          when substr(command,1,7) = RAMMIMS Then do
                                     applims = substr(command,9,8)
                                     BFSESS applims
                                     say time() ' NCC - Logoff from IMS
...'
                                      End
          when opncc = 'Ø7' then view 8 NCCXPØ7B
          when opncc = '12' then view 8 NCCXP12B
          when opncc = '13' then view 8 NCCXP13B
          otherwise nop
         end
        END
    otherwise nop
  END
END
return
```

From this environment it is possible to recall a panel for every event managed to see all the technical details. Some examples are shown below.

Panel NCCXP01

```
/*
*** AT2
+NCCXPØ1                  \ System Support - Napoli +
$                        ¬IEA911E : Dump in SYS1.DUMPxx$
% Description
$ Message IEA911E indicates the production of a dump in the
$ SYS1.DUMPxx datasets
$ This event previews the automatic execution of procedure S1DMPXD,
$ then copies the
$ dump dataset to a temporary archive for eventual analysis and
$ produces a report with a summarized description of the dump.
$
% Causes
$ Error in a component of the system (batch, stc, subsystem, product).
$
% Actions
$ 1. To be sure that procedure S1DMPXD is finished.
$ 2. To execute from console the command D D,ST in order to verify the
$    state of the dump datasets.
$ 3. To execute from console cmd DD CLEAR,DSN=ALL in order to empty the
$    dump datasets.
$
% Contacts
```

```
$ In difficult cases contact system support by e-mail.
$
%CMD==> &COMMAND
$                    PF2/PF7=Return
$         PF6=Roll           comando¬RAMMTSO appl-tso$= sessione con TSO
```

Panel NCCXP13

```
/*
*** AT2
+NCCXP13                  \ System Support - Napoli +
$
$                         ¬IST1Ø5I : Nodo VTAM inactive$
$
% Description
$ The VTAM node specified in the message is not active.
$ The event manages the connections of type SNI (SNA Network
$ Interconnection) with the networks of other companies.
$
% Causes
$ The connection line TD is down or the node CROSS DOMAIN has
$ been inactivated.
$
% Actions
$ 1. To verify the state of the node specified in the message with the
$    command  D NET, ID=node-name,E.
$ 2. If the state of node is INOP or NEVAC to execute VTAM command
$             V NET,ACT,ID=node-name
$
$
$
%CMD==> &COMMAND
$                    PF2/PF7=Return          Enter=Continuous
$       PF6=Roll             command¬RAMMTSO appl-tso$= session with TSO
```

Panel NCCXP13B

```
/*
*** AT2
+NCCXP13B                 \ System Support - Napoli +
$
$                         ¬IST1Ø5I : Nodo VTAM inattivo$
$
$
% Actions
$ 3. If the state of node is PCTD1 or PCTD2 contact the Help Desk for
$    that network.
$
$ Currently our network SNA is interconnected with ......
```

```
$
$
$
% Contacts
$ In difficult cases contact system support by e-mail.
$
$
$
$
$
$                    PF2/PF7=Return
$         PF6=Roll
```

Panel NCCXP07

```
/*
*** AT2
+NCCXPØ7                  \ System Support - Napoli +
$
$                      ¬IOSØØØI : Unit in hardware error$
$
% Description
$ The I/O Supervisor component of the operating system finds an error
$ hardware for the device specified in the message.
$
% Causes
$ The causes that produce the error are multiple. For the complete list
$ consult the handbook '"MVS Messages and Code"'.
$ The more frequent causes are:
$
$ - DCK  data check on tape or disk
$ - CDC  channel error
$ - IOE  error of I/O on tape or disk
$ - WRI  writing on inhibited unit
$ - EOT  it has been caught at the end of the tape
$ - EQC  unit in error
$ - NCA  tape not initialized
$
%CMD==> &COMMAND
$                    PF2/PF7=Return          Enter=Continuous
$       PF6=Roll             command¬RAMMTSO appl-tso$= session with TSO
```

Panel NCCXP07

```
/*
*** AT2
+NCCXPØ7B                 \ System Support - Napoli +
$
$                      ¬IOSØØØI : Unit in hardware error$
```

```
$
$
% Actions
$ For error WRI to verify that the units are qualified in reading/
$  writing.
$ For error NCA to verify that the tape is initialized.
$ For all the other errors consult SYS1.LOGREC or contact the
$ support staff.
$

% Contacts
$ In difficult cases contact system support by e-mail.
$
$
$
$
$
$                    PF2/PF7=Return
$         PF6=Roll
```

Example CLIST to activate the centralized console procedure

```
/* IST1Ø5I */
Trace ?o
/*__________________________________________________________________
  Management Events with NetView Centralized Console procedure.
  Called by msg IST1Ø5I from message automation table.
  ______________________________________________________________e.m.*/
Arg message
node = word(message,2)
VtmTab.1 = 'L15Ø32S   TIM       '
VtmTab.2 = 'PUSNI5    TIM       '
VtmTab.3 = 'CDRM38    TIM       '
VtmTab.4 = 'L1514ØS   TELECOM   '
VtmTab.5 = 'PUSNI3    TELECOM   '
VtmTab.6 = 'CDRM12    TELECOM   '
VtmTab.7 = 'L15133S   NETSIEL   '
VtmTab.8 = 'PUSNIA    NETSIEL   '
VtmTab.9 = 'CDRMC     NETSIEL   '
VtmTab.1Ø= 'L15134S   INTESA    '
VtmTab.11= 'PUSNI$1   INTESA    '
VtmTab.12= 'ITINTEØA  INTESA    '
VtmTab.13= 'L15128S   SISTEMII  '
VtmTab.14= 'PUSNIC    SISTEMII  '
VtmTab.15= 'SSCPSI    SISTEMII  '
do i=1 to 15
 nodet = word(VtmTab.i,1)
 if nodet = node then do
                       nconn = word(VtmTab.i,2)
                       call NCC_proc
                       leave
                      end
                 else iterate
End
Exit
NCC_proc:
message = 'Attention inactive connection with 'nconn' for the resource
'node'.'
codmsg = 'IST105I'
Address Netview
"Excmd NCCOPR,NCCSC00 "codmsg message
say time() '*******************************************************'
say time() '***                                                  ***'
say time() '*** IST105I  -  NetView Centralized Console.         ***'
say time() '***             Management events: RSH running.      ***'
say time() '***                                                  ***'
say time() '*******************************************************'
Return


/* IEA911E  */
Trace ?o
/*
_________________________________________________________________
  Management Events with NetView Centralized Console procedure.
  Called by msg IEA911E from message automation table.
_____________________________________________________________e.m.*/
Arg message
tdump = word(message,2)
ddump = word(message,5)
dmpx = substr(ddump,6,6)
"MVS S S1DMPXD,SIST=S,DUMP="dmpx
say time() '*******************************************************'
say time() '***                                                  ***'
say time() '*** IEA911E -  DUMP DATASET(S) OF SYSTEM IS FULL.    ***'
say time() '***                                                  ***'
say time() '***            EXEC job S1DMPXD for 'dmpx'           ***'
say time() '***                                                  ***'
say time() '*******************************************************'
message = 'ATTENTION: 'tdump' Dump of 'ddump'.'
codmsg = 'IEA911E'
Address Netview
"Excmd NCCOPR,NCCSC00 "codmsg message
say time() '*******************************************************'
say time() '***                                                  ***'
say time() '*** IEA911E  -  NetView Centralized Console.         ***'
say time() '***             Management Events: RSH running.      ***'
say time() '***                                                  ***'
say time() '*******************************************************'
Exit
```

```
/* IOSØØØI */

Trace ?o
/*
_________________________________________________________________
  Management Events with NetView Centralized Console procedure.
  Called by msg IOSØØØI from message automation table.
_____________________________________________________________e.m.*/
Arg message
mtrans = translate(message,' ',',')
ndevice = word(mtrans,2)
terr = word(mtrans,4)
ap="'"
message = 'Hardware Error of type 'ap||terr||ap' for the device
'ndevice'.'
codmsg = 'IOSØØØI'
Address Netview
"Excmd NCCOPR,NCCSCØØ "codmsg message
say time() '*********************************************************'
say time() '***                                                   ***'
say time() '*** IOSØØØI  -  NetView Centralized Console.          ***'
say time() '***             Management Events: RSH running.       ***'
say time() '***                                                   ***'
say time() '*********************************************************'
Exit
```

Sample file of qualified table operators

```
BROWSE -- NCC.SAØ1.UTL.OPR ------------------------- LINE ØØØØØØØØ COL
ØØ1 Ø8Ø
 COMMAND ===>                                            SCROLL ===> CSR
***************************** TOP OF DATA ******************************
  NCCOPR
_ ZZNS611
_ OPER1
_ NETOP
_ OPRCED
**************************** BOTTOM OF DATA ****************************
```

Sample MVS file for table of events

```
BROWSE -- NCC.SAØ1.UTL.TAB ------------------- LINE ØØØØØØØØ COL ØØ1 Ø8Ø
 COMMAND ===>                                           SCROLL ===> CSR
***************************** TOP OF DATA ******************************
 MVS Events   File on workstation   Global Variables

_ IEA911E         1CRØØ1.TXT         NCCVGØ1
_ IEC512I         1MAØØ2.TXT         NCCVGØ2
```

```
_ IEF233A          1MAØØ3.TXT          NCCVGØ3
_ IEF238D          1MAØØ4.TXT          NCCVGØ4
_ IEF45ØI          1CLØØ5.TXT          NCCVGØ5
_ IEF455D          1CRØØ6.TXT          NCCVGØ6
_ IOSØØØI          1CRØØ7.TXT          NCCVGØ7
_ IOSØØ3A          1MAØØ8.TXT          NCCVGØ8
_ IST663I          1CRØØ9.TXT          NCCVGØ9
_ $HASP19Ø         1MIØ1Ø.TXT          NCCVG1Ø
_ §ESF499          1WAØ11.TXT          NCCVG11
_ SLSØ665E         1CRØ12.TXT          NCCVG12
_ IST1Ø5I          1CRØ13.TXT          NCCVG13
**************************** BOTTOM OF DATA ****************************
```

Sample MVS file for Remote Shell script

```
BROWSE -- TEMP.NCCS.UTILITY.RSH.TØ13944 ------ LINE ØØØØØØØØ COL ØØ1 Ø8Ø
 COMMAND ===>                                            SCROLL ===> CSR
***************************** TOP OF DATA ******************************
$e remote shell to execute on workstation.
$e Sviluppo=1
$copy 1CRØ13.TXT lta1:
$exit
**************************** BOTTOM OF DATA ****************************
```

Sample MVS file with text for Remote Shell script

```
BROWSE -- TEMP.NCCS.UTILITY.TXT.TØ13944 ------ LINE ØØØØØØØØ COL ØØ1 Ø8Ø
 COMMAND ===>                                           SCROLL ===> CSR
***************************** TOP OF DATA ******************************
IST1Ø5I - ATTENTION INACTIVE CONNECTION WITH INTESA FOR THE RESOURCE
ITINTEØA.
**************************** BOTTOM OF DATA ****************************
```

```
Sample log of the activities
NCCOPR  IST1Ø5I  ITINTEØA NODE NOW INACTIVE
NCCOPR  DSI268I EXCMD COMPLETE
NCCOPR  15:Ø9:Ø4 ******************************************************
NCCOPR  15:Ø9:Ø4 ***                                                ***
NCCOPR  15:Ø9:Ø4 *** IST1Ø5I  -  NetView Centralized Console.       ***
NCCOPR  15:Ø9:Ø4 ***             Management Events: RSH running.    ***
NCCOPR  15:Ø9:Ø4 ***                                                ***
NCCOPR  15:Ø9:Ø4 ******************************************************
NCCOPR  NCCSCØØ IST1Ø5I ATTENTION INACTIVE CONNECTION WITH INTESA FOR
THE RESOURCE ITINTEØA
NCCOPR   DSI372I FREE COMMAND FAILED, 'NCCTXT' IS NOT ALLOCATED
NCCOPR   DSI372I FREE COMMAND FAILED, 'NCCRSH' IS NOT ALLOCATED
NCCOPR   DSI372I FREE COMMAND FAILED, 'NCCJOB' IS NOT ALLOCATED
NCCOPR   DSI372I FREE COMMAND FAILED, 'NCCTAB' IS NOT ALLOCATED
NCCOPR   CNM272I NCCJOB IS NOW ALLOCATED
NCCOPR   CNM272I NCCJOB IS NOW DEALLOCATED
NCCOPR   CNM272I NCCTXT IS NOW ALLOCATED
NCCOPR   CNM272I NCCTXT IS NOW DEALLOCATED
NCCOPR   CNM272I NCCRSH IS NOW ALLOCATED
NCCOPR   CNM272I NCCRSH IS NOW DEALLOCATED
NCCOPR   CNM272I NCCJOB IS NOW ALLOCATED
NCCOPR   CNM272I NCCTXT IS NOW ALLOCATED
NCCOPR   CNM272I NCCRSH IS NOW ALLOCATED
NCCOPR   CNM272I NCCTAB IS NOW ALLOCATED
NCCOPR   15:Ø9:Ø6  NCC - Read table NCC.SAØ1.UTL.TAB ...
NCCOPR   15:Ø9:Ø6  NCC - Interface creation for NCC-Server ...
NCCOPR   CNM272I NCCTXT IS NOW DEALLOCATED
NCCOPR   CNM272I NCCRSH IS NOW DEALLOCATED
NCCOPR   15:Ø9:Ø6  NCC - Creation TEMP.NCCS.UTILITY.JOB.TØ13944 ...
NCCOPR   CNM272I NCCJOB IS NOW DEALLOCATED
NCCOPR   CNM279I ZZNS611£(JOBØ711Ø) SUBMITTED
NCCOPR   15:Ø9:Ø6  NCC - Submit Job-Utility-NCC ...
NCCOPR   15:Ø9:Ø6  NCC - Jobname  ZZNS611£ ...
NCCOPR   15:Ø9:Ø6  NCC - Jobnumber  JOBØ711Ø ...
NCCOPR   CNM272I SKE IS NOW ALLOCATED
NCCOPR   CNM272I SKE IS NOW DEALLOCATED
NCCOPR   15:Ø9:Ø7  NCC -  RSH in execution.  Remote Shell id £Ø13944
NCCOPR   DSI372I FREE COMMAND FAILED, 'NCCOPR' IS NOT ALLOCATED
NCCOPR   CNM272I NCCOPR IS NOW ALLOCATED
NCCOPR   15:Ø9:Ø7  NCC - Reading operator table NCC.SAØ1.UTL.OPR ...
NCCOPR   15:Ø9:Ø7  NCC - Function refresh in progress ...
NCCOPR   STATION: ZZNS611  TERM: SYSSØ223
NCCOPR   HCOPY: NOT ACTIVE PROFILE: SYSPROF
NCCOPR   STATUS: ACTIVE
NCCOPR   DOMAIN LIST: NONE
NCCOPR   ACTIVE SPAN LIST: NONE
NCCOPR   END OF STATUS DISPLAY
NCCOPR   NCCXCVGI NCCVG13 $NCCVG13 RSH IN EXECUTION.  REMOTE SHELL ID
£Ø13944
NCCOPR   DSI268I EXCMD COMPLETE
NCCOPR   15:Ø9:Ø7 ******************************************************
NCCOPR   15:Ø9:Ø7 ***                                                ***
NCCOPR   15:Ø9:Ø7 *** NCCSC3Ø  -  NetView Centralized Console.       ***
NCCOPR   15:Ø9:Ø7 ***             Operator ZZNS611 logged-on.        ***
NCCOPR   15:Ø9:Ø7 ***             Refresh function active.           ***
NCCOPR   15:Ø9:Ø7 ***                                                ***
NCCOPR   15:Ø9:Ø7 ******************************************************
NCCOPR   15:Ø9:Ø7  NCC - Refresh function in progress ...
NCCOPR   DSIØØ8I 'OPER1' NOT ACTIVE
NCCOPR   15:Ø9:Ø7 ******************************************************
NCCOPR   15:Ø9:Ø7 ***                                                ***
NCCOPR   15:Ø9:Ø7 *** NCCSC3Ø  -  NetView Centralized Console.       ***
NCCOPR   15:Ø9:Ø7 ***             Operator OPER1 not logged-on.      ***
NCCOPR   15:Ø9:Ø7 ***             Refresh function stopped.          ***
NCCOPR   15:Ø9:Ø7 ***                                                ***
NCCOPR   15:Ø9:Ø7 ******************************************************
NCCOPR   DSI372I FREE COMMAND FAILED, 'LLOG1' IS NOT ALLOCATED
NCCOPR   CNM272I LLOG1 IS NOW ALLOCATED
NCCOPR   CNM272I LLOG1 IS NOW DEALLOCATED
NCCOPR   15:Ø9:Ø7 *****************************************************
NCCOPR   15:Ø9:Ø7 ***                                               ***
NCCOPR   15:Ø9:Ø7 *** NCCSCØØ  -  NetView Centralized Console.      ***
NCCOPR   15:Ø9:Ø7 ***             Update Event Log carried out.     ***
NCCOPR   15:Ø9:Ø7 ***                                               ***
NCCOPR   15:Ø9:Ø7 *****************************************************
NCCOPR   EXCMD AUTO1,AFTER Ø1,NCCXDEL TEMP.NCCS.UTILITY.TXT.TØ13944
NCCOPR   AFTER Ø1,NCCXDEL TEMP.NCCS.UTILITY.TXT.TØ13944
NCCOPR   DSI268I EXCMD COMPLETE
NCCOPR   DSIØ34I COMMAND SCHEDULED BY AT/EVERY/AFTER COMMAND - 'NCCXDEL
TEMP.NCCS.UTILITY.TXT.TØ13944'
NCCOPR   EXCMD AUTO1,AFTER Ø2,NCCXDEL TEMP.NCCS.UTILITY.RSH.TØ13944
NCCOPR   DSI268I EXCMD COMPLETE
NCCOPR   DSI2Ø1I TIMER REQUEST SCHEDULED FOR EXECUTION 'ID=SYSØ3Ø76'
NCCOPR   AFTER Ø2,NCCXDEL TEMP.NCCS.UTILITY.RSH.TØ13944
NCCOPR   DSIØ34I COMMAND SCHEDULED BY AT/EVERY/AFTER COMMAND - 'NCCXDEL
TEMP.NCCS.UTILITY.RSH.TØ13944'
NCCOPR   DSI2Ø1I TIMER REQUEST SCHEDULED FOR EXECUTION 'ID=SYSØ3Ø77'
NCCOPR   DSI2Ø8I TIME EXPIRATION - ID= 'SYSØ3Ø76' - CMD= 'NCCXDEL
TEMP.NCCS.UTI LITY.TXT.TØ13944'
NCCOPR   NCCXDEL TEMP.NCCS.UTILITY.TXT.TØ13944
NCCOPR   DSI372I FREE COMMAND FAILED, 'NCCDWRK' IS NOT ALLOCATED
NCCOPR   CNM272I NCCDWRK IS NOW ALLOCATED
NCCOPR   CNM272I NCCDWRK IS NOW DEALLOCATED
NCCOPR   15:1Ø:35 ******************************************************
NCCOPR   15:1Ø:35 ***                                                ***
NCCOPR   15:1Ø:35 *** NCCXDEL - NetView Centralized Console.         ***
NCCOPR   15:1Ø:35 ***           Erase work files:                    ***
NCCOPR   15:1Ø:35 ***           TEMP.NCCS.UTILITY.TXT.TØ13944        ***
NCCOPR   15:1Ø:35 ******************************************************
NCCOPR   DSI2Ø8I TIME EXPIRATION - ID= 'SYSØ3Ø77' - CMD= 'NCCXDEL
TEMP.NCCS.UTI LITY.RSH.TØ13944'
NCCOPR   NCCXDEL TEMP.NCCS.UTILITY.RSH.TØ13944
NCCOPR   DSI372I FREE COMMAND FAILED, 'NCCDWRK' IS NOT ALLOCATED
NCCOPR   CNM272I NCCDWRK IS NOW ALLOCATED
NCCOPR   CNM272I NCCDWRK IS NOW DEALLOCATED
NCCOPR   15:11:35 ******************************************************
NCCOPR   15:11:35 ***                                                ***
NCCOPR   15:11:35 *** NCCXDEL - NetView Centralized Console.         ***
NCCOPR   15:11:35 ***           Erasure work files:                  ***
NCCOPR   15:11:35 ***           TEMP.NCCS.UTILITY.RSH.TØ13944        ***
NCCOPR   15:11:35 ******************************************************
```

Espedito Morvillo
Systems Programmer (Italy)

An SMF termination exit for batch jobs

OS initiators give control to IEFACTRT during the termination of each job and the termination of each step within a job. Terminations may be normal or abnormal. An ample description of the purpose, environment, programming consideration, registers, and parameters associated with this exit is contained in *OS/390 MVS Installation Exits,* SC28-1753. I will not attempt to duplicate that documentation. What I shall do is describe the purpose of IEFACTRT, along with its processing. Descriptions of the processing performed in each section are contained in comments immediately preceding that section.

IEFACTRT comprises three CSECTs, IEFACTRT, ISDACTRT, and ISDRATES. IEFACTRT performs the primary processing. ISDACTRT uses the information on CPU and device usage that has been accumulated by IEFACTRT, calculates the costs of that usage, and creates System Message Blocks (SMBs) containing that information. ISDRATES is a table of rates for CPU seconds and print and punch records.

IEFACTRT processes SMF type 30 records, subtypes 4 and 5. All others are ignored. The work area referenced in IEFACTRT, and freed here at job termination, is obtained in IEFUSI at the beginning of a job; that part of IEFUSI is documented here. One of the first checks made in IEFACTRT is on the validity of that area. Afterwards, each section of a type 30 record is processed basically in the order presented to IEFACTRT.

Part of IEFACTRT that has been obviated by a scheduling product can be removed without difficulty. It was used to notify operators manning the production control area of abends or non-zero return codes that occurred in production jobs (a production job being one having a job name that began with an X or greater).

There is site-specific code that could be removed and some that will need to be modified. Statistical information about a job was not provided for some remotes since the charges approximated here were not the same as their actual bill. System identifiers must be modified so that true charges may be computed. CPU rates and charges for lines printed or cards punched (if anyone does that any more) must be changed to reflect rates at other sites.

ISDACTRT does the cost calculations for the SMF accounting exit, then passes that information back to IEFACTRT for formatting and printing.

ISDRATES contains the rates, edit patterns, titles, and constants that are referenced by the other two CSECTs. One change should be noted – PATWORK must have the value X'F021204040F02120' otherwise the year '2000' will be shown as '20 0' instead.

This exit is currently used on ESA 5.2, and on OS/390 in a test environment. It works with dates before and after 2000.

An extract of IEFUSI that is used to obtain virtual storage for KEEPSECT follows:

```
          STORAGE IS ACQUIRED FOR THE DATA AREA DESCRIBED
          BY KEEPSECT . THIS DATA AREA IS USED FOR INTER-
          STEP/JOB COMMUNICATION.
          THE AREA IS FREED IN IEFACTRT AT JOB TERMINATION.
          A POINTER TO THE AREA IS SAVED IN THE SMF COMMON
          EXIT PARAMETER LIST (+32) - MAPPED BY IEFJMR

          TITLE 'STEP INITIATION EXIT - IEFUSI'
 IEFUSI   CSECT ,
          USING IEFUSI,R12          ESTABLISH ADDRESSABILITY
          USING JMR,R2              ESTABLISH JMR ADDRESSABILITY
          STM   R14,R12,12(R13)     SAVE REGISTERS
          LR    R12,R15             LOAD BASE REGISTER
          LR    R5,R1               SAVE PARAMETER REGISTER
          USING PATDSECT,R5         SET PARAMETER LIST ADDRESSABILITY
          L     R2,PATCOMTB         RETRIEVE ADDRESS OF PARM AREA
          SPACE 1
         TM    JMRFLG,JMRSTRS+JMRCHRS+JMRCNRS+JMRWARM
         BNZ   PPGE            ON ANY RESTART, ISSUE GETMAIN
         SPACE 1
         SR    R8,R8                ZERO R8
         ICM   R8,15,JMRUCOM        PICK UP KEEPSECT ADDRESS
         LA    R3,KEEPLEN(R8)       POINT TO WORK AREA
         USING KEEPSECT,R8          TELL ASSEMBLER
         USING WORKAREA,R3          ESTABLISH WORKAREA ADDRESSABILAITY
         BNE   AUTHCK               BRANCH IF NOT FIRST STEP
         SPACE 1
PPGE STORAGE OBTAIN,LENGTH=KEEPLEN+WORKLEN,SP=KEEPSP
         LR    R8,R1                SAVE ADDRESS
         LA    R3,KEEPLEN(R8)       POINT TO WORK AREA
         XC    KEEPSECT(KEEPLEN),KEEPSECT ZERO OUT AREA
         ST    R8,JMRUCOM          SAVE ADDRESS IN COMMON
*                                  EXIT PARAMETER LIST
         MVI   PATBYTE,Ø           CLEAR ALL INDICATORS
         MVC   CLAMLOVE,=CL4'LOVE' STOW IDENTIFIER INTO WORKAREA
```

This JCL was used to create the statistical information, generated by IEFACTRT, that follows:

```
//XEPHON   JOB (1,BOGUSØØØ,PØ5K,1),ANONYMOUS,MSGCLASS=H,CLASS=J
/*ROUTE XEQ DON
//STEP1    EXEC PGM=IEBGENER,REGION=1ØØK
//SYSUT2   DD UNIT=VIO,SPACE=(TRK,1),DSN=&&LIMERICK,DISP=(,PASS)
//SYSPRINT DD DUMMY
//SYSIN    DD DUMMY
//SYSUT1   DD *
THERE WAS A YOUNG LADY NAMED HARRIS
WHOM MOST NOTHING WOULD EVER EMBARRASS
'TIL THE BATH SALTS ONE DAY
IN THE TUB WHERE SHE LAY
TURNED OUT TO BE PLASTER OF PARIS
//STEP2    EXEC PGM=IEBGENER,REGION=1ØØK
//SYSPRINT DD DUMMY
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=&&LIMERICK,DISP=(OLD,DELETE)
//SYSUT2 DD SYSOUT=*
```

Editor's note: the code from this article and an article by the same author entitled Re-creating USS and Logon Mode table macros *were unfortunately combined. The code published in the previous issue of* TCP/SNA Update *from the subheading IEFACTRT.SRT on page 32 until the end of that article should in fact be part of this article. We apologise for any confusion to our readers.*

```
***********************************************************************
*                                                                     *
*        PROCESS STORAGE AND PAGING SECTION UNLESS STEP FLUSHED       *
*                                                                     *
***********************************************************************
         SPACE 1
         L     R4,SMF3ØTOF         OFFSET TO COMPLETION SECTION
         AR    R4,SMF              COMPUTE COMPLETION SECTION ADDRESS
         USING SMF3ØCMP,R4         SET PAGING SECTION ADDRESSABILITY
         SPACE 1
         TM    SMF3ØSTI,SMF3ØFLH   TEST IF STEP FLUSHED
         BO    CMRFLUSH            BRANCH IF SO
         SPACE 1
         PLINE ,
         SPACE 1
         L     R4,SMF3ØROF         OFFSET TO STORAGE AND PAGING SECTION
         AR    R4,SMF              COMPUTE PAGING SECTION ADDRESS
         USING SMF3ØSAP,R4         SET PAGING SECTION ADDRESSABILITY
         SPACE 1
         MVC   HPPI,PATPAGE        MOVE PAGING EDIT PATTERNS
         MVC   HPPO,PATPAGE         INTO OUTPUT AREA
*        MVC   HPPR,PATPAGE
         MVC   HPVI,PATPAGE
         MVC   HPVO,PATPAGE
         MVC   HPVR,PATPAGE
         SPACE 1
         L     R1,SMF3ØPGI         FETCH STEP PAGE-INS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HPPI,DOUBLE+5
         L     R1,SMF3ØPGO         FETCH STEP PAGE-OUTS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HPPO,DOUBLE+5
*        L     R1,SMF3ØREC         FETCH STEP RECLAIMS
*        CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
*        ED    HPPR,DOUBLE+5
         L     R1,SMF3ØVPI         FETCH STEP VIO PAGE-INS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HPVI,DOUBLE+5
         L     R1,SMF3ØVPO         FETCH STEP VIO PAGE-OUTS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HPVO,DOUBLE+5
         L     R1,SMF3ØVPR         FETCH STEP VIO RECLAIMS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HPVR,DOUBLE+5
         EJECT
         MVC   HCPI,CPPI           MOVE PAGING CONSTANTS
         MVC   HCPO,CPPO            INTO OUTPUT AREA
         MVC   HCPR,CPPR
         MVC   HCVI,CPVI
         MVC   HCVO,CPVO
         MVC   HCVR,CPVR
         SPACE 1
         PLINE ,
         MVC   HCCIN,PATPAGE       MOVE PAGING EDIT PATTERNS
         MVC   HCCRCLAM,PATPAGE     INTO OUTPUT AREA
         MVC   HCLIN,PATPAGE
         MVC   HCLRCLAM,PATPAGE
         SPACE 1
         L     R1,SMF3ØCPI         FETCH CSA PAGE-INS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HCCIN,DOUBLE+5
         L     R1,SMF3ØHPI         FETCH HYPERSPACE PAGE IN
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HCCRCLAM,DOUBLE+5
         L     R1,SMF3ØLPI         FETCH LPA PAGE-INS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HCLIN,DOUBLE+5
         L     R1,SMF3ØHPO         FETCH HYPERSPACE PAGE OUT
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HCLRCLAM,DOUBLE+5
         SPACE 1
         MVC   HCPAGE,CCCSAIN      MOVE COMMON PAGING CONSTANTS
         MVC   HCRECLAM,CCRECLAM     INTO OUTPUT AREA
         MVC   HLPAIN,CLPAIN
         MVC   HLRECLAM,CLRECLAM
         SPACE 1
         PLINE ,
         EJECT
         MVC   HSSIN,PATPAGE       MOVE SWAPPING EDIT PATTERNS
         MVC   HSSOUT,PATPAGE       INTO OUTPUT AREA
         MVC   HSSS,PATPAGE
         MVC   HSSTOLEN,PATPAGE
         SPACE 1
         L     R1,SMF3ØPSI         FETCH NUMBER OF PAGES SWAPPED IN
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HSSIN,DOUBLE+5
         L     R1,SMF3ØPSO         FETCH NUMBER OF PAGES SWAPPED OUT
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HSSOUT,DOUBLE+5
         L     R1,SMF3ØNSW         FETCH NUMBER OF SWAP SEQUENCES
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HSSS,DOUBLE+5
         L     R1,SMF3ØPST         FETCH NUMBER OF PAGES STOLEN
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HSSTOLEN,DOUBLE+5
         SPACE 1
         MVC   HSWAPING,CSWAPING   MOVE SWAPPING CONSTANTS
         MVC   HCSIN,CSWAPIN        INTO OUTPUT AREA
         MVC   HCSOUT,CSWAPOUT
         MVC   HCSTOLEN,CPSTOLEN
         SPACE 1
         PLINE ,
         PLINE ,                   SEPARATOR LINE
         EJECT
         MVC   HRTCONA,CSSIZE      MOVE REGION CONSTANTS
         MVC   HRLOC2(L'CSREQ),CSREQ  INTO OUTPUT AREA
         MVI   HCK,C'K'
         TM    SMF3ØSFL,SMF3ØVQR   TEST IF V=R WAS SPECIFIED
         BNO   CMRVIRT             BRANCH IF NOT
         MVC   HRTYPE2,CSREAL       ELSE SO INDICATE
CMRVIRT  MVC   HRREQ2,PATPAGE      MOVE IN EDIT PATTERN
         SPACE 1
         L     R1,SMF3ØRGN         FETCH REGION REQUESTED
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HRREQ2,DOUBLE+5
         PLINE ,
         SPACE 1
         MVC   HRLOC2,CSUBELOW     REGION CONSTANT INTO OUTPUT AREA
         MVI   HCK,C'K'
         MVC   HRREQ2,PATPAGE      MOVE IN EDIT PATTERN
         SPACE 1
         L     R1,SMF3ØURB         FETCH REGION USED BELOW 16 MEG
         SR    RØ,RØ               CLEAR REMAINDER REGISTER
         D     RØ,CMRF1Ø24         CONVERT BYTES TO 1K UNITS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HRREQ2,DOUBLE+5
         PLINE ,
         SPACE 1
         MVC   HRLOC2,CSUABOVE     REGION CONSTANT INTO OUTPUT AREA
         MVI   HCK,C'K'
         MVC   HRREQ2,PATPAGE      MOVE IN EDIT PATTERN
         SPACE 1
         L     R1,SMF3ØEUR         FETCH REGION USED FROM ABOVE 16 MEG
         SR    RØ,RØ               CLEAR REMAINDER REGISTER
         D     RØ,CMRF1Ø24         CONVERT BYTES TO 1K UNITS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HRREQ2,DOUBLE+5
         PLINE ,
         SPACE 1
         MVC   HRTCONA,CSMAX       MOVE REGION CONSTANTS
         MVC   HRLOC2(L'CSBELOW),CSBELOW INTO OUTPUT AREA
         MVI   HCK,C'K'
         MVC   HRREQ2,PATPAGE      MOVE IN EDIT PATTERN
         SPACE 1
         L     R1,SMF3ØARB         LSQA+SWA REGION BELOW BELOW 16 MEG
         SR    RØ,RØ               CLEAR REMAINDER REGISTER
         D     RØ,CMRF1Ø24         CONVERT BYTES TO 1K UNITS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HRREQ2,DOUBLE+5
         PLINE ,
         SPACE 1
         MVC   HRTCONA,CSMAXC      MOVE REGION CONSTANTS
         MVC   HRLOC2,CSABOVE        INTO OUTPUT AREA
         MVI   HCK,C'K'
         MVC   HRREQ2,PATPAGE      MOVE IN EDIT PATTERN
         SPACE 1
         L     R1,SMF3ØEAR         LSQA+SWA REGION ABOVE 16 MEG
         SR    RØ,RØ               CLEAR REMAINDER REGISTER
         D     RØ,CMRF1Ø24         CONVERT BYTES TO 1K UNITS
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         ED    HRREQ2,DOUBLE+5     BEAUTIFY VALUES
         PLINE ,
         EJECT
************************************************************************
*                                                                      *
*        PROCESS EXCP SECTION                                          *
*                                                                      *
************************************************************************
         SPACE 1
```

```
CMRNOTXA SR    RØ,RØ               CLEAR CORRUPTIBLE REGISTER
         CH    RØ,SMF3ØEON         TEST IF EXCP SECTION EXISTS
         BE    CMRFLUSH            BRANCH IF NOT
         PLINE ,                   WRITE SEPARATOR LINE
         SPACE 1
         MVC   HIODDNAM,CIODDNAM   MOVE EXCP CONSTANTS
         MVC   HIOBLKSZ,CIOBLKSZ    INTO OUTPUT AREA
         MVC   HIOUNIT,CIOUNIT
         MVC   HIOADDR,CIOADDR
         MVC   HIOEXCP,CIOEXCP
         MVC   H2ODDNAM(H2ODDNAM-HIODDNAM-2),HIODDNAM
         PLINE ,
         SPACE 1
CLAMEOF  L     R4,SMF3ØEOF         OFFSET TO EXCP SECTION
         AR    R4,SMF              COMPUTE EXCP SECTION ADDRESS
         USING SMF3ØEXP,R4         SET EXCP SECTION ADDRESSABILITY
         LH    R5,SMF3ØELN         LENGTH OF AN EXCP ENTRY
         LH    R6,SMF3ØEON         NUMBER EXCP SECTIONS IN THIS RECORD
         LR    R3,R4               POINT TO NEXT
         SR    R3,R5                PREVIOUS ENTRY
         SPACE 1
CMREXCPS LA    R15,HIODDNAM        POINT TO OUTPUT AREA
         USING CMRDSECT,R15        ESTABLISH ADDRESSABILITY TO CMRDSECT
         BAS   R8,CMRFILLX         COMPLETE EXCP SECTION
         AR    R4,R5               POINT TO NEXT EXCP SECTION
         SPACE 1
         C     R6,CMRF1            TEST IF ANY SECTIONS LEFT
         BE    CMRDUNN             BRANCH IF SO
         BCTR  R6,RØ               REDUCE I/O SECTION BY ONE
         LA    R15,H2ODDNAM-HIODDNAM(R15) NEXT AREA FOR I/O SECTION
         BAS   R8,CMRFILLX         COMPLETE EXCP SECTION
         AR    R4,R5               POINT TO NEXT EXCP SECTION
CMRDUNN  DS    ØH
         PLINE ,
         BCT   R6,CMREXCPS         PROCESS NEXT SECTION
         EJECT
************************************************************************
*                                                                      *
*        PROCESS SUBSEQUENT EXCP SECTIONS                              *
*                                                                      *
************************************************************************
         SPACE 1
         ICM   RØ,3,SMF3ØEOR       TEST IF ANY SUBSEQUENT EXCP RECORDS
         BE    CMRXDONE            BRANCH IF NOT
         SH    SMF,CMRH4           POINT TO NEXT BUFFER POINTER
         ICM   SMF,15,Ø(SMF)       FETCH IT
         BE    CMRXDONE            SHOULD NOT OCCUR
         LA    SMF,8(SMF)          POINT TO SUBSEQUENT SMF BUFFER
         B     CLAMEOF              AND PROCESS IT
         SPACE 1
CMRXDONE L     SMF,KEEPBMP         RESTORE POINTER TO FIRST SMF BUFFER
```

```
         PLINE ,
         EJECT
         MVC   HTIOEXCP,PATEXCP     TOTAL EXCP PATTERNS
         MVC   HTIOTAPE,PATEXCP       TO OUTPUT AREA
         MVC   HTIOJV,PATEXCP
         SPACE 1
         L     R1,DISKEXCP         FETCH ACCUMULATED DISK EXCP COUNTS
         CVD   R1,DOUBLE           CONVERT TO PACKED DECIMAL
         ED    HTIOEXCP,DOUBLE+3     THENCE TO EBCDIC
         L     R1,TAPEEXCP         FETCH ACCUMULATED TAPE EXCP COUNTS
         CVD   R1,DOUBLE            CONVERT TO PACKED DECIMAL
         ED    HTIOTAPE,DOUBLE+3     THENCE TO EBCDIC
         L     R1,VIOEXCPS         FETCH ACCUMULATED JES+VIO EXCP COUNT
         CVD   R1,DOUBLE            CONVERT TO PACKED DECIMAL
         ED    HTIOJV,DOUBLE+3       THENCE TO EBCDIC
         SPACE 1
         MVC   HCTEXCP,CTIODISK    TOTAL EXCP CONSTANTS
         MVC   HCTAPE,CTIOTAPE      TO OUTPUT AREA
         MVC   HCJESV,CTIOJESV
         PLINE ,
         B     CMROPER
         EJECT
************************************************************************
*                                                                      *
*        FORMAT EXCP SECTION                                           *
*                                                                      *
************************************************************************
         SPACE 1
CMRFILLX MVC   CMRDDNAM,SMF3ØDDN   DDNAME USED TO ACCESS DATASET
         MVI   CMRADDR,C'*'        INITIALIZE ADDR TYPE
         MVC   CMRADDR+1(L'CMRADDR-1),CMRADDR  TO STARS
         CLC   CMRDDNAM,SMF3ØDDN-SMF3ØDEV(R3) SAME DDNAME AS BEFORE?
         BNE   CMRUNIQE            BRANCH IF NOT
         MVC   CMRDDNAM,CMRSAME    SHOW CONCATENATED DATASET
CMRUNIQE AR    R3,R5               POINT TO NEXT PREVIOUS DDNAME
         LH    R1,SMF3ØBSZ         FETCH BLKSIZE
         LTR   R1,R1                AND TEST IF IT IS ZERO
         BNE   CMRMBPAT            BRANCH IF NOT
         MVC   CMRBLKSZ+4(3),CMRNA  ELSE SHOW UNAVAILABLE
         B     CMGSAME             ONWARD...
         SPACE 1
CMRMBPAT MVC   CMRBLKSZ,PATBLKSZ   EDIT PATTERN TO OUTPUT AREA
         N     R1,CMR7FFF           CLEAR TRASH
         CVD   R1,DOUBLE            CONVERT TO PACKED DECIMAL
         ED    CMRBLKSZ,DOUBLE+5     THENCE TO EBCDIC
*        TM    SMF3ØBSZ,X'8Ø'      TEST IF BLOCK SIZE WAS CHANGED
*        BNO   CMGSAME             BRANCH IF NOT(CODE DISABLED
*        MVI   CMRBLKSZ,C'C'       SHOW BLOCK SIZE WAS CHANGED
         SPACE 1
CMGSAME  ICM   R1,15,SMF3ØBLK      FETCH BLOCK COUNT FOR THIS DATASET
         BNE   CMRMCPAT            BRANCH IF AVAILABLE
         MVC   CMREXCP+9(3),CMRNA  ELSE SHOW UNAVAILABLE
         B     CMGRCRØ              AND CONTINUE
         SPACE 1
CMRMCPAT MVC   CMREXCP,PATEXCP     EDIT PATTERN TO OUTPUT AREA
*        IF THIS IS A PSF PAGE DS --- RESET INDICATOR BIT
         N     R1,=X'7FFFFFFF'     RESET PAGE FLAG IF PRESENT
         CVD   R1,DOUBLE           CONVERT TO PACKED DECIMAL
         ED    CMREXCP,DOUBLE+3     THENCE TO EBCDIC
         SPACE 1
CMGRCRØ  SR    RØ,RØ               ZERO WORK REGISTER
         CH    RØ,SMF3ØDEV         TEST IF REAL DEVICE
         BNE   CMREAL              BRANCH IF SO
         EJECT
*        DETERMINE TYPE OF PSEUDO DEVICE
         SPACE 1
         TM    SMF3ØCUA,X'8Ø'      TEST IF VIRTUAL DEVICE
         BNO   CMRTESTJ            BRANCH IF NOT
         MVC   CMRUNIT,CMRVIRTC    SET 'VIRTUAL' DEVICE TYPE
         BR    R8                  RETURN TO CALLER
         SPACE 1
CMRTESTJ A     R1,VIOEXCPS         ACCUMULATE JES+VIO EXCP COUNTS
         ST    R1,VIOEXCPS         REVISE TOTALS
         MVC   CMRUNIT,CMRJES2     ASSUME JES2 DATASET
*        TEST FOR PSF PAGE DS AND FORMAT ACCORDINGLY
         TM    SMF3ØBLK,X'8Ø'      TEST IF PAGE EXCP COUNT
         BZ    CMRNOTPG            ZERO MEANS NOT PAGE COUNT
         NC    SMF3ØBLK,=X'7FFFFFFF' TURN OFF PAGE FLAG BIT
         MVC   CMRUNIT,CMRPSF      MARK UNIT FIELD AS PSF
         MVC   CMRADDR,CMRPSFPG    MARK ADR FIELD AS PSF
CMRNOTPG CLC   SMF3ØCUA,CMR7FFF+2  TEST IF VIO DATASET
         BNE   CMRCKDMY            BRANCH IF NOT
         MVC   CMRUNIT,CMRVIO      INDICATE VIO DATASET
         BR    R8                  RETURN TO CALLER
         EJECT
*        DETERMINE IF DATASET IS ACTUALLY A DUMMY ONE
         SPACE 1
CMRCKDMY ICM   R1,15,ADDRLCT       FETCH LINKAGE CONTROL TABLE ADDRESS
         BCR   8,R8                RETURN IF UNAVAILABLE
         USING LCTDSECT,R1         ESTABLISH LCT ADDRESSABILITY
         SPACE 1
         SR    RØ,RØ               CLEAR VOLATILE REGISTER
         ICM   RØ,7,LCTSCTAD+1     ADDRESS OF STEP'S STEP CONTROL TABLE
         BCR   8,R8                RETURN IF ZERO
         LR    R1,RØ               PRIME SCT BASE
         USING INSMSCT,R1          SET STEP CONTROL TABLE BASE
         SPACE 1
         SR    R14,R14             CLEAR VOLATILE REGISTER
         ICM   R14,7,SCTFSIOT      GET TTR OF FIRST SIOT
         BCR   8,R8                RETURN IF ZERO
         LA    R14,16(R14)         SET VIRTUAL ADDRESS OF SIOT
         USING INDMSIOT,R14        ESTABLISH SIOT ADDRESSABILITY
         SPACE 1
CMRFINDD CLC   SMF3ØDDN,SCTDDNAM   TEST IF SAME DDNAME
         BE    CMRSAMDD            BRANCH IF SO
         ICM   R14,7,SCTPSIOT      FETCH TTR OF NEXT SIOT
         LA    R14,16(R14)         VIRTUAL ADDRESS OF SIOT
         BNZ   CMRFINDD            BRANCH IF NOT
         MVC   CMRADDR,CMRDYN      SHOW UNLOCATABLE DD STATEMENT
         BCR   15,R8               RETURN TO CALLER
         SPACE 1
CMRSAMDD TM    SCTSBYT1,SCTDUMMY   TEST IF DUMMY DATASET
         BNO   Ø(R8)               BRANCH IF NOT
         MVC   CMRUNIT,CMRDUMY     SHOW DATASET IS A DUMMY ONE
         BCR   15,R8               RETURN TO CALLER
         SPACE 1
         DROP  R1,R14
         EJECT
*        DETERMINE REAL DEVICE TYPE
         SPACE 1
CMREAL   UNPK  DOUBLE(5),SMF3ØCUA(3) DEVICE NUMBER
         TR    DOUBLE(4),CMRTRANS  CONVERT TO EBCDIC
         MVC   CMRADDR,DOUBLE       AND MOVE TO OUTPUT AREA
         SPACE 1
         CLI   CMRADDR,C'Ø'        TEST FOR A LEADING ZERO
         BNE   CMRLOTSA            BRANCH IF NOT ONE
         MVI   CMRADDR,C' '         ELSE REPLACE IT WITH A BLANK
         SPACE 1
CMRLOTSA CLI   SMF3ØDEV,UCB3DACC   TEST IF DASD DEVICE
         BE    CMRDISKD            BRANCH IF SO
         CLI   SMF3ØDEV,UCB3TAPE   TEST IF TAPE DEVICE
         BE    CMRTAPED            BRANCH IF SO
         CLI   SMF3ØDEV,UCB3UREC   TEST IF UNIT RECORD DEVICE
         BNE   CMRQMARK            BRANCH IF NOT
         MVC   CMRUNIT,CMRUR       SHOW UNIT RECORD DEVICE
         BR    R8                  RETURN TO CALLER
         SPACE 1
CMRDISKD A     R1,DISKEXCP          ELSE ACCUMULATE DISK EXCP COUNTS
         ST    R1,DISKEXCP           IN ACCUMULTOR
         LA    R14,DASDTABL        POINT TO DASD TYPES
         LA    R1,DASDCNT          SET NUMBER OF ENTRIES
CMRUSRCH CLC   SMF3ØUTP,Ø(R14)     TEST IF DEVICE IS IN TABLE
         BNE   CMRTRYP1            BRANCH IF NOT
         MVC   CMRUNIT,1(R14)      SHOW DEVICE TYPE
         BR    R8                  RETURN TO CALLER
         SPACE 1
CMRTRYP1 LA    R14,CMRUSIZE(R14)   POINT TO NEXT ENTRY
         BCT   R1,CMRUSRCH         BRANCH IF NOT
         SPACE 1
CMRQMARK MVC   CMRUNIT,QMARKS      SHOW UNKNOWN DEVICE TYPE
         BR    R8                  RETURN TO CALLER
         SPACE 1
```

```
CMRTAPED A     R1,TAPEEXCP          ACCUMULATE TAPE EXCP COUNTS
         ST    R1,TAPEEXCP           IN ACCUMULATOR
         EJECT
*        THE DEVICE IS MOUNTABLE SO CHECK TO SEE IF IT HAS ALREADY
*        OCCURRED IN THE DEVICE LIST.  IF SO, DON'T COUNT IT AGAIN.
         SPACE 1
         STM   R2,R6,SAVE1          MAKE WORK REGISTERS AVAILABLE
         ICM   R2,15,TAPEUSCT       IF THIS IS THE FIRST DEVICE
         BZ    MOUNTABL              THEN WE MUST COUNT IT
         SPACE 1
         MVC   TMPDEVAD,SMF3ØCUA    GET ORIGINAL BYTES FOR COMPARE
         LR    R5,R4                GET ADDR OF CURRENT ENTRY AS LIMIT
         BCTR  R5,Ø                  BUT WE DON'T WANT TO LOOK AT IT
         SPACE 1
         LH    R4,SMF3ØELN          LENGTH OF EACH EXCP SECTION
         L     R6,SMF3ØEOF          OFFSET TO BEGINNING OF EXCP SECTION
         AR    R6,SMF               POINT TO FIRST ENTRY IN EXCP SECTION
         SPACE 1
CMRCHECK CLC   TMPDEVAD,SMF3ØCUA-SMF3ØDEV(R6) DEVICE ALREADY COUNTED?
         BE    NOMOUNTA             IF SO, DON'T COUNT IT AGAIN
         BXLE  R6,R4,CMRCHECK       TRY AGAIN
         SPACE 2
MOUNTABL A     R2,CMRF1             INCREMENT TAPE USE COUNT BY ONE
         ST    R2,TAPEUSCT            AND REVISE IT
NOMOUNTA LM    R2,R6,SAVE1          NO LONGER NEED WORK REGISTERS
         SPACE 1
         LA    R14,TAPETABL         POINT TO TAPE TYPES
         LA    R1,TAPECNT           SET NUMBER OF ENTRIES
         B     CMRUSRCH             DETERMINE TAPE DEVICE TYPE
         DROP  R15                  FORGET CMR
         EJECT
***********************************************************************
*                                                                     *
*        FORMAT OPERATOR SECTION                                      *
*                                                                     *
***********************************************************************
         SPACE 1
CMROPER  ICM   R4,15,SMF3ØOOF       OFFSET TO OPERATOR SECTION
         BE    CMRPCS               BRANCH IF NONEXISTENT
         AR    R4,SMF               COMPUTE OPERATOR SECTION ADDRESS
         USING SMF3ØOPS,R4          SET OPERATOR SECTION ADDRESSABILITY
         SPACE 1
         MVC   HTTMSPEC,PATSTEP#    MOVE EDIT PATTERNS
         MVC   HTTMNSPC,PATSTEP#     INTO OUTPUT AREA
         MVC   HTTTUSED,PATSTEP#
         SPACE 1
         L     R1,SMF3ØTPR          FETCH NUMBER OF SPECIFIC TAPE MOUNTS
         CVD   R1,DOUBLE             ALTER RADIX TO PACKED DECIMAL
         AH    R1,KEEPTPR           TALLY SPECIFIC TAPE MOUNTS
         STH   R1,KEEPTPR           REVISE TOTALS
         ED    HTTMSPEC,DOUBLE+6     THENCE TO EBCDIC
         L     R1,SMF3ØPTM         FETCH NUMBER OF NON-SPEC TAPE MOUNTS
         CVD   R1,DOUBLE            ALTER RADIX TO PACKED DECIMAL
         AH    R1,KEEPPTM          TALLY NON-SPECIFIC TAPE MOUNTS
         STH   R1,KEEPPTM          REVISE TOTALS
         ED    HTTMNSPC,DOUBLE+6     THENCE TO EBCDIC
         L     R1,TAPEUSCT         FETCH NUMBER OF TAPE UNITS USED
         CVD   R1,DOUBLE            ALTER RADIX TO PACKED DECIMAL
         AH    R1,KEEPUSCT         TALLY TAPE DRIVE USED
         STH   R1,KEEPUSCT         REVISE TOTALS
         ED    HTTTUSED,DOUBLE+6     THENCE TO EBCDIC
         SPACE 1
         MVC   HCTMOUNT,CTTMOUNT   MOVE TOTAL MOUNTS CONSTANTS
         MVC   HCTNM,CTTMN          INTO OUTPUT AREA
         MVC   HCTUSED,CTTUSED
         SPACE 1
         PLINE ,
CMRFLUSH DS    ØH
         PLINE ,                   SEPARATOR LINE
         EJECT
***********************************************************************
*                                                                     *
*        PROCESS COMPLETION SECTION                                   *
*                                                                     *
***********************************************************************
         SPACE 1
CMRPCS   L     R4,SMF3ØTOF         OFFSET TO COMPLETION SECTION
         AR    R4,SMF              COMPUTE COMPLETION SECTION ADDRESS
         USING SMF3ØCMP,R4         SET COMPLETION SECTION BASE
         SPACE 1
         MVC   HCOMP,CCOMP         SET 'STEP COMPLETION CODE'
         MVC   HCCC,CCCC           SET 'CC=' IN INFORMATIVE LINE
         UNPK  CLAMWORK(3),SMF3ØSCC+1(2) MAKE STEP COMPLETION
         TR    CLAMWORK(2),CMRTRANS                    CODE PRINTABLE
         SPACE 1
         MVC   HCCCODE,CLAMWORK    ASSUME 'NORMAL' STEP TERMINATION
         TM    SMF3ØSTI,SMF3ØABD+SMF3ØFLH TEST ASSUMPTION
         BZ    CMRNORM             BRANCH IF VALID
         SPACE 1
         MVC   HCACODE,CFLUSH      ASSUME STEP WAS FLUSHED
         TM    SMF3ØSTI,SMF3ØFLH   TEST ASSUMPTION
         BO    CMRNORM             BRANCH IF VALID
         MVC   HCACODE,HCACODE-1   FLUSH FLUSHED
         SPACE 1
         MVC   HABERC,CABERC       MOVE REASON CONSTANT IN OUTPUT AREA
         UNPK  CLAMWORK(9),SMF3ØARC(5) MAKE ABEND REASON
         TR    CLAMWORK(8),CMRTRANS                 CODE PRINTABLE
         MVC   HCRCODE,CLAMWORK    TRANSFER IT TO OUTPUT AREA
         SPACE 1
         TM    SMF3ØSCC,SMF3ØSAB   TEST IF USER ABEND CODE
         BO    CMRUSERA            BRANCH IF SO
         SPACE 1
         UNPK  CLAMWORK(5),SMF3ØSCC(3) MAKE SYSTEM ABEND
         TR    CLAMWORK+1(3),CMRTRANS                    CODE PRINTABLE
         MVC   HCSA,CLAMWORK+1     TRANSFER IT TO OUTPUT AREA
         MVI   HCACODE,C'S'        INDICATE SYSTEM ABEND
         B     CMRNORM             TRANSCRIBE LINE
         SPACE 1
CMRUSERA LH    R1,SMF3ØSCC         FETCH USER ABEND CODE
         N     R1,CMRFFF           CLEAR TRASH
         CVD   R1,DOUBLE           ALTER RADIX TO PACKED DECIMAL
         UNPK  HCUA,DOUBLE+5(3)     THENCE TO EBCDIC
         OI    HCUA+L'HCUA-1,24Ø   FOLD LEAST SIGNIFICANT DIGIT
         MVI   HCACODE,C'U'        SHOW USER ABEND
         SPACE 1
CMRNORM  PLINE ,                   COMPLETION CODE LINE
         EJECT
************************************************************************
*                                                                      *
*  REPORT PRODUCTION JOB ANOMALIES TO ROUTCDE 15;                      *
*  SPECIFICALLY, ABENDS AND RETURN CODES GREATER THAN 8.               *
*                                                                      *
************************************************************************
         SPACE 1
         L     R4,SMF3ØIOF         OFFSET TO IDENTIFICATION SECTION
         AR    R4,SMF              ADDRESS OF IDENTIFICATION SECTION
         USING SMF3ØID,R4          SET IDENT SECTION ADDRESSABILITY
         SPACE 1
         CLI   SMF3ØJBN,C'X'       TEST FOR PRODUCTION JOB
         BL    MNOPROD             IF NOT IGNORE IT
         SPACE 1
         L     R4,SMF3ØTOF         OFFSET TO COMPLETION SECTION
         AR    R4,SMF              ADDRESS OF COMPLETION SECTION
         USING SMF3ØCMP,R4         SET COMPLETION SECTION ADDRESSABILITY
         SPACE 1
         TM    SMF3ØSTI,SMF3ØSAB+SMF3ØABD+SMF3ØFLH TEST IF NORMAL COMP
         BZ    MGETCC              BRANCH IF SO
         MVC   MONITOR(MLNGTHMN),MONITORL INITIALIZE MESSAGE AREA
         TM    SMF3ØSTI,SMF3ØSAB+SMF3ØABD TEST IF STEP ABENDED
         BZ    MFLUSH              BRANCH IF NOT
         MVC   MONITOR+MLNGTHMN-4(4),MNOJCTMG+MLNGTHNJ-4 DESC2,R=(1,15)
         UNPK  JOBARGS(5),SMF3ØSCC(3) EXPAND COMPLETION CODE
         TR    JOBARGS+1(3),CMRTRANS CONVERT TO EBCDIC
         MVC   MCC+1(3),JOBARGS+1  PLACE IN MONITOR MESSAGE
         TM    SMF3ØSCC,X'8Ø'      TEST IF USER ABEND CODE
         BZ    MGETJCT             BRANCH IF NOT
         MVI   MCC,C'U'            INDICATE USER ABEND REQUEST
         B     MGETJCT             BUILD MONITOR MESSAGE
         SPACE 1
MJCTLOST MVC   MNOJCTML(MLNGTHNJ),MNOJCTMG ERROR MESSAGE TO WORK AREA
```

```
         SPACE 1
         L     R6,ADDREXD          GET ADDR OF EXD
         USING EXDDSECT,R6          AND ESTABLISH ADDRESSABILITY TO IT
         L     R6,EXDCOMTB         GET ADDR OF COMMON EXIT TABLE
         USING JMR,R6              ESTABLISH TEMP ADDRESSABILITY
         SPACE 1
         MVC   MNOJCTJN,JMRJOB     INCLUDE JOB NAME
         B     MDISPLAY            INFORM OPERATOR AND CONTINUE...
         SPACE 1
MGETCC   CLC   SMF3ØSCC,CMRH8      TEST IF COMP CODE EXCEEDS 8
         BNH   MNOPROD             BRANCH IF NOT
         MVC   MONITOR(MLNGTHMN),MONITORL WTO PATTERN TO WORK AREA
         MVC   MTYPE,CMREND        SHOW STEP ENDED WITH CC GT EIGHT
         B     MTERM               ENTER COMMON CODE
         SPACE 1
MFLUSH   MVC   MTYPE,CFLUSH        SHOW STEP WAS BYPASSED
MTERM    LH    R3,SMF3ØSCC         FETCH COMPLETION CODE
         N     R3,CMRFFFF          DESTROY GARBAGE
         CVD   R3,JOBARGS          ALTER RADIX
         UNPK  MCC-1(5),JOBARGS+5(3) COMPLETION CODE TO MESSAGE
         OI    MCC+3,X'FØ'         INSURE LAST CHARACTER PRINTABLE
         MVI   MCC-1,64            BLANK GARBAGE
MGETJCT  L     R3,ADDRLCT          GET LINKAGE CONTROL TABLE ADDRESS
         USING LCTDSECT,R3         ESTABLISH LCT REFERENCE POINT
         SR    R1,R1               CLEAR REGISTER ONE
         ICM   R1,7,LCTSCTAD+1     OBTAIN JOB STEP SCT ADDRESS
         ICM   R3,7,LCTJCTAD+1     OBTAIN JOB'S JCT ADDRESS
         BZ    MJCTLOST            BRANCH IF UNAVAILABLE
         MVC   MJOBNAME,8(R3)      HANDLE OF FAILING JOB(JCTJNAME(ESA))
         SLR   R2,R2               CLEAR WORK REGISTER
         SPACE 1
         L     R4,SMF3ØIOF         OFFSET TO IDENTIFICATION SECTION
         AR    R4,SMF              ADDRESS OF IDENTIFICATION SECTION
         USING SMF3ØID,R4          SET IDENT SECTION ADDRESSABILITY
         SPACE 1
         IC    R2,SMF3ØSTN         OBTAIN STEP NUMBER
         CVD   R2,JOBARGS          ALTER RADIX
         UNPK  MSTEPNO,JOBARGS+6(2) PLACE STEP NUMBER IN MESSAGE
         OI    MSTEPNO+2,24Ø       INSURE LAST DIGIT IS PRINTABLE
         LTR   R1,R1               TEST IF SCT WAS AVAILABLE
         BE    MOLDWAY             BRANCH IF NOT
         LA    R3,Ø(R1)            INITIALIZE SCT POINTER
         B     MFILLMSG            COMPLETE MESSAGE PROCESSING
         SPACE 1
         USING INSMSCT,R3          ESTABLISH SCT ADDRESSABILITY
MOLDWAY  ICM   R3,7,JCTSDKAD(R3)   DISK ADDRESS OF FIRST SCT
         BZ    MNOSCT              BRANCH IF UNAVAILABLE
MCOUNT   BCTR  R2,Ø                DECREMENT STEP COUNTER
         LA    R3,16(R3)           POINT TO SCT PROPER
         C     R1,SCTANSCT         UCC11 CAUSE PREMATURE END OF SCTS?
         BE    MNOSCT              BRANCH IF SO
         CR    R2,R1               TEST IF CURRENT STEP
         BE    MFILLMSG            BRANCH IF SO
         ICM   R3,7,SCTANSCT       NEXT STEP CONTROL TABLE
         BNZ   MCOUNT              LOOP POWER
         SPACE 1
MNOSCT   DS    ØH
         MVC   MPROCSTP,UNKNOWN    UNKNOWN STEP NAME
         MVC   MSTEPNAM,MPROCSTP   UNKNOWN STEP NAME
         MVC   MPROGRAM,MPROCSTP   UNKNOWN PROGRAM NAME
         B     MDISPLAY            ISSUE ERROR MESSAGE
         SPACE 1
MFILLMSG MVC   MPROCSTP,SCTSCLPC   PROC INVOCATION STEP NAME
         MVC   MSTEPNAM,SCTSNAME   STEP NAME
         MVC   MPROGRAM,SCTPGMNM   PROGRAM NAME
MDISPLAY WTO   MF=(E,MONITOR)      DISPLAY INFORMATION FOR MONITOR GRUP
         SPACE 1
         DROP  R3,R6               REMOVE ADDRESSABILITY
         EJECT
MNOPROD  DS    ØH
         SPACE 1
*        SEE IF THE STEP WAS EXECUTED
         SPACE 1
         L     R4,SMF3ØTOF         OFFSET TO COMPLETION SECTION
         AR    R4,SMF              ADDRESS OF COMPLETION SECTION
         USING SMF3ØCMP,R4         SET COMPLETION SECTION ADDRESSABILITY
         SPACE 1
         TM    SMF3ØSTI,SMF3ØFLH   TEST IF STEP WAS FLUSHED
         BO    RETURN              BRANCH IF SO
         SPACE 1
         L     R4,SMF3ØROF         OFFSET TO STORAGE SECTION
         AR    R4,SMF              ADDRESS OF STORAGE SECTION
         USING SMF3ØSAP,R4         SET STORAGE SECTION ADDRESSABILITY
         SPACE 1
         CLC   SMF3ØRGN,CMRFØ      TEST IF STEP WAS EXECUTED
         BE    RETURN              BRANCH IF NOT
         SPACE 3
*        STEP WAS EXECUTED
         SPACE 1
CMRUN    L     R15,ADDREXD         ADDRESS OF IEFACTRT PARAMETER LIST
         USING EXDDSECT,R15        PROVIDE ADDRESSABILITY TO PARM LIST
         L     R3,EXDSRT           GET ADDR OF STEP CPU TIME
         MVC   RUNTIME+1(3),Ø(R3)   AND SAVE
         L     R3,RUNTIME          GET CPU TIME
         ST    R3,CPUTIME           TO PARAMETER LIST FOR ISDACTRT
         EJECT
```

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

A mailbox system for SMTP under MVS TCP/IP – 6

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

```
IF &STR(&SYSNSUB(1,&DM)) NE &STR() THEN DO
 SET &RV = RV
 SET &RP = Ø
 ISPEXEC VGET (MAXRP)
 DO WHILE &RP < &MAXRP
  SET &RP = &RP + 1
  IF &RP = &MAXRP THEN DO
   WRITE ====> Error, loop on RP terminated (&SYSICMD).
  END
  SET &RN = 1
  ISPEXEC VGET (RV&RN)
  SET &C = &STR(&SYSNSUB(2,&&RV&RN))
  DO WHILE &STR(&SYSNSUB(1,&C)) NE &STR() AND &RN < &MAXRP
   IF &STR(&SYSLC(&SYSNSUB(1,&CC))&ATSIGN+
   &SYSLC(&SYSNSUB(1,&DM))) = &STR(&SYSLC(&SYSNSUB(1,&C))) THEN DO
    SET &RP = &MAXRP
    SET &RN = &MAXRP
   END
   ELSE DO
    SET &RN = &RN + 1
    ISPEXEC VGET (RV&RN)
    SET &C = &STR(&SYSNSUB(2,&&RV&RN))
   END
  END
  IF &RP < &MAXRP THEN DO
   SET &&RV&RN = +
   &SYSLC(&STR(&SYSNSUB(1,&CC)&ATSIGN&SYSNSUB(1,&DM)))
   ISPEXEC VPUT (RV&RN)
   SET &RP = &MAXRP
   SET &FROMNO = &FROMNO + 1
   /* first From: is already default receiver                   */
   IF &FROMNO = 1 AND &STR(&SEARCH) = &STR(&PRIMRECV) THEN DO
    SET &RETCODE = Ø
   END
   IF &FROMNO > 1 OR &STR(&SEARCH) NE &STR(&PRIMRECV) THEN DO
    IF &SYSLC(&STR(&SYSNSUB(1,&DM))) NE &SYSLC(&STR(&HOSTNAME)) +
    OR &SYSLC(&STR(&SYSNSUB(1,&CC))) NE &SYSLC(&STR(&SYSUID)) +
    THEN DO
     SET &MEMOCC = &STR(&SYSNSUB(1,&CC))
     SET &CCDM = &STR(&SYSNSUB(1,&DM))
     ISPEXEC VPUT (MEMOCC,CCDM)
     %ZCC CALLER(&SYSICMD) DEBUG(&STR(&DEBUG))
     IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
      WRITE =======> Reentering &SYSICMD <======
          END
          SET &MEMOCC = &STR()
          SET &CCDM = &STR()
          ISPEXEC VPUT (MEMOCC,CCDM)
          SET &RETCODE = Ø
         END
        END
       END
      END
     END
    END
    SET &P = &N + 1
    IF &STR(&SYSNSUB(2,&&A&P)) = &STR() THEN DO
     SET &N = &MAXCNT
    END
   END
  END
 END
 RETURN CODE(&RETCODE)
END
  /*                                                                  */
```

The CLIST ZSMTPC (edit macro) to syntax check the mail according to RFC standards:

```
 /*                                                                   */
 /*  ZSMTPC
 /*  Syntax check smtp mail and find no of bytes edit macro.
 /*   Called from MAILSENS
 /*
 /*   Utilities used:
 /*    SLEEP
 /*                                                                   */
PROC Ø DEBUG(nEBUG)
CONTROL NOMSG NOFLUSH NOLIST NOCONLIST NOSYMLIST NOCAPS
ERROR DO
 SET &RET = &LASTCC
 RETURN
END
SET &RET = Ø
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
 WRITE ======> Entering &SYSICMD <======
END
IF &SYSISPF = &STR(NOT ACTIVE) THEN DO
 WRITE ====> Sorry only executable under ISPF (&SYSICMD).
 EXIT CODE(16)
END
IF &SYSNEST = NO THEN DO
 ISREDIT MACRO PROCESS
END
```

```
ISPEXEC CONTROL ERRORS RETURN
ISPEXEC VGET (ZSCREEN)
ISREDIT (SAVE) = USER_STATE
ISREDIT (CHANGED) = DATA_CHANGED
ISREDIT (LRECL) = LRECL
ISREDIT BOUNDS = 1 &LRECL
ISREDIT RECOVER OFF
ISPEXEC CONTROL DISPLAY LINE START(14)
SLEEP 1
ISREDIT CURSOR = 1 0
SET &SUBJTX = &STR(Subject:)
SET &LSUBJTX = &LENGTH(&STR(SUBJECT:))
SET &MAXCC = 0
SET &SYNTAX = &STR()
SET &RET = 0
ISREDIT SEEK '&SUBJTX' 1
IF &RET = 0 THEN DO
 ISREDIT (SUBJR,SUBJC) = CURSOR
 SET &SMAXCC = &MAXCC
 SET &RET = 0
 ISREDIT LABEL &SUBJR = .SUBJ 1
 IF &RET > 8 THEN DO     /* allow replace of label */
  IF &RET > &SMAXCC THEN DO
   SET &SMAXCC = &RET
  END
 END
 SET &MAXCC = &SMAXCC
 SET &LASTLBL = &STR(.SUBJ)
END
ELSE DO
 WRITE ====> &SUBJTX    MISSING (&SYSICMD).
 SET &LASTLBL = &STR(.ZL)
END
ISREDIT CURSOR = 1 0
SET &RET = 0
ISREDIT SEEK 'HELO ' .ZF &STR(&LASTLBL)
IF &RET NE 0 THEN DO
 WRITE ====> Helo        MISSING (&SYSICMD).
END
SET &RET = 0
ISREDIT SEEK 'MAIL FROM:<' .ZF &STR(&LASTLBL)
IF &RET NE 0 THEN DO
 WRITE ====> Mail from:  MISSING (&SYSICMD).
END
SET &RET = 0
ISREDIT SEEK 'RCPT TO:<' .ZF &STR(&LASTLBL)
IF &RET NE 0 THEN DO
 WRITE ====> Rcpt to:    MISSING (&SYSICMD).
END
SET &RET = 0
ISREDIT SEEK 'DATA     ' .ZF &STR(&LASTLBL)
```

```
IF &RET NE Ø THEN DO
 WRITE ====> Data        MISSING (&SYSICMD).
END
SET &RET = Ø
ISREDIT SEEK 'DATE: ' .ZF &STR(&LASTLBL)
IF &RET NE Ø THEN DO
 WRITE ====> Date:       MISSING (&SYSICMD).
END
SET &RET = Ø
ISREDIT SEEK 'FROM: ' .ZF &STR(&LASTLBL)
IF &RET NE Ø THEN DO
 ISREDIT CURSOR = 1 Ø
 SET &RET = Ø
 ISREDIT SEEK 'SENDER: ' .ZF &STR(&LASTLBL)
 IF &RET NE Ø THEN DO
  WRITE ====> From:       MISSING or (&SYSICMD)
  WRITE ====> Sender:     MISSING (&SYSICMD).
 END
END
SET &RET = Ø
ISREDIT SEEK 'TO: ' .ZF &STR(&LASTLBL)
IF &RET NE Ø THEN DO
 WRITE ====> To:         MISSING (&SYSICMD).
END
IF &MAXCC > Ø THEN DO
 WRITE ====> Mail is not syntactically correct, mail is not sent +
 (&SYSICMD).
 SET &SYNTAX = ERROR
END
ISREDIT CURSOR = 1 Ø
SET &BHITS = Ø
SET &NBHITS = Ø
SET &RET = Ø
ISREDIT SEEK ' ' ALL
ISREDIT (BHITS,BHL) = SEEK_COUNTS
ISREDIT CURSOR = 1 Ø
ISREDIT SEEK P'^' ALL
ISREDIT (NBHITS,NBHL) = SEEK_COUNTS
SET &BHITS = &BHITS
SET &NBHITS = &NBHITS
ISPEXEC VPUT (BHITS,BHL)
ISPEXEC VPUT (NBHITS,NBHL)
ISPEXEC VPUT (SYNTAX)
ISPEXEC VGET (QUIET)
IF &STR(&QUIET) NE YES THEN DO
 WRITE BLANKS     = &BHITS Bytes
 WRITE NON-BLANKS = &NBHITS Bytes
 WRITE TOTAL      = &EVAL(&NBHITS + &BHITS) Bytes
END
ISREDIT USER_STATE = (SAVE)
IF &CHANGED NE YES THEN DO
```

```
 ISREDIT END
END
IF &STR(&QUIET) NE YES THEN DO
 EXIT CODE(&MAXCC)
END
ELSE DO
 EXIT CODE(1)
END
 /*                                                                   */
```

CLIST ZREPLY (edit macro) to construct a reply with receivers in accordance with RFC standards:

```
 /*                                                                   */
 /* ZREPLY
 /*  Edit macro to mail reply from current edit-dataset to SMTP;
 /*  this assumes that mail is syntactically correct received mail.
 /*  Reply will be saved into mailbox with id REPLY.
 /*   Called directly as Edit-macro.
 /*
 /*   Subroutines/edit macroes:
 /*    %COMPLIB
 /*    %MAILSENS
 /*
 /*   Utilities used:
 /*    SLEEP
 /*                                                                   */
PROC Ø DEBUG(nEBUG)
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
SET &RET = Ø
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
 WRITE ======> Entering &SYSICMD <======
END
IF &FLUSH = FLUSH THEN DO
 SET &ZEDSMSG = &str(Function interrupted)
 ISPEXEC SETMSG MSG(ISRZØØ1)
 EXIT CODE(Ø)
END
IF &SYSISPF = &STR(NOT ACTIVE) THEN DO
 WRITE ====> Sorry only executable under ISPF (&SYSICMD).
 EXIT CODE(16)
END
```

```
IF &SYSNEST = NO THEN DO
 ISREDIT MACRO PROCESS
END
ISPEXEC CONTROL ERRORS RETURN
ISPEXEC VGET (ZSCREEN)
ISREDIT (SAVE) = USER_STATE
ISREDIT (CHANGED) = DATA_CHANGED
ISREDIT (MEMBER) = MEMBER
ISREDIT (DSNAME) = DATASET
SET &ATSIGN = &STR(@)
SET &DS = &STR(&DSNAME)
IF &MEMBER NE &STR() THEN DO
 SET &DS = &STR(&DSNAME(&MEMBER))
END
ISPEXEC CONTROL DISPLAY LINE START(14)
SLEEP 1
SET &SMAXCC = &MAXCC
SET &PRIMRECV = &STR(Reply-to:)
SET &RC = &STR()
SET &DOMAIN = &STR()
SET &RET = 0
  /* return RC and DOMAIN (therefore indicated without ampersand */
SYSCALL SCANFR &PRIMRECV RC DOMAIN &ATSIGN DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
SET &RET = &LASTCC
SET &REPLYTOCC = &RET
SET &MAXCC = &SMAXCC
IF &REPLYTOCC > 0 THEN DO
 SET &PRIMRECV = &STR(From:)
 SET &RC = &STR()            /* clear for call to subroutine return */
 SET &DOMAIN = &STR()        /* clear for call to subroutine return */
 SET &RET = 0
   /* return RC and DOMAIN (therefore indicated without ampersand */
 SYSCALL SCANFR &PRIMRECV RC DOMAIN &ATSIGN DEBUG(&STR(&DEBUG))
   /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
 SET &RET = &LASTCC
 SET &FROMCC = &RET
 SET &MAXCC = &SMAXCC
 IF &FROMCC NE 0 THEN DO
  SET &PRIMRECV = &STR(Sender:)
  SET &RC = &STR()
  SET &DOMAIN = &STR()
  SET &RET = 0
   /* return RC and DOMAIN (therefore indicated without ampersand */
  SYSCALL SCANFR &PRIMRECV RC DOMAIN &ATSIGN DEBUG(&STR(&DEBUG))
  /* SYSCALL RETURN CODE DOES NOT TAKE ERROR ROUTINE */
  SET &RET = &LASTCC
  SET &SENDERCC = &RET
  SET &MAXCC = &SMAXCC
  IF &SENDERCC NE 0 THEN DO
   WRITE ====> Error: Receiver missing; current dataset not mailed +
   (&SYSICMD).
   SET &MAXCC = 8
  END
 END
END
IF &STR(&SYSNSUB(1,&RC)) = &STR() OR +
&STR(&SYSNSUB(1,&DOMAIN)) = &STR() THEN DO
 SET &MAXCC = 16
END
IF &MAXCC > Ø THEN DO
 SET &PRIMRECV = &STR()
END
ISPEXEC VPUT (PRIMRECV)    /* Info to ZSMTPN */
ISREDIT CURSOR = 1 Ø
SET &SUBJTX = &STR(Subject:)
SET &LSUBJTX = &LENGTH(&STR(SUBJECT:))
SET &RET = Ø
ISREDIT SEEK '&SUBJTX' 1
IF &RET = Ø THEN DO
 ISREDIT (SUBJR,SUBJC) = CURSOR
 SET &SMAXCC = &MAXCC
 SET &RET = Ø
 ISREDIT LABEL &SUBJR = .SUBJ 1
 IF &RET > 8 THEN DO     /* allow replace of label */
  IF &RET > &SMAXCC THEN DO
   SET &SMAXCC = &RET
  END
 END
 SET &MAXCC = &SMAXCC
 ISREDIT (NAMBEGIN,NAMFROW) = CURSOR
 ISREDIT (NAMFREC) = LINE &STR(&SYSNSUB(1,&NAMBEGIN))
 SET &LENNAM = &LENGTH(&STR(&SYSNSUB(1,&NAMFREC)))
 SET &SYSDVAL = +
 &SUBSTR(&NAMFROW+&LSUBJTX:&LENNAM,&STR(&SYSNSUB(1,&NAMFREC)))
 SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
 READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8 &A9
 SET &SU = &STR(&SYSNSUB(1,&A1) &SYSNSUB(1,&A2) &SYSNSUB(1,&A3) +
 &SYSNSUB(1,&A4) &SYSNSUB(1,&A5) &SYSNSUB(1,&A6) &SYSNSUB(1,&A7) +
 &SYSNSUB(1,&A8) &SYSNSUB(1,&A9))
 IF &MAXCC = Ø AND &STR(&MEMBER) NE &STR() THEN DO
  IF &CHANGED = YES THEN DO
   SET &SMAXCC = &MAXCC
   SET &RET = Ø
   ISPEXEC LMINIT DATAID(DID) DATASET('&STR(&DSNAME)') ENQ(SHRW)
   ISPEXEC LMOPEN DATAID(&DID) OPTION(INPUT)
   SET &LMRET = &RET
   IF &LMRET = Ø THEN DO
    SET &RET = Ø
    ISPEXEC LMMLIST DATAID(&DID) STATS(YES) MEMBER(MEMBER) OPTION(LIST)
    SET &RLVERS = &STR(&ZLVERS)
    SET &RLMOD = &STR(&ZLMOD)
    SET &RLMDATE = &STR(&ZLMDATE)
    SET &RLMTIME = &STR(&ZLMTIME)
    SET &RLUSER = &STR(&ZLUSER)
   END
   SET &LMRET = &RET
   IF &LMRET > 8 THEN DO        /* ACCEPT MAX RC8 FROM LMMLIST        */
    SLEEP 1
    WRITE &STR(===>) Error &LMRET LMMLIST (&SYSICMD).
   END
   SET &MAXCC = &SMAXCC
   SET &RET = 0
   ISREDIT SAVE
   IF &RET = 20 THEN DO
    %COMPLIB &STR('&DSNAME') SHR
    SET &MAXCC = 0
    IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
     WRITE ======> Reentering &SYSICMD <======
    END
    /* TRY TO SAVE ONCE MORE                                          */
    ISREDIT SAVE
   END
   SET &SMAXCC = &MAXCC
   IF &STR(&RLUSER) = RECEIVE OR &STR(&RLUSER) = MAILED OR +
   &STR(&RLUSER) = REPLY OR &STR(&RLUSER) = SAVED THEN DO
    /* SAVE OF STATISTICS DURING EDIT WORKS ONLY ON ISPF V.4          */
    ISPEXEC LMMSTATS DATAID(&DID) MEMBER(&NRSTR(&MEMBER)) +
    VERSION(&RLVERS) MODDATE(&STR(&RLMDATE)) +
    MODTIME(&STR(&RLMTIME)) USER(&RLUSER)
   END
   ISPEXEC LMMLIST DATAID(&DID) OPTION(FREE)
   ISPEXEC LMCLOSE DATAID(&DID)
   ISPEXEC LMFREE  DATAID(&DID)
   SET &MAXCC = &SMAXCC
  END
 END
 IF &MAXCC = 0 AND &STR(&MEMBER) = &STR() THEN DO
  IF &CHANGED = YES THEN DO
   SET &RET = 0
   ISREDIT SAVE
  END
 END
 IF &MAXCC = 0 THEN DO
  ISPEXEC VPUT (SU)     /* keep case via ispf variable */
  %MAILSENS RC('&STR(&SYSNSUB(1,&RC))') +
  &STR(&SYSNSUB(1,&DOMAIN)) REMAIL(REPLY) +
  ID('&STR(&SYSNSUB(1,&SU))') +
  DS(&STR(&DS)) DEBUG(&STR(&DEBUG))
  SET &ID = &STR()
  ISPEXEC VPUT (SU)
  IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
   WRITE ======> Reentering &SYSICMD <======
  END
  ISPEXEC VGET (SENDSUP)
  ISREDIT USER_STATE = (SAVE)
  IF &STR(&SENDSUP) NE YES THEN DO
   ISREDIT END       /* TERMINATE EDIT OF ORIGINAL MAIL WHEN REPLIED */
  END
  EXIT CODE(1)
 END
 ELSE DO
  WRITE ====> Error detected; current dataset not mailed (&SYSICMD).
  ISREDIT USER_STATE = (SAVE)
  EXIT CODE(1)
 END
END
ELSE DO
 WRITE ====> &SUBJTX     missing; current dataset not mailed (&SYSICMD).
 ISREDIT USER_STATE = (SAVE)
 EXIT CODE(1)
END
 /*                                                                   */
 /*  INLINE SUBROUTINES                                               */
 /*                                                                   */
SCANFR: +
PROC 4 PRIMRECV RC DOMAIN ATSIGN DEBUG(NEBUG)
 /*                                                                   */
 /* INLINE ROUTINE TO RETURN RC AND DOMAIN                            */
 /*                                                                   */
 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
 SET &RMAXCC = &MAXCC
 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
```

```
SET &DEBUG = NEBUG
ISPEXEC CONTROL ERRORS RETURN
SYSREF &RC &DOMAIN
SET &RETCODE = 8
SET &RC = &STR()
SET &DM = &STR()
SET &DOMAIN = &STR()
ISREDIT CURSOR = 1 Ø
SET &RET = Ø
ISREDIT SEEK &STR('______ Reply Separator ')
IF &RET = Ø THEN DO
 ISREDIT (SUBJR,SUBJC) = CURSOR
 SET &SUBJR = &SUBJR - 1
 ISREDIT LABEL &SUBJR = .RS 1
 SET &RS = .RS
END
ELSE DO
 SET &RS = .ZL
END
ISREDIT CURSOR = 1 Ø
SET &RCPTTX = &STR(&PRIMRECV)
SET &LRCPTTX = &LENGTH(&STR(&PRIMRECV))
SET &RET = Ø
ISREDIT SEEK '&RCPTTX' 1 .ZF &RS
IF &RET = Ø THEN DO
 ISREDIT (RCPBEGIN,RCPFROW) = CURSOR
 ISREDIT (RCPFREC) = LINE &STR(&SYSNSUB(1,&RCPBEGIN))
 SET &LENRCP = &LENGTH(&STR(&SYSNSUB(1,&RCPFREC)))
 SET &SYSDVAL = +
 &SUBSTR(&RCPFROW+&LRCPTTX:&LENRCP,&STR(&SYSNSUB(1,&RCPFREC)))
 SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
 READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8 &A9 &A1Ø &A11 &A12
 SET &MAXCNT = 12
 SET &N = Ø
 SET &A = A
 DO WHILE &N < &MAXCNT AND &DOMAIN = &STR()
  SET &N = &N + 1
  SET &LENRC = &LENGTH(&STR(&SYSNSUB(2,&&A&N)))
  IF &LENRC > Ø THEN DO
   IF &SUBSTR(1:1,&STR(&SYSNSUB(2,&&A&N))) = &STR(<) THEN DO
    SET &&A&N = &SUBSTR(2:&LENRC,&STR(&SYSNSUB(2,&&A&N)))
   END
   SET &LENRC = &LENGTH(&STR(&SYSNSUB(2,&&A&N)))
   IF &SUBSTR(&LENRC:&LENRC,&STR(&SYSNSUB(2,&&A&N))) = &STR(>) THEN DO
    SET &&A&N = &SUBSTR(1:&LENRC-1,&STR(&SYSNSUB(2,&&A&N)))
   END
   SET &LENRC = &LENGTH(&STR(&SYSNSUB(2,&&A&N)))
   SET &RC = &STR(&SYSNSUB(2,&&A&N))
   SET &SRCHSPC = &STR(&ATSIGN) /* look for at sign */
   SET &STARTSPC = 1
   SET &LOCSPC = +
    &SYSINDEX(&STR(&SRCHSPC),&STR(&SYSNSUB(1,&RC)),&STARTSPC)
    IF &LOCSPC > 0 THEN DO
     SET &RC = &SUBSTR(1:&LOCSPC-1,&STR(&SYSNSUB(2,&&A&N)))
     SET &DM = &SUBSTR(&LOCSPC+1:&LENRC,&STR(&SYSNSUB(2,&&A&N)))
     IF &STR(&SYSNSUB(1,&DM)) NE &STR() THEN DO
      SET &DOMAIN = DOMAIN('&STR(&SYSNSUB(1,&DM))')
      SET &N = &MAXCNT
      SET &RETCODE = 0
     END
    END
    ELSE DO
     IF &N = &MAXCNT THEN DO
      WRITE ====> Error: Domain missing for Receiver (&SYSICMD).
      SET &RETCODE = 16   /* DOMAIN MISSING */
     END
    END
   END
   ELSE DO
    SET &N = &MAXCNT
   END
  END
 END
 SET &MAXCC = &RMAXCC
 RETURN CODE(&RETCODE)
END
  /*                                                                  */
```

CLIST ZREMAIL (edit macro) to re-send already constructed mail:

```
 /*                                                                   */
 /* ZREMAIL
 /* Edit macro to mail current edit-dataset to SMTP in raw form without
 /*  editing; this assumes that mail is syntactically correct with
 /*  RFC header/address etc already filled in.
 /* Can be used to re-send mail-copy.
 /*  Called directly as Edit-macro.
 /*
 /*   Subroutines/edit macroes:
 /*    %COMPLIB
 /*    %MAILSENS
 /*
 /*   Utilities used:
 /*    SLEEP
 /*                                                                   */
PROC 0 DEBUG(nEBUG)
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
SET &RET = 0
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
 WRITE ======> Entering &SYSICMD <======
END
IF &FLUSH = FLUSH THEN DO
 SET &ZEDSMSG = &str(Function interrupted)
 ISPEXEC SETMSG MSG(ISRZ001)
 EXIT CODE(0)
END
IF &SYSISPF = &STR(NOT ACTIVE) THEN DO
 WRITE ====> Sorry only executable under ISPF (&SYSICMD).
 EXIT CODE(16)
END
IF &SYSNEST = NO THEN DO
 ISREDIT MACRO PROCESS
END
ISPEXEC CONTROL ERRORS RETURN
ISPEXEC VGET (ZSCREEN)
ISREDIT (SAVE) = USER_STATE
ISREDIT (CHANGED) = DATA_CHANGED
ISREDIT (MEMBER) = MEMBER
ISREDIT (DSNAME) = DATASET
SET &ATSIGN = &STR(@)
SET &DS = &STR(&DSNAME)
IF &MEMBER NE &STR() THEN DO
 SET &DS = &STR(&DSNAME(&MEMBER))
END
ISPEXEC CONTROL DISPLAY LINE START(14)
SLEEP 1
SET &RC = &STR(REMAIL)
SET &DM = &STR()
SET &DOMAIN = &STR()
ISREDIT CURSOR = 1 0
SET &RCPTTX = &STR(Rcpt to:)
SET &LRCPTTX = &LENGTH(&STR(Rcpt to:))
SET &RET = 0
ISREDIT SEEK '&RCPTTX' 1
IF &RET = 0 THEN DO
 ISREDIT (RCPBEGIN,RCPFROW) = CURSOR
 ISREDIT (RCPFREC) = LINE &STR(&SYSNSUB(1,&RCPBEGIN))
 SET &LENRCP = &LENGTH(&STR(&SYSNSUB(1,&RCPFREC)))
 SET &SYSDVAL = +
 &SUBSTR(&RCPFROW+&LRCPTTX:&LENRCP,&STR(&SYSNSUB(1,&RCPFREC)))
 SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
 READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8
 SET &LENRC = &LENGTH(&STR(&SYSNSUB(1,&A1)))
 IF &SUBSTR(1:1,&STR(&SYSNSUB(1,&A1))) = &STR(<) THEN DO
```

```
  SET &A1 = &SUBSTR(2:&LENRC,&STR(&SYSNSUB(1,&A1)))
 END
 SET &LENRC = &LENGTH(&STR(&SYSNSUB(1,&A1)))
 IF &SUBSTR(&LENRC:&LENRC,&STR(&SYSNSUB(1,&A1))) = &STR(>) THEN DO
  SET &A1 = &SUBSTR(1:&LENRC-1,&STR(&SYSNSUB(1,&A1)))
 END
 SET &LENRC = &LENGTH(&STR(&SYSNSUB(1,&A1)))
 SET &RC = &STR(&SYSNSUB(1,&A1))
 SET &SRCHSPC = &STR(&ATSIGN) /* look for at sign */
 SET &STARTSPC = 1
 SET &LOCSPC = &SYSINDEX(&STR(&SRCHSPC),&STR(&SYSNSUB(1,&RC)),&STARTSPC)
 IF &LOCSPC > 0 THEN DO
  SET &RC = &SUBSTR(1:&LOCSPC-1,&STR(&SYSNSUB(1,&A1)))
  SET &DM = &SUBSTR(&LOCSPC+1:&LENRC,&STR(&SYSNSUB(1,&A1)))
  IF &STR(&SYSNSUB(1,&DM)) NE &STR() THEN DO
   SET &DOMAIN = DOMAIN('&STR(&SYSNSUB(1,&DM))')
  END
 END
 ELSE DO
  SET &MAXCC = 16     /* DOMAIN MISSING */
  WRITE ====> Error: Domain missing; current dataset not mailed +
  (&SYSICMD).
 END
END
ELSE DO
 WRITE ====> Error: Receiver missing; current dataset not mailed +
 (&SYSICMD).
END
ISREDIT CURSOR = 1 0
SET &SUBJTX = &STR(Subject:)
SET &LSUBJTX = &LENGTH(&STR(SUBJECT:))
SET &RET = 0
ISREDIT SEEK '&SUBJTX' 1
IF &RET = 0 THEN DO
 ISREDIT (SUBJR,SUBJC) = CURSOR
 SET &SMAXCC = &MAXCC
 SET &RET = 0
 ISREDIT LABEL &SUBJR = .SUBJ 1
 IF &RET > 8 THEN DO     /* allow replace of label */
  IF &RET > &SMAXCC THEN DO
   SET &SMAXCC = &RET
  END
 END
 SET &MAXCC = &SMAXCC
 ISREDIT (NAMBEGIN,NAMFROW) = CURSOR
 ISREDIT (NAMFREC) = LINE &STR(&SYSNSUB(1,&NAMBEGIN))
 SET &LENNAM = &LENGTH(&STR(&SYSNSUB(1,&NAMFREC)))
 SET &SYSDVAL = +
 &SUBSTR(&NAMFROW+&LSUBJTX:&LENNAM,&STR(&SYSNSUB(1,&NAMFREC)))
 SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
```

```
READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8 &A9
SET &SU = &STR(&SYSNSUB(1,&A1) &SYSNSUB(1,&A2) &SYSNSUB(1,&A3) +
&SYSNSUB(1,&A4) &SYSNSUB(1,&A5) &SYSNSUB(1,&A6) &SYSNSUB(1,&A7) +
&SYSNSUB(1,&A8) &SYSNSUB(1,&A9))
IF &MAXCC = 0 AND &STR(&MEMBER) NE &STR() THEN DO
 IF &CHANGED = YES THEN DO
  SET &SMAXCC = &MAXCC
  SET &RET = 0
  ISPEXEC LMINIT DATAID(DID) DATASET('&STR(&DSNAME)') ENQ(SHRW)
  ISPEXEC LMOPEN DATAID(&DID) OPTION(INPUT)
  SET &LMRET = &RET
  IF &LMRET = 0 THEN DO
   SET &RET = 0
   ISPEXEC LMMLIST DATAID(&DID) STATS(YES) MEMBER(MEMBER) OPTION(LIST)
   SET &RLVERS = &STR(&ZLVERS)
   SET &RLMOD = &STR(&ZLMOD)
   SET &RLMDATE = &STR(&ZLMDATE)
   SET &RLMTIME = &STR(&ZLMTIME)
   SET &RLUSER = &STR(&ZLUSER)
  END
  SET &LMRET = &RET
  IF &LMRET > 8 THEN DO        /* ACCEPT MAX RC8 FROM LMMLIST        */
   SLEEP 1
   WRITE &STR(===>) Error &LMRET LMMLIST (&SYSICMD).
  END
  SET &MAXCC = &SMAXCC
  SET &RET = 0
  ISREDIT SAVE
  IF &RET = 20 THEN DO
   %COMPLIB &STR('&DSNAME') SHR
   SET &MAXCC = 0
   IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
    WRITE ======> Reentering &SYSICMD <======
   END
   /* TRY TO SAVE ONCE MORE                                          */
   ISREDIT SAVE
  END
  SET &SMAXCC = &MAXCC
  IF &STR(&RLUSER) = RECEIVE OR &STR(&RLUSER) = MAILED OR +
  &STR(&RLUSER) = REPLY OR &STR(&RLUSER) = SAVED THEN DO
   /* SAVE OF STATISTICS DURING EDIT WORKS ONLY ON ISPF V.4          */
   ISPEXEC LMMSTATS DATAID(&DID) MEMBER(&NRSTR(&MEMBER)) +
   VERSION(&RLVERS) MODDATE(&STR(&RLMDATE)) +
   MODTIME(&STR(&RLMTIME)) USER(&RLUSER)
  END
  ISPEXEC LMMLIST DATAID(&DID) OPTION(FREE)
  ISPEXEC LMCLOSE DATAID(&DID)
  ISPEXEC LMFREE  DATAID(&DID)
  SET &MAXCC = &SMAXCC
 END
 END
 IF &MAXCC = Ø AND &STR(&MEMBER) = &STR() THEN DO
  IF &CHANGED = YES THEN DO
   SET &RET = Ø
   ISREDIT SAVE
  END
 END
 IF &MAXCC = Ø THEN DO
  %MAILSENS RC('&STR(&SYSNSUB(1,&RC))') +
  &STR(&SYSNSUB(1,&DOMAIN)) +
  ID('&STR(&SYSNSUB(1,&SU))') REMAIL(YES) +
  DS(&STR(&DS)) DEBUG(&STR(&DEBUG))
  IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
   WRITE ======> Reentering &SYSICMD <======
  END
 END
 ELSE DO
  WRITE ====> Error detected; current dataset not mailed (&SYSICMD).
 END
 ISREDIT USER_STATE = (SAVE)
 EXIT CODE(1)
END
ELSE DO
 WRITE ====> &SUBJTX     missing; current dataset not mailed +
 (&SYSICMD).
 ISREDIT USER_STATE = (SAVE)
 EXIT CODE(1)
END
 /*                                                                   */
```

CLIST ZNOMAIL (edit macro) to suppress send of current mail:

```
 /*                                                                   */
 /* ZNOMAIL
 /*  Suppress mail of current edit dataset, as default save contents
 /*   of suppressed mail into mailbox with id SAVED.
 /*   Called directly as edit macro.
 /*
 /*  Parameter (optional) as macro parameter:
 /*   CANCEL: if indicated, mail will not be saved in mailbox;
 /*   CAN   : if indicated, mail will not be saved in mailbox.
 /*
 /*   Utilities used:
 /*    SLEEP
 /*                                                                   */
PROC Ø DEBUG(nEBUG)
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
SET &RET = 0
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
 WRITE ======> Entering &SYSICMD <======
END
IF &FLUSH = FLUSH THEN DO
 SET &ZEDSMSG = &str(Function interrupted)
 ISPEXEC SETMSG MSG(ISRZ001)
 EXIT CODE(0)
END
IF &SYSISPF = &STR(NOT ACTIVE) THEN DO
 WRITE ====> Sorry only executable under ISPF (&SYSICMD).
 EXIT CODE(16)
END
IF &SYSNEST = NO THEN DO
 ISREDIT MACRO (CANCEL) PROCESS
END
ISPEXEC CONTROL ERRORS RETURN
ISPEXEC VGET (ZSCREEN)
ISPEXEC VGET (FUNCTION,SEND)
IF &FUNCTION = MAIL THEN DO
 ISPEXEC CONTROL DISPLAY LINE START(14)
 SLEEP 1
 IF &SYSCAPS(&STR(&SYSNSUB(1,&CANCEL))) = CAN OR +
 &SYSCAPS(&STR(&SYSNSUB(1,&CANCEL))) = CANCEL THEN DO
  SET &SEND = CANCEL
  WRITE ====> Mailing of current mail suppressed.
  WRITE ====> Save of mail cancelled.
 END
 ELSE DO
  SET &SEND = NO
  WRITE ====> Mailing of current mail suppressed.
  WRITE ====> Mail will be saved into your &SYSUID..SMTP where you +
  can edit it
  WRITE ====> and mail it later using command ZREMAIL or ZREPLY.
 END
 ISPEXEC VPUT (SEND)
 ISREDIT END
END
ELSE DO
 ISPEXEC CONTROL DISPLAY LINE START(14)
 SLEEP 1
 WRITE ====> No suppression performed; environment is not "MAIL" +
 (&SYSICMD).
END
```

```
EXIT CODE(1)
  /*                                                                  */
```

CLIST ZMAIL (edit macro) send current dataset as mail:

```
 /*                                                                   */
 /* ZMAIL
 /* Edit macro to mail current edit-dataset to Smtp;
 /* address and other SMTP data is build.
 /*  Called directly as Edit-macro.
 /*
 /*  Parameters (optional) as macro parameters:
 /*   RECEIVER: If indicated will be used as Mail receiver.
 /*   DOMAIN  : If indicated will be used as Mail receiver's Domain.
 /*
 /*   SUBROUTINES/EDIT MACROES:
 /*    %COMPLIB
 /*    %MAILSENS
 /*   PANEL USED     : INST81A
 /*
 /*   Utilities used:
 /*    SLEEP
 /*                                                                   */
PROC 0 DEBUG(nEBUG)
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
SET &RET = 0
IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
 WRITE ======> Entering &SYSICMD <======
END
IF &FLUSH = FLUSH THEN DO
 SET &ZEDSMSG = &str(Function interrupted)
 ISPEXEC SETMSG MSG(ISRZ001)
 EXIT CODE(0)
END
IF &SYSISPF = &STR(NOT ACTIVE) THEN DO
 WRITE ====> Sorry only executable under ISPF (&SYSICMD).
 EXIT CODE(16)
END
IF &SYSNEST = NO THEN DO
 ISREDIT MACRO (RECEIVER,DOMAIN) PROCESS
END
```

```
ISPEXEC CONTROL ERRORS RETURN
ISPEXEC VGET (ZSCREEN)
ISREDIT (SAVE) = USER_STATE
ISREDIT (CHANGED) = DATA_CHANGED
ISREDIT (MEMBER) = MEMBER
ISREDIT (DSNAME) = DATASET
SET &DS = &STR(&DSNAME)
SET &SDEBUG = &DEBUG
IF &MEMBER NE &STR() THEN DO
 SET &DS = &STR(&DSNAME(&MEMBER))
END
SET &MEMOID = &STR()
ISPEXEC VPUT (MEMOID)
SET &MEMORC = &STR()
ISPEXEC VPUT (MEMORC)
SET &REMAIL = NO
ISREDIT CURSOR = 1 Ø
SET &SMAXCC = &MAXCC
SET &RET = Ø
ISREDIT SEEK ' ** MESSAGE ** ' .ZF .ZF
IF &RET = Ø THEN DO
 ISREDIT (JOBMSGL,JOBMSGR) = CURSOR
 ISREDIT DELETE &JOBMSGL               /* REMOVE HEADER */
 ISREDIT DELETE &JOBMSGL               /* REMOVE HEADER */
 ISREDIT (CHANGED) = DATA_CHANGED
END
SET &MAXCC = &SMAXCC
IF &MAXCC = Ø AND &STR(&MEMBER) NE &STR() THEN DO
 IF &CHANGED = YES THEN DO
  SET &SMAXCC = &MAXCC
  SET &RET = Ø
  ISPEXEC LMINIT DATAID(DID) DATASET('&STR(&DSNAME)') ENQ(SHRW)
  ISPEXEC LMOPEN DATAID(&DID) OPTION(INPUT)
  SET &LMRET = &RET
  IF &LMRET = Ø THEN DO
   SET &RET = Ø
   ISPEXEC LMMLIST DATAID(&DID) STATS(YES) MEMBER(MEMBER) OPTION(LIST)
   SET &RLVERS = &STR(&ZLVERS)
   SET &RLMOD = &STR(&ZLMOD)
   SET &RLMDATE = &STR(&ZLMDATE)
   SET &RLMTIME = &STR(&ZLMTIME)
   SET &RLUSER = &STR(&ZLUSER)
  END
  SET &LMRET = &RET
  IF &LMRET > 8 THEN DO        /* ACCEPT MAX RC8 FROM LMMLIST         */
   SLEEP 1
   WRITE &STR(===>) Error &LMRET LMMLIST (&SYSICMD).
  END
  SET &MAXCC = &SMAXCC
  SET &RET = Ø
  ISREDIT SAVE
  SET &SAVERET = &RET
  IF &RET = 4 THEN DO        /* NEW MEMBER */
   SET &RET = Ø
   SET &SAVERET = Ø
   SET &MAXCC = Ø
  END
  IF &SAVERET = 2Ø THEN DO
   %COMPLIB &STR('&DSNAME') SHR
   SET &MAXCC = Ø
   IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
    WRITE ======> Reentering &SYSICMD <======
   END
   /* TRY TO SAVE ONCE MORE                                           */
   ISREDIT SAVE
  END
  SET &SMAXCC = &MAXCC
  IF &STR(&RLUSER) = RECEIVE OR &STR(&RLUSER) = MAILED OR +
  &STR(&RLUSER) = REPLY OR &STR(&RLUSER) = SAVED THEN DO
    /* SAVE OF STATISTICS DURING EDIT WORKS ONLY ON ISPF V.4          */
   ISPEXEC LMMSTATS DATAID(&DID) MEMBER(&NRSTR(&MEMBER)) +
   VERSION(&RLVERS) MODDATE(&STR(&RLMDATE)) +
   MODTIME(&STR(&RLMTIME)) USER(&RLUSER)
  END
  ISPEXEC LMMLIST DATAID(&DID) OPTION(FREE)
  ISPEXEC LMCLOSE DATAID(&DID)
  ISPEXEC LMFREE  DATAID(&DID)
  SET &MAXCC = &SMAXCC
 END
END
IF &MAXCC = Ø AND &STR(&MEMBER) = &STR() THEN DO
 IF &CHANGED = YES THEN DO
  SET &RET = Ø
  ISREDIT SAVE
 END
END
IF &MAXCC = Ø THEN DO
 ISREDIT CURSOR = 1 Ø
 SET &SUBJTX = &STR(SUBJECT:)
 SET &LSUBJTX = &LENGTH(&STR(SUBJECT:))
 SET &MAXCC = Ø
 SET &SYNTAX = &STR()
 SET &RET = Ø
 ISREDIT SEEK '&SUBJTX' 1
 IF &RET = Ø THEN DO
  ISREDIT (SUBJR,SUBJC) = CURSOR
  SET &SMAXCC = &MAXCC
  SET &RET = Ø
  ISREDIT LABEL &SUBJR = .SUBJ 1
  IF &RET > 8 THEN DO     /* allow replace of label */
   IF &RET > &SMAXCC THEN DO
    SET &SMAXCC = &RET
   END
  END
  SET &MAXCC = &SMAXCC
  SET &LASTLBL = &STR(.SUBJ)
  ISREDIT (NAMBEGIN,NAMFROW) = CURSOR
  ISREDIT (NAMFREC) = LINE &STR(&SYSNSUB(1,&NAMBEGIN))
  SET &LENNAM = &LENGTH(&STR(&SYSNSUB(1,&NAMFREC)))
  SET &SYSDVAL = +
  &SUBSTR(&NAMFROW+&LSUBJTX:&LENNAM,&STR(&SYSNSUB(1,&NAMFREC)))
  SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
  READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8 &A9
  SET &SU = &STR(&SYSNSUB(1,&A1) &SYSNSUB(1,&A2) &SYSNSUB(1,&A3) +
  &SYSNSUB(1,&A4) &SYSNSUB(1,&A5) &SYSNSUB(1,&A6) &SYSNSUB(1,&A7) +
  &SYSNSUB(1,&A8) &SYSNSUB(1,&A9))
  SET &MEMOID = &STR(&SYSNSUB(1,&SU))
  ISPEXEC VPUT (MEMOID)
  SET &REMAIL = FORWARD
 END
 ELSE DO
  IF &STR(&SYSNSUB(1,&RECEIVER)) NE &STR() THEN DO
   SET &MEMOID = &STR(Noname)
  END
 END
 IF &STR(&SYSNSUB(1,&RECEIVER)) = &STR() THEN DO  /* no external recv */
  ISPEXEC DISPLAY PANEL(INST81A)
  IF &STR(&DEBUG) = &STR() THEN DO
   SET &DEBUG = &SDEBUG
  END
  ISPEXEC VGET (MEMOID,MEMORC,DM)
  ISPEXEC VGET (RESP)
  IF &RESP = CANCEL THEN DO
   SET &RESP = &STR()
   ISPEXEC VPUT (RESP)
   ISPEXEC CONTROL DISPLAY LINE START(14)
   SLEEP 1
   WRITE ====> Mailing Cancelled.
   ISREDIT USER_STATE = (SAVE)
   EXIT CODE(1)
  END
 END
 ELSE DO
  SET &MEMORC = &STR(&SYSNSUB(1,&RECEIVER))
  IF &STR(&SYSNSUB(1,&DOMAIN)) NE &STR() THEN DO
   SET &DM = DOMAIN('&STR(&SYSNSUB(1,&DOMAIN))')
  END
 END
 SET &SU = &STR(&SYSNSUB(1,&MEMOID))
 ISPEXEC VPUT (SU)     /* keep case via ispf variable */
```

```
 %MAILSENS RC('&STR(&SYSNSUB(1,&MEMORC))') +
 ID('&STR(&SYSNSUB(1,&MEMOID))') &STR(&SYSNSUB(1,&DM)) +
 REMAIL(&REMAIL) DS(&STR(&DS)) DEBUG(&STR(&DEBUG))
 SET &SU = &STR()
 ISPEXEC VPUT (SU)
 IF &SYSCAPS(&STR(&DEBUG)) = DEBUG THEN DO
  WRITE ======> Reentering &SYSICMD <======
 END
END
ELSE DO
 SET &ZEDLMSG = +
 &STR(=========> Execution could not be performed <==========)
 ISPEXEC SETMSG MSG(ISRZØØ1)
END
ISREDIT USER_STATE = (SAVE)
EXIT CODE(1)
 /*                                                                  */
```

CLIST TIMEOFFS gets time zone offset to GMT as defined in clock member in SYS1.PARMLIB:

```
 /*                                                                  */
 /*  TIMEOFFS
 /*   Return Time Offset from GMT as from Clock member in SYS1.PARMLIB
 /*    Called from MAILSENS
 /*
 /*    RC:  OFFSET IN ZHHMM
 /*          IF Z = Ø THEN ZONE IS EAST OF GMT
 /*          IF Z = 1 THEN ZONE IS WEST OF GMT
 /*          IF RC = 24**2-1 (MAX) THEN NO OFFSET CALCULATED
 /*
 /*   Utilities used:
 /*    IKJCT44B (tsoe clist built-in functions)
 /*                                                                  */
PROC Ø DEBUG(nEBUG)
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)) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
 WRITE ======> Entering &SYSICMD <======
END
IF &FLUSH = FLUSH THEN DO
 CLOSEFILE MNBVCZXZ
 FREE FI(MNBVCZXZ)
 SET &ZEDSMSG = &STR(Function interrupted)
 ISPEXEC SETMSG MSG(ISRZØØ1)
 EXIT CODE(Ø)
END
ISPEXEC VGET (ZSCREEN)
SET &OFFSET = 2**24-1
  /* ALLOW READ OF SYS1.PARMLIB IF UACC(NONE); BEGIN */
SET &PRIV  = &SYSXPRIVILEGED               /* clist built-in function */
NOBREAK
&SYSXPRIVILEGED(ON)                        /* clist built-in function */
  /* ALLOW READ OF SYS1.PARMLIB IF UACC(NONE); END */
ALLOC FI(MNBVCZXZ) DA('SYS1.PARMLIB(IEASYSØØ)') SHR REUSE
SET &RET = Ø
OPENFILE MNBVCZXZ INPUT
SET &OPENRET = &RET
  /* RESET READ OF SYS1.PARMLIB IF UACC(NONE); BEGIN */
&SYSXPRIVILEGED(&PRIV)                     /* clist built-in function */
BREAK
  /* RESET READ OF SYS1.PARMLIB IF UACC(NONE); END */
IF &OPENRET NE Ø THEN DO
 EXIT CODE(&OFFSET)
END
SET &MAXCNT = 256
SET &N = Ø
SET &RET = Ø
DO WHILE &RET NE 4ØØ AND &N < &MAXCNT
 SET &N = &N + 1
 SET &RET = Ø
 GETFILE MNBVCZXZ
 IF &RET NE 4ØØ THEN DO
  SET &SYSDVAL = &STR(&SYSNSUB(1,&MNBVCZXZ))
  SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
  READDVAL &B1 &B2 &B3 &B4 &B5 &B6 &B7 &B8
  SET &LENB1 = &LENGTH(&STR(&SYSNSUB(1,&B1)))
  SET &CLE = &STR(CLOCK=)
  SET &LENCLE = &LENGTH(&STR(&CLE))
  IF &LENB1 = &EVAL(&LENCLE+2) THEN DO
   IF &STR(&SUBSTR(1:&LENCLE,&SYSNSUB(1,&B1))) = &STR(&CLE) THEN DO
    SET &CLM = &STR(&SUBSTR(&LENCLE+1:&LENCLE+2,&SYSNSUB(1,&B1)))
    SET &N = &MAXCNT
   END
  END
 END
END
CLOSFILE MNBVCZXZ
FREE FI(MNBVCZXZ)
  /* ALLOW READ OF SYS1.PARMLIB IF UACC(NONE); BEGIN */
NOBREAK
SET &PRIV  = &SYSXPRIVILEGED               /* clist built-in function */
```

```
&SYSXTERMNOBREAK(ON)                       /* clist built-in function */
&SYSXPRIVILEGED(ON)                        /* clist built-in function */
  /* ALLOW READ OF SYS1.PARMLIB IF UACC(NONE); END */
ALLOC FI(MNBVCZXZ) DA('SYS1.PARMLIB(CLOCK&STR(&CLM))') SHR REUSE
SET &RET = 0
OPENFILE MNBVCZXZ INPUT
SET &OPENRET = &RET
  /* RESET READ OF SYS1.PARMLIB IF UACC(NONE); BEGIN */
&SYSXPRIVILEGED(&PRIV)                     /* clist built-in function */
BREAK
  /* RESET READ OF SYS1.PARMLIB IF UACC(NONE); END */
IF &OPENRET NE 0 THEN DO
 EXIT CODE(&OFFSET)
END
IF &STR(&CLM) NE &STR() THEN DO
 SET &MAXCNT = 256
 SET &N = 0
 SET &RET = 0
 DO WHILE &RET NE 400 AND &N < &MAXCNT
  SET &N = &N + 1
  SET &RET = 0
  GETFILE MNBVCZXZ
  IF &RET NE 400 THEN DO
   SET &SYSDVAL = &STR(&SYSNSUB(1,&MNBVCZXZ))
   SET &SYSDVAL = &STR(&SYSNSUB(1,&SYSDVAL))
   READDVAL &B1 &B2 &B3 &B4 &B5 &B6 &B7 &B8
   SET &LENZOF = &LENGTH(&STR(D.HH.MM.SS))
   IF &STR(&B1) = TIMEZONE THEN DO
    IF &LENZOF = &LENGTH(&STR(&B2)) THEN DO
     SET &OFFSET = &SUBSTR(3:4,&STR(&B2))&SUBSTR(6:7,&STR(&B2))
     SET &Z = &STR(+)
     IF &SUBSTR(1:1,&STR(&B2)) = W THEN DO
      SET &Z = &STR(-)
      SET &OFFSET = &OFFSET + 10000  /* SIMULATE WEST                 */
     END
     SET &N = &MAXCNT
    END
   END
  END
 END
END
CLOSEFILE MNBVCZXZ
FREE FI(MNBVCZXZ)
EXIT CODE(&OFFSET)
 /*                                                                   */
```

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

Nils Plum
Systems Programmer (Denmark)

ATM – the future

The ubiquitous qualities of ATM have been proven from the desktop to the backbone of a service provider’s network. Forecasts indicate that ATM will enjoy substantial growth in all aspects of local and wide area networking over the course of the next decade. But where, when, and how will ATM become deployed?

Specifications from the ATM Forum, ITU, and Bellcore have assisted greatly in the general acceptance of ATM, allowing interoperability and, therefore, multi-vendor competition. It is hoped that the momentum that has propelled ATM to date does not fall to the slow pace that so affected the sluggish implementation of X.25, Frame Relay, and ATM’s perceived precursor, narrow-band ISDN.

The one great advantage of ATM is in its inherent evolutionary approach. Users and service providers alike can merge existing technology onto ATM and migrate to higher bandwidth as and when the need arises. There is no big-bang replacement, forcing overnight changes to new infrastructure, rather a step-by-step approach of adding bandwidth and applications.

It is becoming apparent that, during 1999, service providers will introduce ATM-based services, providing advantages to the end-user community that include reduced-cost bandwidth on demand and full-managed services. Services will be in the manner of Virtual Private Managed Networks for Native LAN, Framed Data, Video, and Voice. Businesses that wish to retain their own private networks and deploy ATM for whatever reason will benefit from Native ATM services, purchasing the quality of service suited to their applications’ requirements.

Another target market for service providers is the home user, both in terms of entertainment (video-on-demand, home banking/shopping) and telecommuting. There will be much vying for market share between national PTTs and cable companies in this lucrative market, with debates about the most appropriate delivery mechanism – the telephone line or TV cable.

In the private business sector, ATM to the desktop will become a reality. Today, ATM interfaces are available on workgroup hubs, bridges, and routers. Bandwidth-hungry applications are more prevalent as the cost of PC/Workstation hardware becomes more affordable. The introduction of ATM to the desktop does not decry technology such as switched Ethernet, or 100 Mbps variants. What ATM introduces is selectable bandwidth and interoperability between multiple vendors' devices, allowing businesses choice.

It is obvious that, for ATM to become widely accepted, the cost must reduce. Today, ATM is affordable by sharing the cost between a number of users, for example a workgroup hub has multiple users connected via Ethernet and the ATM bandwidth is utilized between departments or buildings. Another example is to have an ATM interface on a server to reduce the bottleneck often experienced with shared LAN technology. The same consideration exists for service providers. To attract users/customers, the cost of delivery is critical. ATM cost per port will need to reduce significantly.

Chip manufacturers are already working vigorously to address cost. Off-the-shelf Physical, Switch Fabric, ATM Traffic Management, and Adaptation chipsets are beginning to arrive. This will allow ATM vendors to concentrate their efforts on differentiating products in software, while reducing research costs in hardware. One area of consideration is that off-the-shelf chips will be more likely to impact products where performance is not a key issue. Those requiring high performance, such as those used in service provider networks, will still require vendor-specific development.

It is not unreasonable, therefore, to expect ATM desktop connectivity to compete on price with today's Ethernet within 18 months. Similarly, the type of product used as access to the wide area network will change in terms of type and cost. ATM multiplexers, where all data is sent to the link, rather than having the additional ability of being locally switched, are now appearing and will deliver the kind of cost/ performance needed as an access device.

Although ATM will proliferate in areas where bandwidth is key to a business, or for purposes of delivering services, what of branch networking? It is not conceivable in the short term that a single bank branch, retail outlet, or small subsidiary office will need the bandwidth we associate with ATM. Another consideration is the efficiency of ATM. Its 10% overhead (before adaptation is taken into account), results in a sizeable loss of usage on a typical link of 64kbps.

The ATM Forum has addressed this area with the introduction of Framed UNI. Using a 99% link efficiency of Frame, the adaptation does not take place until it reaches a centrally located ATM switch. Frame UNI is to all intents and purposes a derivation of Frame Relay, with the added advantage of sharing the same signalling and management as native ATM. This level of integration does not exist with Frame Relay.

ATM will affect everyone. As a migratory technology ATM will first act as a delivery mechanism for existing services. The first profound effect will be the readily available services provided to the business and home user. As applications demand greater bandwidth, ATM will reach closer to the individual user, eventually to the PC/workstation. As the importance of information increases, the amount of bandwidth must also increase. Today, shared resources are proving key to reducing cost and increasing efficiency. Applications are emerging in such fields as medicine, education, and research, where centres of expertise are distributed among many participants. The Internet has opened the window of information to both the business and home user. It is conceivable to envisage multimedia on-line libraries, with video, audio, and transaction-based applications being available across the broad spectrum of the populace. As greater demand for information increases, ATM will be the key facilitator of the information age.

Dee Gibbs
GDC (UK)

TCP/SNA news

IBM has announced a number of improvements to its 2216 Multiaccess Connector Model 400, which provides System/370 and System/390 host access and WAN concentration for remote sites.

New features include Internet Key Exchange (IKE) for IP Security (IPSec), automating the usage of cryptographic keys, plus PPTP and L2F, giving remote users the ability to dial locally and tunnel across the IP network to the 2216.

There's now policy-based networking with LDAP client support, allowing users to retrieve policy information for IPSec, differentiated services, and IP filters from a central LDAP directory server.

TN3270 enhancements include host initiated dynamic Logical Unit (LU) definition, IBM Host On-Demand client cache, TN3270 sub-area SNA connectivity over IP, SDLC or X.25 networks, and improved transaction performance previously available in Multiprotocol Access Services (MAS) Version 3.2 PTF01.

There are also improvements to IP support, including DHCP services and SDLC enhancements.

For further information contact your local IBM representative.

* * *

Sane Solutions has announced Version 4.0 of NetTracker Proxy, which analyses proxy server and firewall log files to monitor corporate Web surfing patterns.

For further information contact:
Sane Solutions, 3 Preston Drive, North Kingstown, RI 02852-5519, USA.
Tel: (401) 295 4809.
Guildsoft, The Software Centre, East Way, Lee Mill Industrial Estate, Ivybridge, Devon, PL21 9EG, UK.
Tel\ (01752) 895100.
URL: http://www.sane.com.

* * *

OpenConnect Systems has announced Version 4.1 of its OC://WebConnect Pro software, its Web-to-host connectivity software.

The product allows 3270 and 5250 character-based mainframe application screens to be converted to graphical Web-like front ends.

The product also provides the ability to locally cache an applet, enhancing performance.

OpenConnect Systems are also shipping their OC://WebConnect Management Server, which is a centralized management and monitoring solution for Web-to-host server deployments.

For further information contact:
OpenConnect Systems, 2711 LBJ Frwy, Suite 800, Dallas, TX 75234-7324, USA.
Tel: (972) 484 5200.
URL: http://www.openconnect.com.

* * *

xephon