



# 38

# TCP/SNA

*June 2000*

---

## **In this issue**

- 3 Exploiting TCP/IP
  - 12 Picking the right tn3270(E) server
  - 22 NvDM send procedure
  - 54 E-business and Web-to-host integration
  - 66 Information point – reviews
  - 72 TCP/SNA news
- 

© Xephon plc 2000

update

# TCP/SNA Update

---

## Published by

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

## North American office

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

## Subscriptions and back-issues

A year's subscription to *TCP/SNA Update*, comprising four quarterly issues, costs £130.00 in the UK; \$190.00 in the USA and Canada; £136.00 in Europe; £142.00 in Australasia and Japan; and £140.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the March 1991 issue, are available separately to subscribers for £33.00 (\$48.00) each including postage.

## Editorial panel

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

## Editor

Fiona Hewitt

## 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.

## Contributions

Articles published in *TCP/SNA Update* are paid for at the rate of £170 (\$250) per 1000 words 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/tcpsnaupdate.html>; you will need the user-id from your address label.

---

© Xephon plc 2000. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

*Printed in England.*

# Exploiting TCP/IP

Although TCP/IP certainly makes things simpler, what you actually get out of it depends on the way in which you use it. This article considers the use of SMTP, FTP, and LPR, all of which are available under OS/390 eNetwork Communications Server Version 2 Release 5.

## FTP

The FTP file transfer facility enables the transfer of files or data between between two hosts. However, although many of us use FTP for transferring files over mainframes, it's not so commonly used to submit jobs over the mainframe.

For decades, job submission was possible only with the help of NJE/RJE; now, it can be achieved with TCP/IP. Gone are the days when you had to log on to the mainframe using 3270 emulation in order to submit jobs; now you can construct your JCL in Win95/Unix and submit it to the mainframe through FTP. (And it's easy to retrieve data using both FTP and SMTP too – see below.)

Let's look more closely at how to construct JCL and submit it to JES on the mainframe (server) from a Win95 machine (client). The simple IKJEFT01 program (TSO Batch) shown below involves the steps shown in Figure 1 (note that the commands are case-sensitive).

```
//USERIDT JOB (ACCTRP),'PROGINFO',CLASS=x,  
//          MSGCLASS=holdclass  
//STEP1 EXEC PGM=IKJEFT01  
//SYSTSPRT DD SYSOUT=*
```

<i>Steps</i>	<i>Description</i>	<i>Command</i>
1	Start FTP session	run FTP
2	Initiate a session with host(m/f)	open hostnm/ip
3	Login	enter userid/pw
4	Start talking with JES	quote SITE FILETYPE=JES
5	Submit the constructed JCL	put filename

*Figure 1: Steps for IKJEFT01*

```
//SYSTSIN DD *  
  H SEND  
/*
```

The response from JES should be something like

```
It is known to JES as JOBnnnnn
```

Note that you should prefix the jobname with your user-id. If you don't, the job will be submitted, but you won't be able to see it from the FTP session.

Enter the 'dir' command to get the list of spool files. The output will look like this:

```
xxxxxxT  JOBnnnnnn  OUTPUT    4 Spool Files
```

You can get the job output by issuing a single 'mget \*' command after setting 'prompt off'.

So far, our discussion has assumed that the mainframe is acting as the server and Win95/Unix as the client. However, it will also work the other way round. For example, you may need to get data from a Unix machine every day – assuming that your Unix machine runs FTP Server, you could schedule an FTP background job to get data every day and process it. See the *OS/390 eNetwork Communications Server* manual for more details on how to do this.

## SMTP

Once you've triggered a job using FTP, it's not difficult to get output back from the mainframe. One of the many joys inherent in any job is getting a report via e-mail, and it still seems astonishing to be able to get one from the mainframe – but you can indeed send and receive e-mail using mainframe TCP/IP. There is a task SMTP running under OS/390 for this purpose, and the IBM-supplied utility SMTPNOTE can be customized to send e-mails over the mainframe. You might use TSO RECEIVE for receiving e-mails. For sending e-mails, it might be easiest to construct e-mail (SMTP) headers and send them along with the body of the mail in a background job.

We saw above how to submit a job using FTP; now let's find out how to route output to our e-mail box. Imagine you've got a generic

program which is being used to retrieve output from DB2 tables or flat files. You could feed data for this program into JCL constructed in Win95/Unix to get the result to your mail box and just submit it.

Your JCL steps might look like this:

```
//USERIDx JOB (ACCTRP), 'PROGINFO', CLASS=x,
//          MSGCLASS=x, NOTIFY=&SYSUID
//PROC DATA EXEC PGM=GENUTIL
//SYS PRINT DD SYSOUT=*
//PROCIN DD *, DLM=ZZ
xxxxxxx
...
ZZ
//REPDATA DD DSN=&&RPDATA, DISP=(,PASS), LRECL=80, DSORG=PS
//          SPACE=(TRK,(5,5))"
//MAILRPT EXEC PGM=IEBGENER
//SYSUT2 DD SYSOUT=(B,SMTP)
//SYS PRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD *, DLM=YY
HELO jesnode
MAIL FROM:<userid@domain name>
RCPT TO:<your smtp mail box id>
DATA
FROM:      userid@domain name
TO : your smtp mail box id
SUBJECT: Processed Report
Date:
YY
//          DD DSN=&&RPDATA, (OLD,PASS)
//*
```

This is a two-step JCL. The first step processes the data. PROCIN could be used as a parameter list for GENUTIL (a user-written generic utility designed to fetch data from DB2 or flat files based on the parameter it receives) and produces a report on DD REPDATA. This DD is passed on to the next step, where IEBGENER uses it as the body of the mail. The e-mail headers in IEBGENER SYSUT1 DD are the standard headers used for any SMTP mail. You could improve things by sending it as an attachment using the MIME format. (Note that this no longer needs to be a two-step process; it could be reduced to a single-step process if your GENUTIL program produces e-mail headers, in which case REPDATA DD should be routed to SMTP.)

The Assembler code below (an example of a single-step process)

produces the e-mail header and body (it finds UCB and free space information for the volumes passed on to it, and also marks the IPL volume) and routes it to SMTP. Note that the e-mail addresses are hard-coded in the Assembler code. You may pass this information as in-stream data for Assembler and read it, but it's application-specific and it's up to you to decide how you want to use this.

## LPR

TCP/IP (now called Communications Server) for OS/390 offers both client and server support for remote printing, and you can route spool datasets to LPD (line printer daemon) running on a Unix machine.

The LPR command can be used to print a dataset. The IP address of a printer's host machine and printer name can be supplied with this command to route the output. Alternatively, you may want to set the default printer using the LPRSET command, which routes all the output printed using LPR to a printer@IP. (Note that all of these can be protected by RACF or security products, so you will need authority to use these facilities.)

Finally, you could even go one step further, and write a CGI routine in REXX/simple HTML and have it browsed using Internet Explorer or Netscape Navigator – but that's beyond the scope of this article.

## SOURCE

```
FRSPCXP  CSECT
*****
*** Finds out UCB and free space for the volumes passed on to it and
*** emails it to the email address (hard coded)
***      (see EMLHDRn data constant declarations )
*** Volumes must be concatenated and passed as a parameter like this
***      PGM=FRSPCXP,PARM='SYSRESSYSDLBSYSC01USR001'
***
***      Sridhar M N  snmanivel@hotmail.com
*****
*START OF MAIN
*****
      SAVE (14,12)          LINKAGE
      BALR  R12,0
      USING *,R12
      LA   R2,SAVEAREA
      ST   R2,8(,R13)
```

```

        ST    R13,SAVEAREA+4
        LR    R13,R2
*
        BAL   R7,PARMCHK
*
        OPEN (OUTFL,OUTPUT)
*
        BAL   R7,PROCEML
*
        MVC   LINEOUT(70),HYPHENS
        MVC   LINEOUT+70(10),=10CL1' '
        PUT   OUTFL,LINEOUT,80
        MVC   LINEOUT(L'HEADR),HEADR
        PUT   OUTFL,LINEOUT,80
        MVC   LINEOUT(70),HYPHENS
        PUT   OUTFL,LINEOUT,80
*
NEXTVOL  L    R8,PARMADR
        MVC   SRCHVOL(6),0(R8)
        BAL   R7,SUBMAIN
        L    R9,PARMLEN
        S    R9,=F'6'
        BC   B'1100',FOOTER
        LA   R8,6(R8)
        ST   R8,PARMADR
        ST   R9,PARMLEN
        B    NEXTVOL
*
FOOTER  DS    0H
        MVC   LINEOUT(70),HYPHENS
        PUT   OUTFL,LINEOUT,80
        MVC   LINEOUT(80),EMLFTR1
        PUT   OUTFL,LINEOUT,80
*
CLOSALL DS    0H
        CLOSE (OUTFL)
        L    R13,SAVEAREA+4
        RETURN (14,12),RC=0
*****
**END OF MAIN
*****
*SUBMAIN
SUBMAIN DS    0H
        BAL   R6,SCANUCB
        LTR   R15,R15
        BNZ   NOMATCH
        BAL   R6,GETSPC
        BAL   R6,PICKDATA
        BAL   R6,FMTLINE
        PUT   OUTFL,LINEOUT,80
        BR    R7

```

```

**ENDOF SUBMAIN *****
*
*
*****
***SUBR*****
*****
*
*SCANUCB
*
SCANUCB DS      0H
        XC      WAREA,WAREA
        UCBSCAN COPY,UCBAREA=MYUCB,WORKAREA=WAREA,VOLSER=SRCHVOL,
*
        DEVNCHAR=UCBN
        BR      R6
*ENDOF SCANUCB
*
*GETSPC
*
GETSPC DS      0H
        LSPACE DATA=FRDATA,UCB=ADDRUCB,MF=I
        BR      R6
*ENDOF GETSPC
*
PICKDATA DS      0H
        LA      R3,4           R3 INDEX FRDATA
        LA      R4,LINE       R4 INDEX 0/PFLD
        LA      R5,8           R5 COUNTER
NEXTFLD DS      0H           LOOP THRU FRDATA
        L        R1,FRDATA(R3)  LOAD THE FIELD
        CVD     R1,FRNO        CVD THE FIELD
        MVC     0(10,R4),=X'40202020212020404040' MOVE TEMPLT W/3SPC
        ED      0(7,R4),FRNO+5  EDIT 5 DIGIT CHAR
        LA      R4,10(R4)      SKIP 10 CHARS IN LINE
        LA      R3,4(R3)       SKIP TO NEXT ADDR
        BCT     R5,NEXTFLD     8 FIELDS BROWSD?
        BR      R6
*ENDOF PICKATA
*
*NOMATCH
*
NOMATCH DS      0H
        MVC     LINEOUT(80),=80CL1' '
        MVC     LINEOUT(L'SRCHVOL),SRCHVOL
        MVC     LINEOUT+8(L'MSGNFND),MSGNFND
        PUT     OUTFL,LINEOUT,80
        BR      R7
*
*
*FMTLINE
*

```



```

FMTLINE  DS      0H
MVC      LINEOUT(80),=80CL1' '
MVC      DSECLN(80),LINE
MVC      FIELD1(L'UCBN),UCBN
MVC      FIELD1+4(5),=C'      '          CLEAR THE MARKER
BAL      R5,IPLCHK
MVC      FIELD2(L'SRCHVOL),SRCHVOL
MVC      FIELD3(10),FTOTCY
MVC      FIELD4(10),FADFTRK
MVC      FIELD5(10),FEXT
MVC      FIELD6(10),FLARCYL
MVC      FIELD7(10),FADTRKL
MVC      LINEOUT(80),REORDER
BR       R6

*
*IPLCHK
*
IPLCHK   DS      0H
L        R2,16
L        R2,X'30'(R2)
MVC      IVSER(6),X'1C'(R2)
CLC      SRCHVOL(6),IVSER
BNZR    R5          DON'T MARK THIS LINE
MVC      FIELD1+4(5),=C'(IPL)'
BR       R5

*
*PARMCHK
*
PARMCHK  DS      0H
L        R1,0(R1)
LA       R2,0
LH       R2,0(R1)
LTR      R2,R2
BZ       NOPARM
LA       R1,2(R1)
ST       R1,PARMADR
ST       R2,PARMLEN
BR       R7
NOPARM   DS      0H
* NO PARM PASSED, SKIP EMAIL MESSAGE
B        CLOSALL
*PARMCHK ENDS
*
*PROCEML
*
PROCEML  DS      0H
MVC      LINEOUT(80),EMLHDR1
PUT      OUTFL,LINEOUT,80
MVC      LINEOUT(80),EMLHDR2
PUT      OUTFL,LINEOUT,80
MVC      LINEOUT(80),EMLHDR3

```

```

PUT   OUTFL,LINEOUT,80
MVC   LINEOUT(80),EMLHDR4
PUT   OUTFL,LINEOUT,80
MVC   LINEOUT(80),EMLHDR5
PUT   OUTFL,LINEOUT,80
MVC   LINEOUT(80),EMLHDR6
PUT   OUTFL,LINEOUT,80
MVC   LINEOUT(80),EMLHDR7
PUT   OUTFL,LINEOUT,80
BR    R7

```

\*\*\*\*\*

\*\*\* END OF SUBR

\*\*\*\*\*

```

                YREGS
SAVEAREA DS    18F
          CNOP  2,4
MYUCB    DS    CL48
ADDRUCB  DC    A(MYUCB)
UCBN     DS    CL4
FRDATA   DS    CL40
FRNO     DS    CL8
FREDIT   DS    CL9
LINE     DS    CL80' '
WAREA    DS    CL100
SRCHVOL  DS    CL6
LINEOUT  DS    CL80
MSGNFND  DC    CL50'DOES NOT EXIST OR ONLINE ON THIS SYSTEM'
DSECLN   DS    CL80
          ORG   DSECLN
FEXT     DS    CL10  NUMBER OF FREE EXTENTS
FTOTCY   DS    CL10  TOTAL FREE CYLINDERS
FADFTRK  DS    CL10  TOTAL ADDITIONAL FREE TRACKS
FLARCYL  DS    CL10  NUMBER OF CYLINDERS IN LARGEST FREE EXTENT
FADTRKL  DS    CL10  NUMBER OF ADDITIONAL TRACKS IN LARGEST FREE EXTENT
FDSCB    DS    CL10
FVIR     DS    CL10
FFRAGIX  DS    CL10
*
HEADR    DS    CL80
          ORG   HEADR
HEADR1   DC    CL10'UCB      '          ucb
HEADR2   DC    CL10'Volume  '          Volume
HEADR3   DC    CL10'Tot Cyl'          FTOTCY
HEADR4   DC    CL10'Add Trk'          FADFTRK
HEADR5   DC    CL10'Free Ext'          FEXT
HEADR6   DC    CL10'Larg.Cyl'          FLARCYL
HEADR7   DC    CL10'Add.fr.Tk'          FADTRKL
JUNK     DC    CL30' '
*
REORDER  DS    CL80
          ORG   REORDER

```

```

FIELD1  DC  CL10''
FIELD2  DC  CL10''
FIELD3  DC  CL10''
FIELD4  DC  CL10''
FIELD5  DC  CL10''
FIELD6  DC  CL10''
FIELD7  DC  CL10''
FIELD8  DC  CL10''
*
IVSER   DS  ØCL6
IVPRFX  DS  CL3
IVSFX   DS  CL3
*
PARMADR DS  F
PARMLEN DS  F
*
EMLHDR1 DC  CL80'HELO Jesnode'
EMLHDR2 DC  CL80'MAIL FROM:<sender@domain.name>'
EMLHDR3 DC  CL80'RCPT TO:<recp@recp.domain.name>'
EMLHDR4 DC  CL80'DATA'
EMLHDR5 DC  CL80'FROM:      userid@domain.com'
EMLHDR6 DC  CL80'TO :      recp@domain.com'
EMLHDR7 DC  CL80'SUBJECT: MAIL FROM MAINFRAME FOR VOLUMES REQUESTED'
EMLFTR1 DC  CL80'.'
*
HYPHENS DC  7ØCL1'*'
*
OUTFL   DCB DDNAME=OUTD1,LRECL=8Ø,RECFM=FB,DSORG=PS,
*
                MACRF=(PM)
                DSECT
                IEFUCBOB
                CVT DSECT=YES
                END

```

## TEST

Name: testit.txt

```

//useridF JOB (ACCT),'xxxx',
//      REGION=4M,MSGCLASS=T,CLASS=C,NOTIFY=&SYSUID
//COMASM EXEC PGM=FRSPCXP,PARM='XXXXXXXXXXXXXXXXXXXX'
//STEPLIB DD DSN=load.module.name,DISP=SHR
//SYSPRINT DD SYSOUT=*
//OUTD1   DD SYSOUT=(B,SMTP)
/*

```

---

*Sridhar Nellyappan Manivel*  
*MVS Systems Programmer*  
*working in an MVS installation team (USA)*

© Xephon 2000

---

## Picking the right tn3270(E) server

Now that the SNA sub-area backbones that were once mission-critical are being replaced by TCP/IP-centric networks (whether intranets, b2b extranets, or the Internet), tn3270 is becoming the pivotal methodology for SNA access. Fortunately, tn3270 is now a 15-year old technology, and is not a newcomer to the SNA arena. However, recent changes in technology, as well as IBM's role as a provider of networking hardware, mean that the optimum ways for implementing and deploying scalable and secure tn3270 solutions need to be carefully re-evaluated.

Tn3270 is currently being used daily, in production mode, by over 12 million users, around the world, through a variety of schemes including full-function, 'fat-clients' (eg Attachmate's EXTRA!), browser-invoked, Java-based 'thin-clients' (eg IBM's Host On-Demand), and 'on-the-fly' 3270-to-HTML conversion (eg Eicon's Aviva Web-to-Host Server) where the HTML gateway gains host access via tn3270. However, current developments mean that the schemes being used to realize much of this access – especially the tn3270(E) Server implementations – may prove both problematic and non-strategic.

Tn3270 is a 'split-stack', client/server mechanism that is essentially a variant of a traditional, circa 1990, SNA PU-controller gateway such as Novell's NetWare for SAA. It works by having a TCP/IP-client that performs 3270 emulation, using standard 3270 datastream (including extended datastream for 7-colour and highlighting) – that is exchanged via TCP/IP – with a tn3270(E) server component. This server includes multiple full-function SNA nodes and considers a mainframe as a group of SNA Type 2 nodes (eg 3174s), each with one PU and up to 255 LUs. (The 'E' notation in the context of tn3270 refers to enhanced mode operation which supports printing, status notification, LU assignment by name, positive/negative responses, SYSREQ and ATTN key support, and SNA BIND transmittal.) The challenge being faced by MIS professionals today is to choose the right platform and software implementation for the tn3270(E) server component.

Three recent developments have had a profound impact on the future of tn3270(E) servers:

- The first development was the two-billion-dollar IBM-Cisco alliance, formulated in August 1999, whereby IBM relinquished its entire internetworking hardware business, including the 22xx bridge/router family, to Cisco.
- The second factor is the availability of the 2.6Gbps-capable OSA-Express adapter as a bundled-in 'no-charge' feature on the latest IBM System/390 mainframes. The OSA-Express eliminates at a stroke the 136Mbps maximum throughput 'bottleneck' of today's ESCON II-based channel-attach systems, and makes it possible to realize network-to-mainframe data transfer using a Gigabit Ethernet LAN at 2Gbps, full-duplex.
- The third and final factor is the emergence of Secure Sockets Layer (SSL) as the *de facto* industry standard for tn3270(E)-based networking – thanks primarily to IBM's support for end-to-end data encryption through SSL in its Communications Server family of products, including CS for OS/390, CS/NT, and CS for AIX.

These three factors together dictate the future direction of tn3270(E) servers in terms of:

- Scalability – ie the maximum number of concurrent sessions that can be supported.
- Security in terms of user authentication and end-to-end data encryption.
- Affordability – ie cost per concurrent session.
- Performance in terms of 'round-trip' response times.
- Availability and serviceability in terms of redundancy and fail-over.
- Minimization of complexity by streamlining the end-to-end configuration.

## INITIAL RECOMMENDATIONS

In general, the optimum solution for implementations that require 10,000 or more concurrent tn3270(E) sessions, with end-to-end security and high-throughput performance, is likely to be the tn3270(E) Server within IBM's CS for OS/390 – running on a mainframe whose interface to the TCP/IP network is via a Cisco router. (The lack, at present, of SSL security in Sterling Software's SOLVE:TCPaccess precludes it from being a contender, unless you're willing to implement end-to-end encryption via an alternative scheme such as Virtual Private Networking (VPN).) For optimum performance, the Cisco router would be connected to the mainframe via an OSA-Express adapter through a 100Mbps Fast Ethernet or 1,000Mbps Gigabit Ethernet LAN. Furthermore, leveraging the Cryptographic Coprocessor of System/390 mainframes can offset the significant processing overhead and performance penalty associated with 128- or 156-bit-based encryption.

Overall, given the large number of concurrent sessions involved, this mainframe-based configuration is also likely to be the most cost-effective. If parallel sysplex is exploited along with the Virtual Internet Protocol Address (VIPA) capability found in OS/390 Version 2 Release 5 onwards (which enables the same TCP/IP address to be used to transparently access dispersed instances of a TCP/IP service), this solution could have exceptional availability, resilience, and serviceability characteristics. The tn3270(E) servers available with Cisco's Channel Interface Processor (CIP) for Cisco 7500 family routers or the Channel Port Adapter (CPA) for Cisco 7200 family routers currently do not support SSL-based security. They could also prove more expensive if you require a lot of (over 16,000) concurrent sessions.

If your tn3270 requirements are more modest – say, under 2,000 concurrent sessions – then a three-tier solution with IBM's CS/NT on an NT server, CS for AIX on an RS/6000 machine, or Novell's NetWare for SAA 4 on a Novell Server may be more than adequate – especially given that all of these implementations support end-to-end SSL-based encryption for tn3270(E) interactions. The Cisco CIP and

CPA is also an option, at least from an affordability standpoint – though the lack of SSL support is still a problem. Many of today’s tn3270(E) implementations are three-tier based or use a Cisco CIP/CPA approach. Many do not use end-to-end encryption. This is acceptable if the traffic is restricted to a secure, in-house intranet – SNA was rarely, if ever, encrypted when used across private, SNA-only or bridge/router-based multiprotocol networks. On the other hand, sending mission-critical SNA traffic, complete with user-ids and passwords, in unencrypted ‘clear text’ across the Internet or on extranets is somewhat imprudent.

Given the wide disparity when it comes to the requirements, expectations, and constraints of specific tn3270 implementations, the above initial recommendations are obviously unlikely to be granular enough to address all instances. General rules of thumb are also too vague. The remainder of this article therefore describes the implications of the various issues involved, and shows how to factor the relative importance of these issues in order to determine the best long-term solution for a given scenario.

## THE SSL CONUNDRUM

The SSL protocol was developed by Netscape in 1995 and is now widely used to realize secure interactions across the Internet. Yet, despite being universally supported by all major commercial Web servers (including IBM’s WebSphere) as well as by Netscape Navigator and Microsoft’s Internet Explorer Web browsers, SSL is not a part of the official tn3270(E) standard as represented by Internet Engineering Task Force (IETF) Request for Comment (RFC) documents RFC 2355, RFC 1647, and RFC 1576. What’s more, SSL will never be part of the official standard. Instead, the IETF is working on a variant of SSL referred to as Transport Layer Security (TLS). TLS is specified in the informational RFC 2246. TLS is ‘backward’ compatible with the latest level of SSL (which is SSL 3.0).

Rather incongruously, the current tn3270(E) standard does not yet address end-to-end data encryption or user authentication. This reflects

the fact that this kind of security was not that important in SNA networking because SNA traffic was restricted to either secure private networks or hard-to-consistently-intercept X.25 or Frame Relay public packet switching networks. (IBM, however – to its credit – included end-to-end Data Encryption Standard (DES) based encryption for LU-LU sessions in the SNA architecture as far back as 1978. But although it's still available in ACF/VTAM, most peripheral devices never bothered to implement SNA LU-LU encryption.) When the IETF tn3270 working group gets around to endorsing a security mechanism for tn3270(E), probably towards the end of 2000, that security scheme will be TLS as opposed to SSL. This is already known and understood.

In September 1998, realizing that the lack of a 'standardized' method for ensuring tn3270(E) security was impacting Web-to-host deployment, IBM decided to take preemptive action by adding SSL security to the tn3270(E) server in Version 2 Release 6 of CS for OS/390. Given that its Host On-Demand Java 'tn' client could already work in SSL mode (albeit initially only with an IBM-provided, Java-based 're-director' component), IBM immediately had an elegant and compelling two-tier end-to-end solution; ie Host On-Demand talking tn3270(E) with SSL-negotiated end-to-end encryption to a mainframe-resident, highly-scalable tn3270(E) server. This mainframe-centric solution has since been enhanced with the ultra-fast network-to-mainframe communications made possible by OSA-Express and the performance-enhancing cryptographic coprocessor, and has now become the baseline for high-end, easily scalable, SSL-capable tn3270(E) server implementations.

Today, SSL support is available on the following tn3270(E) servers:

- IBM Communications Server family
- NetWare for SAA 4 (which was co-developed by IBM)
- Attachmate's SNA Gateway 2.0.

Microsoft's SNA Server and the Cisco CIP are therefore the only widely used tn3270(E) servers that do not support SSL. SSL support is available on a wide range of 'tn' clients including:



- IBM's Host On-Demand
- IBM's PComm
- Attachmate's Java and ActiveX viewers
- Eicon's Aviva for Java Version 2.0
- Hummingbird's HostExplorer Web 2.0
- Zephyr's Passport family
- Renex's BlueZone browser plug-in
- ResQNet's 'tn' applet.

The only major client vendor that did not endorse SSL was Wall Data (whose acquisition towards the end of last year by NetManage ended eighteen months of financial woes that effectively prevented it from responding to emerging technologies like SSL). WRQ, a relative newcomer to IBM host access, is the first and only vendor to support TLS-based security with regard to tn3270. Given that there are no tn3270(E) servers that support TLS, WRQ's scheme works only with a WRQ-supplied, Java-based 'Proxy TLS Server' that has to sit in front of a tn3270(E) server.

#### SSL NEGOTIATES BUT DOESN'T DO THE ENCRYPTION

One essential point to make is that neither SSL nor TLS actually perform the end-to-end encryption process. Rather, they are essentially 'hand-shake' protocols that negotiate a common encryption scheme between the server (eg Web server or now a tn3270(E) server) and a client (eg Web browser, or, in the case of tn3270(E) and tn3270(E), client as in Host On-Demand). In the early days, SSL supported only the commercial encryption schemes made available by RSA Data Security. Today, SSL can also be used with public domain encryption schemes such as 56-bit DES, 168-bit Triple DES, or 'AV4' which is a variant of RSA's RC4 scheme.

Before SSL starts the negotiation process between the server and the client to determine a mutually acceptable encryption scheme, it

performs a client/server authentication process – typically realized using digital certificates from trusted authorities that can confirm that the two entities involved are indeed who they claim to be. This digital certificate-based authentication, which has to be completed before the negotiation step can begin, is a relatively low-level process that does not involve the user having to enter a userid and password. SSL, however, does have an optional user authentication process which is realized using a user-entered userid and password. Since it's unacceptable to send userids and passwords in unencrypted form, this user authentication process takes place only after the client and server have agreed on an encryption scheme – and, moreover, have started to encrypt the data being send between them.

### GOING AROUND SSL

With 'cyber terrorism' so prevalent that it has now become a common household term, it's imprudent to transmit any kind of mainframe data, especially if it contains userids and passwords, in unencrypted form across any non-secure network – especially the Internet. SSL, primarily in its HyperText Transfer Protocol (HTTP) variant known as Secure HTTP (HTTPS), is by far the best known and most widely used security scheme on the Web. SSL, initially through 'mid-tier' servers like IBM's Host On-Demand 're-director' or OpenConnect Systems' OC://WebConnect Pro, was also the encryption scheme most favoured by Web-to-host solutions. Support for it in CS and NetWare for SAA 4 was the final icing on the cake in terms of approval, acceptability, and ready accessibility. However, although SSL is probably the most convenient, it's not the only way to realize end-to-end encryption for tn3270(E) transactions.

IP Security (IPSec), which can be implemented, relatively easily, on a point-to-point basis between Cisco routers (or between Cisco routers and OS/390 2.8), is an option – and is obviously what Cisco would unhesitatingly recommend. IPSec, in reality, is typically realized as a part of an overall VPN scheme that enables private, corporate-specific IP traffic to be securely tunnelled across the Internet. VPN support is available on Cisco routers and in OS/390 2.5 (onwards).

Although VPN (or, for that matter, IPSec) could be implemented on an end-to-end client-to-host basis, this is typically too cumbersome and impractical. Instead, VPN (or IPSec) is best used on a campus-to-campus, office-to-headquarters, router-to-router or router-to-OS/390 basis. It is thus not suitable for travelling user, telecommuter, or field agent-to-host applications. It is certainly not feasible for browser-invoked, Java applet-based Web-to-host scenarios for home banking, on-line investing, personal travel reservation etc, where the general public is given access to mainframe applications. Tn3270(E) with SSL, or 3270-to-HTML conversion over HTTPS, is invariably the only way to adequately cater for this rapidly growing opportunity for public-to-mainframe, Web-to-host interactions. Unlike SSL, VPN and IPSec also do not offer userid/password-based user authentication as a standard, built-in option.

#### CHANNEL-ATTACH VERSUS MAINFRAME RESIDENT

The demise of the channel-attachable IBM 2216-400 Multiaccess Concentrator, with its tn3270(E) server and VPN capability, as a part of the IBM-Cisco alliance eliminates what was an attractive solution for a channel-attached, 'off-board' tn3270(E) server – albeit without SSL. The 2216-400's overriding virtue was affordability. In a bid to garner market share in the face of stiff competition from Cisco, IBM was extremely aggressive in setting the tn3270(E) server pricing on the 2216-400. It was possible to get a 2216-400, capable of supporting 15,000 concurrent tn3270(E) sessions, for around \$66,245, including the hardware and software. That was list. At that price, the cost per concurrent tn3270(E) user was just above \$4.00. Cisco's CIP and CPA prices are considerably higher. Now that the cost-compelling 2216-400 is not an option, the whole issue of 'off-board' tn3270(E) servers needs to be re-examined.

The performance, scalability, and high-availability of a mainframe solution, especially if deployed in conjunction with OSA-Express, the cryptographic coprocessor, and parallel sysplex, now tend to favour mainframe-resident solutions ahead of off-board solutions. Then, there's also the whole matter of standardizing on TCP/IP vis-à-vis the

mainframe. Right now, about 70% of MVS and OS/390 shops have TCP/IP on the mainframe. This number will probably get up to 90% in 2001. With TCP/IP on the mainframe and a mainframe-resident tn3270(E) server, all the traffic to and from the mainframe can be IP-based. (Enterprise Extender, ie HPR-over-IP, which is now available on Cisco routers as an alternative to Data Link Switching (DLSw), could be leveraged to accommodate any residual SNA traffic to and from legacy control units). Standardizing on IP for all traffic to and from the mainframe can simplify and streamline data centre operations and management while also marginally expediting overall throughput by eliminating the need to support disparate protocols across the network interface.

However, for those that want an interim solution or a small-scale implementation, there are some attractive, 'off-board', channel-attached solutions over and above the Cisco CIP/CPA. Channel-attachment options, either from IBM or Bus-Tech, are available for CS/NT, CS for AIX, and NetWare for SAA 4. All three of these solutions support SSL, with the latter being the obvious choice for those that have a Novell infrastructure and have been using NetWare for SAA as their SNA-LAN Gateway. Scalability is the primary difference between these solutions. NetWare for SAA 4 can support around 4,000 concurrent sessions on a single, 200MHz Pentium II with 384MB. It's safe to assume half this for a CS/NT implementation on a comparable server. Based on the power of the Unix box being used, CS for AIX should be able to accommodate 10,000 to 20,000 sessions.

## SUMMARY

The demand for tn3270(E)-based SNA-TCP/IP networking will continue to increase over the next couple of years as SNA subarea networks continue to be replaced by TCP/IP networks. Despite the cost-competitive 2216-400 becoming a casualty of the IBM-Cisco alliance, there are still very attractive mainframe and 'off-board' solutions for realizing scalable tn3270(E) servers. Security in the form of SSL is a pivotal and profound issue even though SSL *per se* is not part of the tn3270(E) standard.

The first choice for large installations should be a mainframe-resident solution, probably augmented with cryptographic coprocessor, OSA-Express, and parallel sysplex. The scalability, security, availability, performance, and serviceability of such a solution will be exceptional and unparalleled. The OSA-Express alone redefines the performance expectations for network-to-mainframe communications, while the cryptographic co-processor offsets the considerable processing required to deal with encrypted data. With IBM's CS family and Novell's NetWare for SAA 4 supporting SSL, smaller installations may be able to get by with a channel-attached, 'off-board' solution.

---

*Anura Gurugé*  
*Strategic Analyst (USA)*

© Xephon 2000

---

## **Free weekly news by e-mail**

Xephon has four weekly news services for enterprise systems users, covering the following subject areas:

- Data centre
- Distributed systems
- Networks
- Software

Each week, subscribers receive, by e-mail, a short news bulletin consisting of a list of items; each item has a link to the page on our Web site that contains the corresponding article. Each news bulletin also carries links to the main industry news stories of the week.

To subscribe to one or more of these news services, or review recent articles, point your browser at <http://www.xephon.com/news.htm>.

## NvDM send procedure

Most medium and large organizations need to distribute software between a number of centres. The various hosts are generally distributed over a number of sites, and they can belong to the same or a number of different networks. This kind of software distribution is particularly frequent in companies that produce applications software (eg software houses).

The NvDM SEND procedure presented here is based on NetView Distribution Manager, and is designed to make software distribution simple by transparently creating, executing, and controlling the NvDM utilities in order to manage the software shipment phases. It also supplies a historical log of all the main functions carried out.

The procedure is simple to use, and requires no knowledge of the rules or syntax of NetView Distribution Manager. For the software shipment phase, the user must select the datasets/files to be sent and their destinations (other NvDM nodes, ie other sub-area hosts); for the construction of the NvDM parameters, there is no need to know the file type and/or the dataset control block of the file.

The tool was developed and tested under OS/390 2.7.0 with NetView Distribution Manager 1.6.2

NvDM SEND PROCEDURE: REXX EXECs

### C-list NSPC000

```
/* REXX */
/* NvDM Send Procedure
   C-List NSPC000 - Called by ISPF user panel or user command
   Execute procedure for shipment of datasets between more OS/390 hosts
   The functions are: - Check user
                     - Allocation environment for execution Send Procedure
                     - If user enabled then run clist main NSPC000          */
Trace ?o
libpan = 'NVDM162.TOOLS.PANEL' ; libexe = 'NVDM162.TOOLS.CLIST'
/*          Insert here the user of projects          */
if userid() = NDMOPR1 & userid() = NDMADM ,
    & userid() = XXXXXXX & userid() = YYYYYYY
then do
```

```

say time() ' >>>'
say time() ' >>> *----- Distribution Software -----*'
say time() ' >>>          "NvDM Send Procedure"'
say time() ' >>>'
say time() ' >>>          USER NOT AUTHORIZED.'
say time() ' >>>' ; say time() ' >>>'
end
else do
address tso
'ALTLIB ACT APPLICATION(EXEC) DA(''libexe'') UNCOND'
"ISPEXEC LIBDEF ISPPLIB DATASET ID(''libpan'')'"
call nspc010
'ALTLIB DEACT APPLICATION(EXEC)'
ISPEXEC LIBDEF ISPPLIB DATASET
end
say time() ; say time() ' Send Procedure Software ended.' ; say time()
exit

```

## C-list NSPC010

```

/* REXX */
/* NvDM Send Procedure - C-List main NSPC010 Called by c-list NSPC000
Execute procedure for shipment of datasets between more OS/390 hosts
The functions are: - Verification if user is enable to use NvDM
functions;
- select NvDM destination and datasets name to send;
- prepare job batch for NvDM MAINTDAT function;
- prepare job batch for NvDM TCFMAINT function;
- All the jobs are submitted and controlled;
*/
trace ?o
/***** Variables main
*****/
ndes = Destination number
wexe = Environment with execs and tables of the procedure
      = Jcl for check NvDm user password
workj = Dataset of work-jobs
wjske = User library with job skeleton
tblnam = Table of project and destination
desnam = Table of NvDM destinations (applid NvDM)
wlsnd = List datasets to send (work dataset)
wjpac = Jcl for prepare package to send
wjpln = Jcl for define NvDM plan
wjrrr = Jcl for report of the NvDM Resource Repository
wjplo = Jcl for prepare LOAD to send
wtms = Timestamp for name of files to send
wjtyp = Type of NvDM utility (MAINTDAT/TCFMAINT/...)
wjout = Dataset name of output job utility
wlogs = Dataset name of log NvDM Send Procedure
*****/
ndes = 8 ; jobendmsg = 'OUTPUT QUEUE' ; wexe = 'NvDM162.TOOLS.CLIST'

```

```

/***** Verify on datasets of procedure work *****/
wjske = userid()||'.JCLLIB' ; ww="rx=SYSDSN('"'wjske"'"")
INTERPRET ww
if rx = 'OK' then do
  say '>>>' ; say '>>> Dataset status for 'wjske' is "'rx"'. To
control.'
  say '>>> User not enable.' ; say '>>>'
  return
end
nomej = wjske||'(NDMSKEL)' ; ww="rj=SYSDSN('"'nomej"'"")
INTERPRET ww
if rj = 'OK' then do
  say '>>> Job Skeleton status for 'nomej' is "'rj"'. To control.'
  say '>>> User not enable.'
  return
  end
ww="rk=SYSDSN('"'wexe"'"")
INTERPRET ww
if rk = 'OK' then do
  say '>>> Status of the dataset "'wexe"'" ; say '>>> is "'rk"'. To
control.'
  return
end
desnam = wexe||'(NSPT010)' ; ww="rq=SYSDSN('"'desnam"'"")
INTERPRET ww
if rq = 'OK' then do
  say '>>> Status of the dataset "'desnam"'" ; say '>>> is "'rq"'. To
control.'
  return
end
tblnam = wexe||'(NSPT000)' ; ww="rt=SYSDSN('"'tblnam"'"")
INTERPRET ww
if rt = 'OK' then do
  say '>>> Status of the dataset "'tblnam"'" ; say '>>> is "'rt"'. To
control.'
  return
end
wlogs = 'NVDM162.TOOLS.LOG' ; wlogs1 = wlogs||'1'
ww="rh1=SYSDSN('"'wlogs1"'"")
INTERPRET ww
if rh1 = 'OK' then do
  say '>>> Status of the dataset "'wlogs1"'" ; say '>>> is "'rh1"'. To
control.'
  return
end
wlogs2 = wlogs||'2' ; ww="rh2=SYSDSN('"'wlogs2"'"")
INTERPRET ww
if rh2 = 'OK' then do
  say '>>> Status of the dataset "'wlogs2"'" ; say '>>> is "'rh2"'. To
control.'
  return

```



```

end
/*****
/* Prefix dataset name for progressive number to plan name */
/* Read destination table */
*****/
pcount = 'NVDM162.TOOLS.PLANUM.'
"ALLOC DA('desnam') F(DES) SHR REUSE"
"EXECIO * DISKR DES (STEM DE. FINIS"
"FREE F(DES)"
i = 0
do a = 1 to de.0
  UPPER de.a
  if pos('LOCALNDM',de.a) = 0 then loc1 = word(de.a,3)
  c1 = left(strip(de.a),1)
  if c1 = '*' | c1 = ' ' then iterate
  i = i+1 ; des.i = de.a ; dcount.i = right(word(de.a,3),4)
end
if i = ndes then do
  say '>>>' ; say '>>> Destination number is incorrect in ''desnam''.'
  say '>>> There are 'i' instead 'ndes' destinations. To control.' ; say
  '>>>'
  return
end
do a = 1 to i
  if word(des.a,3) = loc1 then leave
  ncount = pcount||dcount.i ; ww="rp=SYSDSN(''ncount'')"
  INTERPRET ww
  if rp = 'OK' then do
    say '>>> Status of the dataset ''ncount'' ; say '>>> is ''rp''. To
control.'
    return
  end
end
if a < i+1 then do
  say '>>> Local NDM name ('loc1') is equal to the 'word(des.a,2)' one.'
  say '>>> To control in ''desnam''.'
  return
end
end
/***** Read job skeleton for prepare utility job
*****/
"ALLOC DA('nomej') F(KKK) SHR REUSE"
"EXECIO * DISKR KKK (STEM KK. FINIS"
"FREE F(KKK)"
do a = 1 to kk.0
  if pos('VOL=SER=',kk.a) = 0 & pos('UNIT=',kk.a) = 0 then do
    riga = kk.a ; a = 1000 ; leave
  end
end
if a = 1000 then do
  say '>>>' ; say '>>> Job skeleton is incorrect in ''nomej''. To
control.'

```

```

    say '>>>'
    return
end
parse value riga with n1 'UNIT=' un ',' n2 'VOL=SER=' vs ',' n3
/***** Dataset allocation to write utility jobs
*****/
workj = userid()||'.NVDM.SEND'
xx=OUTTRAP(dd.)
    ADDRESS TSO "DELETE "'workj'"
    "ALLOC DA("'workj'") DIR(500) SPACE(1,1) CYL DSORG(PO) ,
    "RECFM(F,B) LRECL(80) BLKSIZE(6160) VOLUME("vs") UNIT("un") NEW
CATALOG"
xx=OUTTRAP(OFF)
if rc > 0 then do
    do a = 1 to dd.0
        say dd.a
    end
    say '>>> Allocation of "'workj'" ; say '>>> in error (RC='rc'). To
control.'
    return
end
/***** Members name for execution of utility jobs
*****/
wexe = workj||'(WORKPWDC)' ; wlsnd = workj||'(WORKLIST)'
wjpac = workj||'(WRKMAINT)' ; wjpln = workj||'(WRKPLAN)'
wjout = workj||'.OUTPUT' ; wjrrr = workj||'(WRKJOBRR)'
wjrep = workj||'.RESREP' ; wjplo = workj||'(IEBCOPY)'
nomej = workj||'.OUTCOPY' ; pwok = 0
Ancor:
/***** Display panel to select function
*****/
DO forever
    if pwok = 0 then zzz = 'PW' ; else zzz = 'Q'
    ADDRESS ISPEXEC 'DISPLAY PANEL(NSPP000) CURSOR('zzz')'
    UPPER pw q r pp tt d
    if RC = 8 then Return ; if q = 'S' then leave
    if r = 'S' then do
        if pp = '' then pp = '*'
        if tt = '' then tt = '*'
        if d = '' then d = '*'
        if d = '1' then d = '8'
        if d = '2' then d = '7'
        if d = '3' then d = '2'
        ars = pp||'+'||tt||'+'||d
/***** Call clist NSPC015 for report Resource Repository
*****/
        Call NSPC015 ars,nomej,wjout,wjrrr,wjrep,desnam,ndes,pw
        if left(result,length(userid())) = userid() then pwok = 1
    end
END
if pwok = 0 then do

```

```

Call Checkpw
  if workjn4 > 99 then Return
  if workjn4 = -1 then signal Ancor
  pwok = 1
end
/** Call NSPC020 to define list objects to send at relative destination
**/
  Call NSPC020 wlsnd,desnam,tblnam,ndes
  if result = 100 then Signal Ancor ; if result = 101 then Signal Iesci
  wtms = 'T' || right(date('D'),3,'0') || left(time(),2) || substr(time(),4,2)
  say time() ; say time() ' Identification files is "'wtms'".'
  say time()
  /*****
  /* Verify if there are LOADs to send. */
  /* The loads module must be dealt differently */
  /*****
  "ALLOC DA('"wlsnd"') F(WKL) SHR REUSE"
  "EXECIO * DISKR WKL (STEM LL. FINIS"
  "FREE F(WKL)"
  do a = 1 to 11.0
    li = 11.a ; if pos('(',li) = 0 then iterate
    li = translate(li,' ','(.)')
    if word(li,ndes+3) = 'LOAD' then leave
  end
  if a < 1+11.0 then do
  say time() ; say time() ' There are LOAD. They will be processed
  opportunely.'
  say time()
  /***** Call clist NSPC025 to prepare Iebcopy utility for loads
  *****/
    Call NSPC025 wtms,wlsnd,nomej,nomey,wjplo,desnam,ndes,a
    If result = 100 then Signal IESCI ; else workjn3 = result
    end
  else workjn3 = ''
  jname = userid() || 'M' ; wjtyp = 'MAINTDAT'
  say time() ; say time() ' ... define job 'wjtyp ; say time()
  /*****
  /* Call clist NSPC030 to prepare maintdat utility (package) */
  /* Call NSPC035 to submit/control of the job for maintdat utility */
  /*****
  Call NSPC030 wtms,wlsnd,nomej,wjout,wjpac,jname,ndes
  If result = 100 then Signal IESCI
  Call NSPC035 wjpac,jname,wjtyp,wjout,pw
  workjn1 = result
  if f = 1 then do
    "ALLOC DA('"wlsnd"') F(WL0) SHR REUSE"
    "EXECIO * DISKR WL0 (STEM LL. FINIS"
    "FREE F(WL0)"
    ldstr = '.LOADS'
    do a = 1 to 11.0
      if pos(ldstr,word(11.a,ndes+1)) = 0 then do

```

```

        xx=OUTTRAP('dd.')
        ADDRESS TSO "DELETE '"word(11.a,ndes+1)'"
        xx=OUTTRAP('OFF')
    end
end
end
if workjnl = 100 then Signal Iesci
wjtyp = 'PLAN' ; jname = userid()||'P'
say time() ; say time() ' ... define job 'wjtyp
/*****
/* Call clist NSPC040 to prepare NvDM Plan (tcfmaint utility) */
/* Call clist NSPC035 to submit and control of the job for      */
/* tcfmaint utility. Call clist NSPC045 to write procedure log */
*****/
Call NSPC040 wtms,wlsnd,nomej,wjout,wjpln,desnam,jname,ndes,pcount
if result = 100 then Signal IESCI
Call NSPC035 wjpln,jname,wjtyp,wjout,pw
workjn2 = result
if workjn2 = 100 then Signal Iesci
Call NSPC045 wlsnd,wlogs,ndes
IESCI:
/***** Delete job output and work datasets *****/
xx=OUTTRAP('dd.')
ADDRESS TSO "CANCEL "workjn1" PURGE" ; ADDRESS TSO "CANCEL "workjn2"
PURGE"
ADDRESS TSO "CANCEL "workjn3" PURGE" ; ADDRESS TSO "CANCEL "workjn4"
PURGE"
xx=OUTTRAP('OFF')
if userid() = NDMADM then do
    delm1 = userid()||'.NVDM.SEND(WORKPWDC)'
    delm2 = userid()||'.NVDM.SEND(WRKMAINT)'
    delm3 = userid()||'.NVDM.SEND(WRKPLAN)' ; ADDRESS TSO "DELETE
'"delm1'"
    ADDRESS TSO "DELETE '"delm2'" ; ADDRESS TSO "DELETE '"delm3'"
    end
/* End procedure. */
return
Checkpw:
/** Read job skeleton and prepare job for verify user authorization NvDM
**/
wjtyp = 'CHECKPW' ; jname = userid()||'C'
say time() ; say time() ' ....define job 'wjtyp ; say time()
"ALLOC DA('"nomej"') F(JS2) SHR REUSE"
"EXECIO * DISKR JS2 (STEM WJ. FINIS"
"FREE F(JS2)"
wjn = left(jname,9) ; wj.1 = '//'||wjn||substr(wj.1,12)
do a = 2 to wj.0
    pn = pos('*FUNCT*',wj.a)
    if pn = 0 then leave
end

```

```

wj.a = left(wj.a,pn-1)'MAINTDAT' substr(wj.a,pn+7)
wj.a = left(wj.a,lastpos(', ',wj.a)-1)||''''
do a = 2 to wj.Ø
  pn = pos('*OUT*',wj.a)
  if pn = Ø then leave
end
wj.a = left(wj.a,pn-1)||wjout|| substr(wj.a,pn+5)
wj.a = left(wj.a,8Ø) ; wn = wj.Ø
wn = wn + 1 ; wj.wn = left(' END',8Ø) ; wn = wn + 1
wj.wn = left('/',|'| '*',8Ø) ; wn = wn + 1
wj.wn = left('//BATCHPW DD *',8Ø) ; wn = wn + 1
wj.wn = left('<PASSWORD>',8Ø) ; wn = wn + 1
wj.wn = left('/',|'| '*',8Ø) ; wj.Ø = wn
xx=OUTTRAP(dd.)
  ADDRESS TSO "DELETE "'wexe''''
xx=OUTTRAP(off)
"ALLOC DA(''wexe'') F(JCP) SHR REUSE"
"EXECIO * DISKW JCP (STEM WJ. FINIS"
"FREE F(JCP)"
pwj = '*'|pw
/**** Call NSPCØ35 to submit and control of the job for check password
****/
  Call NSPCØ35 wexe,jname,wjtyp,wjout,pwj
workjn4 = result
Return

```

## C-list NSPC015

```

/* REXX */
/* NvDM Send Procedure
   C-List NSPCØ15 - Called by c-list main NSPCØ1Ø
   Define job of utility to print directory of NvDM Resource Repository.
   The functions are: - prepare jcl for utility Maintdat;
                     - call clist NSPCØ35 to submit function;
*/
Trace ?o
ARG ars, nomej, wjout, wjrrr, wjrep, desnam, ndes, pw
wjtyp = 'PRINT DIR' ; jc = 'R' ; jname = userid()||jc
"ALLOC DA(''nomej'') F(JSK) SHR REUSE"
"EXECIO * DISKR JSK (STEM WJ. FINIS"
"FREE F(JSK)"
say time() ; say time() ' ... Define job 'wjtyp
jn = left(jname,9) ; wj.1 = '/'||jn|| substr(wj.1,12)
do a = 2 to wj.Ø
  pn = pos('*FUNCT*',wj.a)
  if pn = Ø then leave
end
wj.a = left(wj.a,pn-1)||'MAINTDAT'|| substr(wj.a,pn+7)
wj.a = left(wj.a,lastpos(', ',wj.a)-1)||''''
do a = 2 to wj.Ø

```

```

    pn = pos('*OUT*',wj.a)
    if pn = 0 then leave
end
wj.a = left(wj.a,pn-1)||wjout||substr(wj.a,pn+5)
wj.a = left(wj.a,80) ; dpd = 'DSXPRINT DD'
do a = 2 to wj.0
    if pos(dpd,wj.a) = 0 then leave
end
wj.a = '//dpd' DSN='wjrep',' ; wj.a = left(wj.a,80)
a1 = a + 1 ; sysin = wj.a1 ; a = a + 1
b = a-3 ; wj.a = wj.b ; a = a+1 ; b = a-3 ; wj.a = wj.b
a = a+1 ; wj.a = sysin ; a = a+1 ; wj.a = ' PRINT DIRECTORY
TYPE=DATASET'
wj.a = left(wj.a,80) ; a = a+1 ; wj.a = ' END'
wj.a = left(wj.a,80) ; a = a+1 ; wj.a = left('/','|'*,80)
a = a+1 ; wj.a = left('//BATCHPW DD *',80) ; a = a+1
wj.a = left('<PASSWORD>',80) ; a = a+1 ; wj.a = left('/','|'*,80)
wj.0 = a
xx=OUTTRAP(DD.)
    ADDRESS TSO "DELETE ""wjrrr"""
xx=OUTTRAP(OFF)
"ALLOC DA("""wjrrr""") F(JRR) SHR REUSE"
"EXECIO * DISKW JRR (STEM WJ. FINIS"
"FREE F(JRR)"
xx=OUTTRAP('jobm.')
    ADDRESS TSO "DELETE ""wjrep"""
xx=OUTTRAP('OFF')
Call NSPC035 wjrrr,jname,wjtyp,wjout,pw
if result = 100 then Signal Iesci ; else workjn = result
"ALLOC DA("""wjrep""") F(RRD) SHR REUSE"
"EXECIO * DISKR RRD (STEM RR. FINIS"
"FREE F(RRD)"
str.1 = 'Resource type' ; str.2 = 'Resource name' ; str.3 = 'Complete. .
.'
str.4 = 'Creation date' ; str.5 = 'Creation time' ; qstr = 5 ; e = 1 ;
it = 0
do a = 1 to rr.0
    if pos(str.e,rr.a) = 0 then do
        if e = 1 then do
            it = it+1 ; lrr.it = ''
        end
        lrr.it = lrr.it word(rr.a,words(rr.a)) ; e = e+1
        if e > qstr then e = 1
    end
end
if it = 0 then do
    say '>>>'
    say '>>>' The Resource Repository does not contain some relative' ,
    say '>>>' object to the selected project.' ; say '>>>'
    Signal Iesci
end

```

```

"ALLOC DA('desnam') F(DES) SHR REUSE"
"EXECIO * DISKR DES (STEM DS. FINIS"
"FREE F(DES)"
do a = 1 to ds.Ø
  UPPER ds.a
  end
a = Ø
do b = 1 to ds.Ø
  c1 = left(strip(ds.b),1)
  if c1 = '*' | c1 = ' ' then iterate
  a = a+1 ; des.a = word(ds.b,2)
  end
TNa = Ø ; tabf = Ø
do a = 1 to it
  if words(lrr.a) = qstr | word(lrr.a,1) = 'DATASET' | ,
    word(lrr.a,3) = 'YES' then iterate
  nome = word(lrr.a,2)
  Call NSPCØ16 nome,ars
  if result = 'NO' then iterate
  nome = result
  if tabf = Ø then do
    tabf = 1 ; tabella='@@TRR@@'
    ADDRESS ISPEXEC 'TBOPEN 'tabella' NOWRITE'
    if rc = 8 then ,
      ADDRESS ISPEXEC 'TBCREATE 'tabella ,
        'NAMES(TN TNAM TDAT TTIM TDES) NOWRITE'
    end
    TNAM = nome ; TNa = TNa+1 ; TN = TNAM
    TDAT = word(lrr.a,4) ; TTIM = word(lrr.a,5)
    ndes = substr(word(lrr.a,2),lastpos('.',word(lrr.a,2))+1)
    TDES = des.ndes
    ADDRESS ISPEXEC 'TBADD 'tabella
  end
  if tabf = Ø then do
    say '>>>'
    say '>>> No object found in the Resouce Repository that corresponds'
    say '>>> to the carried out selection.' ; say '>>>'
    Signal Iesci
  end
  ADDRESS ISPEXEC 'TBSORT 'tabella' FIELDS(TDAT,C,A,TTIM,C,A,TNAM,C,A)'
  ADDRESS ISPEXEC 'TBTOP 'tabella
  do a = Ø to TNa
    ADDRESS ISPEXEC 'TBGET 'tabella
    TN = a
    ADDRESS ISPEXEC 'TBPUT 'tabella
    ADDRESS ISPEXEC 'TBSKIP 'tabella
  end
  ADDRESS ISPEXEC 'TBTOP 'tabella
  Do forever
  ADDRESS ISPEXEC 'TBDISPL 'tabella' PANEL(NSPPØØ4)'

```

```

    if rc = 8 then leave
    end
ADDRESS ISPEXEC 'TBEND 'tabella
Iesci:
xx=OUTTRAP('pu.')
if workjn = 100 then ADDRESS TSO "CANCEL "workjn" PURGE"
xx=OUTTRAP('off')
Return result

```

## C-list NSPC016

```

/* REXX */
/* NvDM Send Procedure - C-List NSPC016 Called by c-list NSPC015.
   Reorg datasets name to display user. */
Trace ?o
ARG nome,criteri
if nome = '' then Return 'NO'
nomb = translate(nome,' ','.')
if words(nomb) = 5 then Return 'NO'
q1 = word(nomb,1) ; q2 = word(nomb,2) ; q3 = word(nomb,3)
q4 = word(nomb,4) ; q5 = word(nomb,5)
if datatype(q5) = 'NUM' | length(q5) = 1 then Return 'NO'
ndes = 8
if q5 > ndes then Return 'NO'
if criteri = '' then do
  parse var criteri pp '+' tt '+' dd
  if pp = '*' & pp = q1 then Return 'NO'
  if tt = '*' then do
    if tt = 'LOAD' then do
      if q3 = 'LOADS' then Return 'NO'
    end
    else if tt = q3 then Return 'NO'
  end
  if dd = '*' & dd = q5 then Return 'NO'
end
nome = q1||'.'||q2||'.'||q3||'.'||q4
Return nome

```

## C-list NSPC020

```

/* REXX */
/* NvDM Send Procedure - C-List NSPC020 Called by c-list main NSPC010
   It selects the datasets to prepare for the shipment.
   The functions are:
   - display panel for select datasets; - prepare maintdat function */
Trace ?o
signal on syntax
ARG wlsnd,desnam,tblnam,ndes
recl. = '' ; tbs = 0 ; tbcnt = 0
"ALLOC DA('desnam') F(DS1) SHR REUSE"

```



```

"EXECIO * DISKR DS1 (STEM DS. FINIS"
"FREE F(DS1)"
i = 0
do a = 1 to ds.0
  c1 = left(strip(ds.a),1)
  if c1 = '*' | c1 = ' ' then iterate
  i = i+1 ; dest.i = word(ds.a,2) ; UPPER dest.i
end
"ALLOC DA('tblnam') F(DS2) SHR REUSE"
"EXECIO * DISKR DS2 (STEM PD. FINIS"
"FREE F(DS2)"
pd0 = 0
do a = 1 to pd.0
  UPPER pd.a
  c1 = left(strip(pd.a),1)
  if c1 = '*' | c1 = ' ' then iterate
  if words(pd.a) = ndes+1 then do
    say '>>>'
    say '>>> The record 'a' into 'tblnam' is incorrect. To control.'
    say '>>>' ; Return 101
  end
  pd0 = pd0+1 ; pd.pd0 = strip(pd.a)
end
due = ' '
Do forever
  ADDRESS ISPEXEC 'DISPLAY PANEL(NSPP001)'
  UPPER o uno due tre mem das p q r s t u v w x
  if RC = 8 | o = 'Y' then leave
  if p = ' ' | p = '' then p = '_'
  if q = ' ' | q = '' then q = '_'
  if r = ' ' | r = '' then r = '_'
  if s = ' ' | s = '' then s = '_'
  if t = ' ' | t = '' then t = '_'
  if u = ' ' | u = '' then u = '_'
  if v = ' ' | v = '' then v = '_'
  if w = ' ' | w = '' then w = '_'
  if x = 'S' then do
    Call AllDest
    if result = 100 then iterate
  end
  else dests = _ q _ s _ _ v w
  if das = '' then do
    if pos('(',das) = 0 then do
      say '>>>' ; say '>>> Selection area for single members is
wrong.'
      say '>>>' ; iterate
    end
    nome = das
  end
  else do
    if tre = 'LOAD' & due = 'SHIPMS' then do

```

```

    say ; say '"LOAD" type is wrong.' ; say ; iterate
  end
  nome = uno'.'due'.'tre
end
if x ≠ 'S' then do
  prog = left(nome,pos('.',nome)-1)
  Call CheckPD
  if result = 100 then iterate
end
ww = "xx=SYSDSN('"'nome"'"")
INTERPRET ww
if xx ≠ 'OK' then do
  say '>>>' ; say '>>> Status of "'nome'" is "'xx"'. To control.'
  iterate
end
chkload = word(translate(nome,' ','.()'),3)
if chkload = 'LOAD' then do
  ww="xx=LISTDSI('"'nome"'"")
  INTERPRET ww
  if left(SYSDSORG,2) ≠ 'P0' | pos('U',SYSRECFM) = 0 then do
    say '>>>'
    say '>>> The datasets LOAD must be "P0" and RecFm="U". To
control.'
    say '>>>' ; iterate
  end
end
if das ≠ '' then do
  Call Membri
  if rms.0 = 'N' then Call InTab
  else do
    do a = 1 to rms.0
      nome = das('rms.a') ; Call InTab
    end
    if rms.0 = 0 then do
      say '>>>'
      say '>>> "'nome'" is Partitioned but members not found.' ,
' To control.' ; say '>>>'
    end
  else do
    say '>>>' ; say '>>>All members of "'das'" will be sended.' ; say
'>>>'
  end
end
iterate
end
mem = strip(mem) ; if mem = '*' then mem = ''
if mem ≠ '' then do
  ast = pos('*',mem)
  if ast = 0 then do
    nome = nome('mem') ; ww = "xx=SYSDSN('"'nome"'"")
    INTERPRET ww
  end
end

```

```

if xx = 'OK' then do
say '>>>' ; say '>>> Status of "'nome'" is "'xx'". To control.'
say '>>>' ; iterate ; end
    else do
        Call RdcB
        Call InTab
        iterate
    end
end
else do
Call Membri
if rms.Ø = 'N' then do
    say '>>>' ; say '>>> "'nome'" is Sequential. To control.'
    say '>>>' ; iterate
end
if rms.Ø = Ø then do
    say '>>>' ; say '>>> Members not found in "'nome'". To control.'
    Iterate ; end
else do
    lu1 = ast-1 ; inimem = left(mem,lu1) ; b = Ø
    do a = 1 to rms.Ø
        initab = left(rms.a,lu1)
        if inimem = initab then do
            b = b + 1 ; sms.b = rms.a
        end
    end
    if b = Ø then do
say '>>>' ; say '>>> "'nome'" members not found with "'inimem'*". To
control.'
say '>>>' ; iterate ; end
        else do
            sms.Ø = b
            Call SelMems
            iterate
        end
    end
end
end
else do
    Call Membri
if rms.Ø = 'N' then do
say '>>>' ; say '>>> Specify the sequential dataset into correct field.'
say '>>>' ; iterate
end
if rms.Ø = Ø then do
    say '>>>' ; say '>>> Members not found into "'nome'". To control.'
    say '>>>' ; iterate
end
else do
    do a = Ø to rms.Ø
        sms.a = rms.a
    end
end

```

```

Call SelMems
iterate
end ; end ; End
if tbs = 0 | o = 'Y' then Return 100
tabout.1 = recl.1 ; wn = 1
if tbs > 1 then do a = 2 to tbs
  ugu = 0
  do b = 1 to wn
    if recl.a = tabout.b then do
      ugu = 1 ; b = wn
    end
  end
end
if ugu = 0 then do
  wn = wn + 1 ; tabout.wn = recl.a
end
end
tabout.0 = wn
Call LastList
If abort = 'Y' then Return 100
xx=OUTTRAP(dd.)
  ADDRESS TSO "DELETE "'wlsnd'"
xx=OUTTRAP('OFF')
"ALLOC DA("'wlsnd'') F(WKL) SHR REUSE"
"EXECIO "tabout.0" DISKW WKL (STEM TABOUT. FINIS"
"FREE F(WKL)"
Return
InTab:
tbs=tbs + 1 ; recl.tbs = dests nome lr bl rf do
Return
Membri:
xx=OUTTRAP(memtrap.)
ADDRESS TSO "LISTDS "'nome"' MEMBER"
xx=OUTTRAP('OFF')
b = 0
do a = 1 to memtrap.0
  lr = substr(memtrap.3,9,5) ; bl = substr(memtrap.3,15,5)
  rf = substr(memtrap.3,3,3) ; do = substr(memtrap.3,23,3)
  if pos('--MEMBERS--',memtrap.a) = 0 then do
    b = 1 ; leave
  end
end
end
if b = 0 then rms.0 = 'N'
else do
  b = 0
  do c = a+1 to memtrap.0
    b = b+1
    rms.b = strip(memtrap.c)
  end
end
rms.0 = b
end
Return

```

```

SelMems:
  tabella = '@@TB1@@' ; nomm = nome
  ADDRESS ISPEXEC 'TBOPEN 'tabella' NOWRITE'
  if rc = 8 then,
    ADDRESS ISPEXEC 'TBCREATE 'tabella ,
      'NAMES(MNAME SELE) NOWRITE'
  do a = 1 to sms.Ø
    MNAME = sms.a; SELE='          ' ; SELE = '          '
    ADDRESS ISPEXEC 'TBADD 'tabella
  end
  ADDRESS ISPEXEC 'TBTOP 'tabella
  csrr = 1
DO Forever
  ADDRESS ISPEXEC 'TBDISPL 'tabella,
  ' PANEL(NSPPØØ2) CSRROW('csrr') AUTOSEL(NO)'
  IF RC = 8 then leave
  if zcmd = 'ALL' then do
    do a = 1 to sms.Ø
      nome = nomm('sms.a')'
      call InTab
    end
  leave
  end
  else do
    smn = Ø ; ss = ztdsels+Ø; if ss=Ø then,
    do a=1 to ss
      if a = 1 then ADDRESS ISPEXEC 'TBDISPL 'tabella'
POSITION(CRP)'
      if ms = 'S' then do
        nome = nomm('mname')'
        if SELE = 'SELECTED' then iterate
        SELE = 'SELECTED'
        call InTab
        ADDRESS ISPEXEC 'TBPUT 'tabella
      end
    end
  end
  if words(zcmd) = 2 & word(zcmd,1) = 'L' then do
    do y = 1 to sms.Ø
      if sms.y > word(zcmd,2) then leave
    end
    ADDRESS ISPEXEC 'TBSKIP 'tabella' NUMBER('y-1')'
  end
END
ADDRESS ISPEXEC 'TBEND 'tabella
Return
LastList:
  tabella = '@@TB2@@'
  ADDRESS ISPEXEC 'TBOPEN 'tabella' NOWRITE'
  if rc = 8 then,
    ADDRESS ISPEXEC 'TBCREATE 'tabella,
      'NAMES(DOBJ DTR DMI DRM DPA DFI DGE DNA1 DNA2) NOWRITE'

```

```

do a = 1 to tabout.0
  DOBJ = word(tabout.a,ndes+1)
  if word(tabout.a,1) = 'S' then DTR='TR' ; else DTR='--'
  if word(tabout.a,2) = 'S' then DMI='MILSA' ; else DMI='-----'
  if word(tabout.a,3) = 'S' then DRM='RM' ; else DRM='--'
  if word(tabout.a,4) = 'S' then DPA='PA' ; else DPA='--'
  if word(tabout.a,5) = 'S' then DFI='FI' ; else DFI='--'
  if word(tabout.a,6) = 'S' then DGE='GE' ; else DGE='--'
  if word(tabout.a,7) = 'S' then DNA1='NA1' ; else DNA1='----'
  if word(tabout.a,8) = 'S' then DNA2='NAPO2' ; else DNA2='-----'
  select
    when na2='__' then DNA2='----' ; when na2='S_' then DNA2='N06'
    when na2='_S' then DNA2='N07' ; Otherwise DNA2='NA2'
  end
  ADDRESS ISPEXEC 'TBADD 'tabella
end
ADDRESS ISPEXEC 'TBTOP 'tabella
abort = 'N' ; csrr = 1 ; csr = '' ; nno = 0
DO Forever
  sprstt = (0+tabout.0)-nno
  if sprstt < 16 then ADDRESS ISPEXEC 'TBTOP 'tabella
  ADDRESS ISPEXEC 'TBDISPL 'tabella,
  ' PANEL(NSPP003) CURSOR('csr')',
  ' CSRROW('csrr') POSITION(CRP) AUTOSEL(NO)'
  IF RC = 8 then leave
  if zcmd ^= '' then UPPER zcmd
  if zcmd = 'ALL' then do
  abort = 'Y' ; leave
  end
  ss = ztdsels+0
  if ss ^= 0 then,
  do a = 1 to ss
    if a ^= 1 then ADDRESS ISPEXEC 'TBDISPL 'tabella' POSITION(CRP)'
    UPPER ds
    if ds = 'D' then do
      nno = nno+1
      if nno = tabout.0 then do
        abort = 'Y' ; leave
      end
      if DTR='--' then d1='_' ; else d1='S'
      if DMI='-----' then d2='_' ; else d2='S'
      if DRM='--' then d3='_' ; else d3='S'
      if DPA='--' then d4='_' ; else d4='S'
      if DFI='--' then d5='_' ; else d5='S'
      if DGE='--' then d6='_' ; else d6='S'
      if DNA1='----' then d7='_' ; else d7='S'
      if DNA2='-----' then d8='_' ; else d8='S'
      nonno = d1 d2 d3 d4 d5 d6 d7 d8 dobj ; tabno.nno = strip(nonno)
      ADDRESS ISPEXEC 'TBDELETE 'tabella
      csr = 'DOBJ'
    end
  end
end

```

```

        end
    if abort = 'Y' then leave
END
if abort = 'N' then Do
wt = 0
do a = 1 to tabout.0
    elim = 0
    do b = 1 to nno
        if strip(tabout.a) = tabno.b then do
            elim = 1 ; leave
        end
    end
    if elim = 0 then do
        wt = wt + 1 ; tabout.wt = tabout.a
    end
end
tabout.0 = wt
End
else do
    say '>>>' ; say '>>> The distributions have been cancelled.' ; say '>>>'
end
ADDRESS ISPEXEC 'TBEND 'tabella
Return
CheckPD:
result = 0
do a = 1 to pd0
    if word(pd.a,1) = prog then do
        do b = 1 to ndes
            if word(dests,b) = 'S' & word(pd.a,b+1) ≠ 'S' then do
                say '>>>' ; say '>>> The project "'prog'" is not managed by "'dest.b"'.'
                say '>>> Input is ignored.' ; say '>>>'
                result = 100 ; a = 9999 ; leave
            end
        end
    end
    leave
end
end
if a = pd0 + 1 then do
    say '>>>' ; say '>>> This project "'prog'" is not managed.'
    say '>>> Input is ignored.' ; say '>>>' ; result = 100
end
Return result
AllDest:
result = 0 ; dests = ''
do a = 1 to pd0
    if word(pd.a,1) = uno then do
        do b = 1 to ndes
            if word(pd.a,b+1) = 'S' then dests=dests||'S ' ; else dests = dests||'_ '
        end
    end
end

```

```

    end
    dests = strip(dests) ; leave
  end
end
dests = '_ S _ S _ _ S S'
if a = pd0 + 1 then do
  say '>>>' ; say '>>> The project "'uno'" is not managed.'
  say '>>> Input is ignored.' ; say '>>>' ; result = 100
end
Return result
syntax:
say copies('|',70) ; say 'Code 'sigl' in NSPC020 : "'errortext(rc)'"
RC='rc
say right(sigl,6,'0')' 'strip(sourceline(sigl),'t') ; say copies('|',70)
Return 101
Rdcb:
xx=OUTTRAP(dcbtrap.)
ADDRESS TSO "LISTDS "'nome'"
xx=OUTTRAP('OFF')
  do a = 1 to dcbtrap.0
    lr = substr(dcbtrap.3,9,5) ; bl = substr(dcbtrap.3,15,5)
    rf = substr(dcbtrap.3,3,3) ; do = substr(dcbtrap.3,23,3)
  return

```

## C-list NSPC025

```

/* REXX */
/* NVDM Send Procedure - C-List NSPC025 Called by c-list main NSPC010
   It manages IEBCOPY utility in order to manipulate the loads to send.
   The functions are: - write jcl for Iebcopy utility;
   - call clist NSPC035 to submit and verify utility */
Trace ?o
ARG wtms,wlsnd,nonej,nomey,wjplo,desnam,ndes,f
result = 0 ; jobtype = 'IEBCOPYs' ; jc = 'I' ; jname = userid()||'I'
"ALLOC DA('wlsnd') F(WL1) SHR REUSE"
"EXECIO * DISKR WL1 (STEM LL. FINIS"
"FREE F(WL1)"
l = 0 ; n = f-1
do a=f to ll.0
  if pos('(',ll.a) = 0 then do
    n = n + 1 ; ll.n = ll.a ; iterate
  end
  li = word(ll.a,ndes+1) ; li = translate(li,' ','(.)')
  if word(li,3) = 'LOAD' then do
    l = l+1 ; rl.l = ll.a ; bl.l = li
  end
  else do
    n = n+1 ; ll.n = ll.a
  end
end
ll.0 = n ; rl.0 = l ; bl.0 = l

```



```

if bl.Ø > 1 then do y = 1 to (-1)+bl.Ø
  w = y + 1
  if WORD(bl.w,1) < WORD(bl.y,1) then do
    c1 = rl.y ; c2 = bl.y ; rl.y = rl.w ; bl.y = bl.w
    rl.w = c1 ; bl.w = c2 ; y = Ø
  end
end
qp = Ø ; pp = ''
do a = 1 to bl.Ø
  if word(bl.a,1) ≠ pp then do
    qp = qp+1 ; pp = word(bl.a,1)
  end
end
"ALLOC DA(''desnam'') F(DES) SHR REUSE"
"EXECIO * DISKR DES (STEM DE. FINIS"
"FREE F(DES)"
do a=1 to de.Ø
  UPPER de.a
end
i = Ø ; locl = Ø
do a=1 to de.Ø
  c1 = left(strip(de.a),1)
  if c1 = '*' | c1 = ' ' then iterate
  i = i + 1 ; ds.i = word(de.a,3)
  if pos('*HOME*',de.a) ≠ Ø then locl = i
end
"ALLOC DA(''nomej'') F(JSK) SHR REUSE"
"EXECIO * DISKR JSK (STEM JS. FINIS"
"FREE F(JSK)"
say time() ; say time() ' Define job 'jobtype' ...'
ji.1 = left('/',||jname,11)||substr(js.1,12)
do a=2 to 6
  ji.a = js.a
end
ii = 6
do a=7 to js.Ø
  if pos('*OUT*',js.a) ≠ Ø then leave
end
ww="xx=LISTDSI(''wlsnd'')"
INTERPRET ww
wvol=SYSVOLUME ; wunt=SYSUNIT
st = Ø ; ioli = Ø ; pm = Ø
do a=1 to qp
  cp = pm + 1 ; proj = word(bl.cp,1) ; qm = Ø
  do b=1 to bl.Ø
    if word(bl.b,1) = proj then qm = qm+1
  end
  do b=1 to ndes
    qmand = Ø ; inst = Ø
    do c=1 to qm
      d = pm+c
    end
  end
end

```

```

        if word(rl.d,b) = 'S' then do
            if inst = 0 then DO
                inst = 1
                if ioli = 0 then do
                    ji.ii = ji.ii)' ; ji.ii = left(ji.ii,80)
                    ii = ii + 1 ; ji.ii = '//*'
                    ji.ii = left(ji.ii,80) ; iol.lolin = iol.lolin mems
                end
            st = st + 1 ; iebout = proj'.'ds.b'.LOADS.'wtms
            xx=OUTTRAP(utt.)
            ADDRESS TSO "DELETE "'iebout'"
            xx=OUTTRAP(OFF)
            ii = ii + 1 ; ji.ii = '//right(ds.b,4)'P'right(st,3,'0')' EXEC
            PGM=IEBCOPY'
            ji.ii = left(ji.ii,80) ; tdsn = word(rl.d,ndes+1)
            ldsn = left(tdsn,pos('(',tdsn)-1) ; ii = ii + 1
            ji.ii = '//IDD      DD DSN='ldsn',DISP=SHR' ; ji.ii = left(ji.ii,80)
            ii = ii + 1 ; ji.ii = '//ODD      DD DSN='iebout','
            ji.ii = left(ji.ii,80) ; ii = ii + 1
            ji.ii = '// UNIT='wunt',VOL=SER='wvol',DISP=(NEW,CATLG),'
            ji.ii = left(ji.ii,80)
            ww="xx=LISTDSI('"'ldsn"'"')
            INTERPRET ww
            bsize = SYSBLKSIZE ; ioli = ioli + 1
            iol.ioli = ds.b iebout bsize ; lolin = ioli
            ii = ii + 1 ; ji.ii = '// DCB=(RECFM=V,BLKSIZE='bsize',LRECL=80),'
            ji.ii = left(ji.ii,80) ; ii = ii + 1 ; ji.ii = '// SPACE=(CYL,(5,5,0))'
            ji.ii = left(ji.ii,80) ; ii = ii + 1
            ji.ii = '//SYSUT3  DD UNIT=3390,SPACE=(CYL,(10,1))'
            ji.ii = left(ji.ii,80) ; ii = ii + 1
            ji.ii = '//SYSUT4  DD UNIT=3390,SPACE=(CYL,(10,1))'
            ji.ii = left(ji.ii,80) ; ii = ii + 1
            ji.ii = '//SYSPRINT DD SYSOUT=*' ; ji.ii = left(ji.ii,80)
            ii = ii + 1 ; ji.ii = '//SYSIN  DD *'
            ji.ii = left(ji.ii,80) ; ii = ii + 1
            ji.ii = '      COPY INDD=IDD,OUTDD=ODD'
            ji.ii = left(ji.ii,80) ; ii = ii + 1
            ji.ii = '      SELECT MEMBER=('word(bl.d,4)
            ioli = ioli + 1 ; iol.ioli = '*M* 'word(bl.d,4) ; mems = 1
        END
    else do
        ugu = 0
        do q=1 to qmand
            if word(bl.d,4) = mand.q then do
                ugu = 1 ; leave ; end
            end
        if ugu = 1 then iterate
            ji.ii = ji.ii',' ; ji.ii = left(ji.ii,71)'X'
            ji.ii = left(ji.ii,80) ; ii = ii + 1
            ji.ii = '      'word(bl.d,4) ; ioli = ioli + 1
            iol.ioli = '*M* 'word(bl.d,4) ; mems = mems + 1

```

```

                end
                qmand = qmand + 1 ; mand.qmand = word(bl.d,4)
            end
        end
    end
    pm = pm + qm
end
iol.lolin = iol.lolin mems ; ji.ii = ji.ii)'
ji.ii = left(ji.ii,80) ; ii = ii + 1 ; ji.ii = '/*'
ji.ii = left(ji.ii,80) ; ji.0 = ii
xx=OUTTRAP(utt.)
    ADDRESS TSO "DELETE '"wjplo'"
xx=OUTTRAP(OFF)
"ALLOC DA('"wjplo"') F(IEB) SHR REUSE"
"EXECIO * DISKW IEB (STEM JI. FINIS"
"FREE F(IEB)"
xx=OUTTRAP('jobm.')
    ADDRESS TSO "DELETE '"nomey'"
xx=OUTTRAP('OFF')
Call NSPC035 wjplo,jname,jobtype,'***','***'
say time() ; say time() ' ... Verify 'jobtype ; say time()
xx=OUTTRAP(utt.)
    ADDRESS TSO "DELETE '"wlsnd'"
xx=OUTTRAP(OFF)
ll0 = ll.0
do a=1 to ioli
    if left(strip(iol.a),1) = '*' then do
        ll0 = ll0 + 1 ; ll.ll0 = iol.a ; iterate
    end
    dess = ''
    do b=1 to ndes
        if ds.b = word(iol.a,1) then dess = dess||'S ' ; else dess =
dess||'_ '
    end
    ll0 = ll0 + 1 ; ll.ll0 = dess||word(iol.a,2) word(iol.a,3)
word(iol.a,4)
end
ll.0 = ll0
"ALLOC DA('"wlsnd"') F(WL2) SHR REUSE"
"EXECIO * DISKW WL2 (STEM LL. FINIS"
"FREE F(WL2)"
Iesci:
if result = 100 then result = workjn
Return result

```

## C-list NSPC030

```

/* REXX */
/* NvDM Send Procedure
   C-List NSPC030 - Called by c-list main NSPC010
   Define jcl to execute Maintdat NvDM utility

```

The functions are: - read skeleton job and dataset list to send;  
 - write jcl for maintdat; \*/

Trace ?o

ARG wtms,wlsnd,nonej,wjout,wjpac,jname,ndes

jc = 'M'

/\*\*\*\*\* Read skeleton and work dataset list \*\*\*\*\*/

"ALLOC DA('wlsnd') F(WLL) SHR REUSE"

"EXECIO \* DISKR WLL (STEM LL. FINIS"

"FREE F(WLL)"

"ALLOC DA('nonej') F(JS1) SHR REUSE"

"EXECIO \* DISKR JS1 (STEM JD. FINIS"

"FREE F(JS1)"

ndds = 0 ; nts = 0

do a=1 to ll.0

if left(strip(ll.a),1) = '\*' then iterate

nomo = word(ll.a,ndes+1) ; lrlo = word(ll.a,ndes+2) ; blk0 =  
 word(ll.a,ndes+3)

refo = word(ll.a,ndes+4) ; dsoo = word(ll.a,ndes+5)

if pos('V',refo) = 0 then tf = 'N' ; else tf = 'Y'

do b=1 to ndds

if nomo = word(ndd.b,1) then leave

end

ddn = b

if b > ndds | ndds = 0 then do

ndds = ndds+1 ; ndd.ndds = nomo lrlo blk0 tf ; ddn = ndds

end

do b=1 to ndes

if word(ll.a,b) = 'S' then iterate

nomets = nomo||'|.'||b||'|' ddn||'|' ||lrlo||'|' 'blk0||'|' '||tf

if nts > 0 then do

do c=1 to nts

if nomets = nts.c then leave

end

if c < nts + 1 then iterate

end

nts = nts + 1 ; nts.nts = nomets

end

end

jn = left(jname,9) ; jd.1 = '/'||jn||substr(jd.1,12)

do a=2 to jd.0

pn = pos('\*FUNCT\*',jd.a)

if pn = 0 then leave

end

jd.a = left(jd.a,pn-1)'MAINTDAT'substr(jd.a,pn+7)

jd.a = left(jd.a,lastpos(', ',jd.a)-1)||''

do a=2 to jd.0

pn = pos('\*OUT\*',jd.a)

if pn = 0 then leave

end

jd.a = left(jd.a,pn-1)||wjout||substr(jd.a,pn+5) ; jd.a = left(jd.a,80)

do a=2 to jd.0

```

    if left(jd.a,7) = '//SYSIN' then leave
end
sysin = jd.a ; a = a-1
do b=1 to ndds
  a = a + 1 ; jd.a = '//USER' || left(b,4) || ' DD DSN=' || word(ndd.b,1) || ','
  if length(jd.a) + 8 < 72 then jd.a = jd.a || 'DISP=SHR'
    else do
      a = a + 1 ; jd.a = '//          DISP=SHR'
    end
end
a = a + 1 ; jd.a = sysin ; b15 = copies(' ',15) ; cts = 4
ct.1 = ' PREPARE DATASET INDD=*DD2*,'
ct.2 = b15'ID=*NN2*,' ; ct.3 = b15'FCTLNAME=NDM,DISP=(OLD,DELETE),'
ct.4 = b15'LBLRIND=LR,LRLLEN=80,LBLEN=256,VARLEN=N,SRCTYPE=PS'
ctsl = 4 ; ct1.1 = ' PREPARE DATASET INDD=*DD2*,'
ct1.2=b15'ID=*NN2*,' ;
ct1.3=b15'FCTLNAME=NDM,DISP=(OLD,DELETE),SRCTYPE=PS,'
ct1.4 = b15'VARLEN=Y,LBLRIND=LB,LBLEN='
do c=1 to nts
  wnc1 = word(nts.c,1) ; wnc2 = word(nts.c,2) ; wnc3 = word(nts.c,3)
  wnc4 = word(nts.c,4) ; wnc5 = word(nts.c,5)
  if pos('.LOADS',wnc1) = 0 then do
    do b=1 to cts
      pd = pos('*DD2*',ct.b)
      if pd = 0 then do
        pp = pd ; st = 'USER' || wnc2
        call Sos2
      end
      else do
        pn = pos('*NN2*',ct.b)
        if pn = 0 then do
          pp = pn ; st = wnc1
          call Sos2
        end
        else do
          a = a + 1 ; jd.a = left(left(ct.b,71)'X',80)
          if b = cts then call Sos4
        end
      end
    end
  end
end
end
else do
  do b=1 to cts1
    pd = pos('*DD2*',ct1.b)
    if pd = 0 then do
      pp = pd ; st = 'USER' || wnc2
      call Sos3
    end
    else do
      pn = pos('*NN2*',ct1.b)
      if pn = 0 then do

```

```

        pp = pn ; st = wnc1
        call Sos3
        end
        else do
            a = a + 1 ; jd.a = left(left(ct1.b,71)'X',80)
            if b = cts then jd.a = left(ct1.b||word(ndd.wnc2,2),80)
            end
        end
    end
end
end
end
a = a + 1 ; jd.a = left(' END',80) ; a = a + 1 ; jd.a = left('/
'|'*',80)
a = a + 1 ; jd.a = left('//BATCPW DD *',80) ; a = a + 1
jd.a = left('<PASSWORD>',80) ; a = a + 1 ; jd.a = left('/'|'*',80)
jd.0 = a
xx=OUTTRAP(dd.)
    ADDRESS TSO "DELETE ""wjpac""
xx=OUTTRAP(off)
"ALLOC DA(""wjpac"") F(JMD) SHR REUSE"
"EXECIO * DISKW JMD (STEM JD. FINIS"
"FREE F(JMD)"
Return
Sos2:
a = a + 1 ; st = translate(st,'. ','()') ; st = strip(st) ; pb = pos('
',st)
if pb = 0 then st = left(st,pb-1)||substr(st,pb+1)
pdst = pos('.SHIPMS.',st)
if pdst > 0 then st = left(st,pdst)||wtms||substr(st,pdst+7)
jd.a = left(ct.b,pp-1)||st ; con = substr(ct.b,pp+5,1)
if con = ' ' then jd.a = left(jd.a,80)
    else if con = ',' then jd.a = left(left(jd.a||','',71)||'X',80)
    else do
        say ; say '*** ERROR: after ""XXX"" must be comma or blank' ; say
    end
Return
Sos3:
a = a + 1 ; jd.a = left(ct1.b,pp-1)||st ; con = substr(ct1.b,pp+5,1)
if con = ' ' then jd.a = left(jd.a,80)
    else if con = ',' then jd.a = left(left(jd.a||','',71)||'X',80)
    else do
        say ; say '*** ERROR: after ""XXX"" must be comma or blank' ; say
    end
Return
Sos4:
jd1 = left(ct.b,32) ; jd2 = substr(ct.b,35,7)
jd3 = substr(ct.b,45,8) ; jd4 = substr(ct.b,54,11)
jd.a = jd1||wnc3||jd2||wnc4||jd3||wnc5||jd4
Return

```

## C-list NSPC035

```
/* REXX */
/* NvDM Send Procedure
   C-List NSPC035 - Called by c-list main NSPC010 Execute NvDM
utilities
   The functions are: - submit jobs of utilities;
                     - verification the corrected execution of the utilities;
*/
Trace ?o
ARG tosub,jn,wjtyp,jout,pw
cpwf = 0
if pw ^= '***' & left(pw,1) = '*' then do
  cpwf = 1 ; pw = substr(pw,2)
end
jobendmsg = 'OUTPUT QUEUE'
if pw ^= '***' then do
  "ALLOC DA('"tosub"') F(SUB) SHR REUSE"
  "EXECIO * DISKR SUB (STEM LL. FINIS"
  do a=1 to ll.0
    upper ll.a
    if strip(ll.a) = '<PASSWORD>' then ll.a = pw
  end
  "EXECIO * DISKW SUB (STEM LL. FINIS"
  "FREE F(SUB)"
end
say time() ; say time() ' ... Submit job 'wjtyp
xx=OUTTRAP('jobm.')
  if jout ^= '***' then ADDRESS TSO "DELETE '"jout'"
  ADDRESS TSO "SUBMIT '"tosub'"
xx=OUTTRAP('OFF')
if pw ^= '***' then do
  "ALLOC DA('"tosub"') F(SUB) SHR REUSE"
  "EXECIO * DISKR SUB (STEM LL. FINIS"
  do a=1 to ll.0
    if strip(ll.a) = pw then ll.a = '<PASSWORD>'
  end
  "EXECIO * DISKW SUB (STEM LL. FINIS"
  "FREE F(SUB)"
end
do a=1 to jobm.0
  if pos(jn,jobm.a) ^= 0 & pos('SUBMITTED',jobm.a) ^= 0 then do
    sp = pos('(',jobm.a) ; lu = pos(')',jobm.a)-sp+1
    jid = substr(jobm.a,sp,lu)
  end
end
workjn = jn||jid ; ended = 0
do until ended = 1
  xx=OUTTRAP('jobx.')
  ADDRESS TSO "STATUS "workjn
  xx=OUTTRAP('OFF')
  do z=1 to jobx.0
```

```

        if pos(jid,jobx.z) = 0 & pos(jobendmsg,jobx.z) = 0 then ended =
1
    end
end
if jout = '***' then Return workjn
say time() ; say time() ' ... Verification job 'wjtyp
ww="xx=SYSDSN('"'jout"'"")
INTERPRET ww
if xx = 'OK' then do
    say '>>>>>>>' ; say '>>>>>>> Output dataset "'jout'"'
    say '>>>>>>> status is "'xx"'. To Control.' ; say '>>>>>>>'
    Return 100 ; end
"ALLOC DA('"jout"') F(DD1) SHR REUSE"
"EXECIO * DISKR DD1 (STEM REC. FINIS"
"FREE F(DD1)"
/*****
/* Routine of search error messages into NvDM utility output      */
/* dataset. It select only error records.                          */
/*****
E = 0 ; PD = ''
DO I = 1 TO REC.0
    A = POS('NDM',REC.I) ; If A = 0 then Iterate
    B = SUBSTR(REC.I,A+3,4) ; If DATATYPE(B) = 'NUM' then Iterate
    C = SUBSTR(REC.I,A+7,1) ; If C = 'E' & c = 'T' then Iterate
    D = SUBSTR(REC.I,A,8) ; If D = 'NDM3324E' then Iterate
    If D = 'NDM2647E' then do
        PD = D ; Iterate ; end
    If D = 'NDM2605E' & PD = 'NDM2647E' then Iterate
    If D = 'NDM2003T' & cpwf = 1 & e = 0 then do
        e = -1 ; leave ; end
/***** Message error found *****/
    E = E + 1 ; ER.E = strip(rec.i)
END
If E > 0 then do
    say '>>>>>>>' ; say '>>>>>>> The job 'wjtyp' ended with errors.'
    say '>>>>>>> The error messages are:'
    do a=1 to e
        say left(er.a,79)
    end
    say '>>>>>>>' ; say '>>>>>>> To control.' ; say '>>>>>>>' ; E = 100
end
If E < 0 then do
    say '>>>>>>>' ; say '>>>>>>> Password is incorrect. To control.'
    say '>>>>>>>' ; end
if e = 100 & e = -1 then e = workjn
RETURN E

```

## C-list NSPC040

```

/* REXX */
/* NvDM Send Procedure - C-List NSPC040 Called by c-list main NSPC010

```



```

Execute define NvDM plan
The functions are: - create job of utility TCFMAINT;
- call program NDMCOUNT to generate NvDM plan-id name;
  - display to user of NvDM plan name;
*/
trace ?o
ARG wtms,wlsnd,nonej,wjout,wjpln,desnam,jname,ndes,pcount
/***** Read destinations NvDM node. Read job skeleton. *****/
"ALLOC DA('desnam') F(DST) SHR REUSE"
"EXECIO * DISKR DST (STEM DS. FINIS"
"FREE F(DST)"
do a=1 to ds.0
  UPPER ds.a
end
i = 0
do a=1 to ds.0
  c1 = left(strip(ds.a),1)
  if c1 = '*' | c1 = ' ' then iterate
  i = i + 1 ; dest.i = word(ds.a,3) ; fas.i = word(ds.a,2)
end
"ALLOC DA('nonej') F(JS2) SHR REUSE"
"EXECIO * DISKR JS2 (STEM JD. FINIS"
"FREE F(JS2)"
jn = left(jname,9) ; jd.1 = '/'||jn||substr(jd.1,12)
do a=2 to jd.0
  pn = pos('*FUNCT*',jd.a)
  if pn = 0 then leave
end
jd.a = left(jd.a,pn-1)'SUBMIT'substr(jd.a,pn+7)
jd.a = left(jd.a,lastpos(', ',jd.a)-1)||'"'
do a=2 to jd.0
  pn = pos('*OUT*',jd.a)
  if pn = 0 then leave
end
jd.a = left(jd.a,pn-1)||wjout||substr(jd.a,pn+5) ; jd.a = left(jd.a,80)
/***** Read dataset with list object to send *****/
"ALLOC DA('wlsnd') F(WL2) SHR REUSE"
"EXECIO * DISKR WL2 (STEM LL. FINIS"
"FREE F(WL2)"
io = jd.0 ; ph = 0 ; b15 = left(' ',15)
do a=1 to ndes
  qmand = 0 ; fd = 'NO'
  do b=1 to ll.0
    if left(strip(ll.b),1) = '*' then iterate
    ss = word(ll.b,a)
    if ss = 'S' then do
      dsn = word(ll.b,ndes+1) ; dsn = translate(dsn,'. ','()')
      dsn = strip(dsn) ; dsn = dsn||'.'||a ; pdd =
pos('.SHIPMS.',dsn)
      if pdd > 0 then dsn =
left(dsn,pdd)||wtms||substr(dsn,pdd+7)
      if fd = 'NO' then do

```

```

fd = 'SI' ; ph = ph + 1 ; sigla = right(dest.a,4)
filcount = pcount||sigla ; mm=MSG('OFF')
"FREE F(ZFILE1)" ; "FREE F(ZFILE10)"
do xx=1 to 500
  "ALLOC F(SYSOUT) DUMMY"
  "ALLOC DA('"filcount"') F(ZFILE1) SHR"
  "ALLOC DA('"filcount"') F(ZFILE10) SHR"
  if rc = 0 then leave
end
"CALL 'NVDM162.USER.LOAD(NDMCOUNT)'"
numero = rc
"FREE F(ZFILE1)" ; "FREE F(ZFILE10)"
mm=MSG(mm)
if sign(numero) = -1 then do
say time() ; say time() ' Call in error. Return Code
"'numero"'.'
  say time() ; Return 100 ; end
tsfx.ph = sigla||right(numero,4,'0') ; io = io + 1
jd.io = ' DELETE PLAN NAME='||tsfx.ph
jd.io = left(jd.io,80) ; io = io + 1
jd.io = ' DEFINE PLAN NAME='||tsfx.ph||',AUTOSUB=YES'
jd.io = left(jd.io,80) ; io = io + 1
jd.io = ' DEFINE PHASE NAME='fas.a',NODE='dest.a',STATUS=RELEASE'
jd.io = left(jd.io,80)
end
else do
  ugu = 0
  do c=1 to qmand
    if dsn = mand.c then do
      ugu = 1 ; c = qmand
    end
  end
  if ugu = 1 then iterate
end
qmand = qmand + 1 ; mand.qmand = dsn ; io = io + 1
jd.io = ' SEND RESOURCE RESTYPE=DATASET,FCTLNAME=NDM,'
jd.io = left(jd.io,71)||'X' ; jd.io = left(jd.io,80)
io = io + 1 ; jd.io = b15||'NAME='||dsn||','
jd.io = left(jd.io,71)||'X' ; jd.io = left(jd.io,80)
io = io + 1 ; jd.io = b15||'ORIGIN=DSX,DISP=(OLD,DELETE)'
jd.io = left(jd.io,80)
end
end
end
io = io + 1 ; jd.io = left(' END',80) ; io = io + 1 ; jd.io = left('/
'|'*',80)
io = io + 1 ; jd.io = left('//BATCHPW DD *',80) ; io = io + 1
jd.io = left('<PASSWORD>',80) ; io = io + 1 ; jd.io = left('/'|'*',80)
Call DelTCF
/***** Write job define NvDM Plan
*****/

```

```

"ALLOC DA("wjp1n") F(JPL) SHR REUSE"
"EXECIO * DISKW JPL (STEM JOUT. FINIS"
"FREE F(JPL)"
Return
DelTCF:
Do a=1 to io
  px = pos(' EXEC ',left(jd.a,20))
  if px = 0 & left(jd.a,3) = '//'||'*' then leave
  jout.a = jd.a
end
ist = a ; jout.a = '//STEP0 'substr(jd.a,px)
jout.a = left(jout.a,80) ; i = a ; str1 = 'FUNCTION=SUBMIT'
add1 = 8 ; str2 = 'TCFMAINT' ; add2 = 15 ; flg = 0
do a = ist + 1 to io
  if (left(jd.a,3) = '// ' & flg = 1) | left(jd.a,3) = '//'||'*' then
iterate
  i = i + 1 ; pn = pos(str1,jd.a)
  if pn = 0 then do
    jout.i = left(jd.a,pn+add1)||str2||substr(jd.a,pn+add2)
    jout.i = left(jout.i,80) ; flg = 1 ; iterate
  end
  if left(jd.a,10) = '//SYSPRINT' then do
    jout.i = '//SYSPRINT DD SYSOUT=*' ; jout.i = left(jout.i,80)
    iterate
  end
  jout.i = jd.a
  if left(jd.a,7) = '//SYSIN' then leave
end
do a = 1 to ph
i = i + 1 ; z = ' ERASE PLAN NAME='||tsfx.a
jout.i = left(z,80) ; ono = tsfx.a
say time() '>>>' ; say time() '>>>' ; say time() '>>> The PLAN name is:
'ono'.'
say time() '>>>' ; say time() '>>>' ; end
i = i + 1 ; jout.i = left(' END',80) ; i = i + 1
jout.i = left('//'||'*',80) ; i = i + 1 ; jout.i = left('//BATCPW DD
*',80)
i = i + 1 ; jout.i = left('<PASSWORD>',80) ; i = i + 1
jout.i = left('//'||'*',80)
do a=ist to io
  i = i + 1 ; jout.i = jd.a
end
jout.0 = i
Return

```

## C-list NSPC045

```

/* REXX */
/* NvDM Send Procedure - C-List NSPC045 Called by clist main NSPC010
Execute management of procedure log
The functions are: - Write into log datasets all files to send

```

```

*/
Trace ?o
ARG wlsnd,wlogs,ndes
log1 = wlogs||'1' ; log2 = wlogs||'2'
"ALLOC F(LLOG1) DA(''log1'') SHR"
"EXECIO 1 DISKR LLOG1 (STEM LR FINIS"
if rc = 0 then do
    w1 = 0 ; frc.1 = 1 ; end
else do
    frc.1 = 0 ; w1 = word(LR1,1)
    if datatype(w1) = 'NUM' | length(w1) = 11 then w1 = 0
end
"FREE F(LLOG1)"
"ALLOC F(LLOG2) DA(''log2'') SHR"
"EXECIO 1 DISKR LLOG2 (STEM LR FINIS"
if rc = 0 then do
    w2 = 0 ; frc.2 = 1 ; end
else do
    frc.2 = 0 ; w2 = word(LR1,1)
    if datatype(w2) = 'NUM' | length(w2) = 11 then w2 = 0
end
"FREE F(LLOG2)"
if w2 > w1 then log = '2' ; else log = '1'
nome = wlogs||log ; result = 0
if frc.log = 1 then Call Primi2
if result = 1000 then Return
si.1 = 'TRENTO' ; si.2 = 'MILANO' ; si.3 = 'ROMA' ; si.4 = 'PALERMO'
si.5 = 'FIRENZE' ; si.6 = 'GENOVA' ; si.7 = 'NAPOLI1' ; si.8 = 'NAPOLI2'
b3 = left(' ',3,' ') ; b6 = left(' ',6,' ') ; b15 = left(' ',15,' ')
"ALLOC DA(''wlsnd'') F(KLW) SHR REUSE"
"EXECIO * DISKR KLW (STEM LL. FINIS"
"FREE F(KLW)"
righe.1 = 'User: 'userid()' - Date: 'date('E')' - Time: 'time()'
aa = 1 ; qtd = 0
do a=1 to ll.0
    if left(strip(ll.a),4) = '*M*' then do
        nora = word(ll.a,ndes+1) ; nobb = translate(nora,' ','.(.))
        ndel = ''
        if word(nobb,3) = 'LOADS' then do
            nora = word(nobb,1)||'.'||word(nobb,2)||'.LOADS.'||word(nobb,4)
            ndel = word(nobb,1)||'.SHIPMS.LOAD'
        end
        if ndel = '' then do
            qtd = qtd+1 ; dele.qtd = nora ; end
        dests = ''
        do b=1 to ndes
            if word(ll.a,b) = 'S' then dests = dests si.b
        end
        aa = aa + 1
        righe.aa = b3||'Send of ''||nora||'' to' ||dests ; mems = 0
    end
end

```

```

else do
  aa = aa + 1 ; mems = mems + 1
  if mems = 1 then righe.aa = b6||'Members: '||word(11.a,2)
    else righe.aa = b15||word(11.a,2)
  qtd = qtd + 1 ; dele.qtd = ndel||'('||word(11.a,2)||')'
end
end
aa = aa + 1 ; righe.aa = left('-',70,'-')
/**** If log is full then switch dataset. Write procedure log ****/
ww="xx=LISTDSI('"'nome"'"")
interpret ww
if SYSEXTENTS = 16 & SYSUSED=SYSALLOC then do
oldlog = nome
if log = 1 then log = 2 ; else log = 1
nome = wlogs||log ; say time() ; say time() ' Log dataset "'oldlog'" is
full.'
say time() ' Change log with "'nome"'. ' ; say time() ; result = 0
Call Primi2
if result = 100 then Return
end
do xx=1 to 500
  "ALLOC F(LLOGH) DA('"'nome"'"') MOD"
  if rc = 0 then leave
end
"EXECIO "aa" DISKW LLOGH (STEM RIGHE. FINIS"
zz=rc
"FREE F(LLOGH)"
if zz = 0 then do
  say time() ; say time() ' Write error for log dataset "'nome"'. '
  say time() ' Return Code 'zz'.'. ' ; say time() ; end
Return
Primi2:
/***** Init procedure log *****/
zz = 0 ; Dati = left(date('B'),6,'0')||left(time('S'),5,'0')
rigel = Dati||' - Init at '||date('E')' time 'time()'. '
rige2 = left('-',70,'-')
"ALLOC F(LLOGN) DA('"'nome"'"') SHR"
"EXECIO 2 DISKW LLOGN (STEM RIGE FINIS"
zz = rc
"FREE F(LLOGN)"
if zz = 0 then do
  say time() ; say time() ' Write error for init log dataset "'nome"'. '
  say time() ' Return Code 'zz'.'. ' ; say time() ; zz = 100 ; end
Return zz

```

*(Editor's note: the next issue of TCP/SNA Update will contain sample tables, skeletons, and programs, and the ISPF panels for this procedure.)*

---

*Espedito Morvillo*  
*(Italy)*

© Xephon 2000

---

## **E-business and Web-to-host integration**

The turn of the millennium will be remembered for years to come not just for the hullabaloo related to Y2K, but also as the time when e-business and e-commerce finally came of age (where e-business refers to the entire range of business-to-business (b2b) interactions over the Web, with e-commerce dealing with just the process of buying and selling items over the Web). Twelve billion dollars' worth of e-commerce was carried out in the last two months of 1999. This exceeded all expectations, and was four times more than the year before.

E-commerce, which focuses on the consumer side of the Web, invariably gets all of the media and public attention; most people can relate to it in some form, even if it's just by reading the news stories about on-line shopping and electronic investing. But e-commerce is actually just a small subset of the entire e-business revolution. What's more, it's the speed, efficiencies, and economies realized by e-business when it comes to 'b2b' interactions that make e-commerce viable. Traditional b2b processes are too cumbersome and slow to keep up with the dynamic demands of an increasingly Web-oriented society. Since e-business depends on employees of various companies being able to rapidly and securely access computer applications running on each others' systems, safe and reliable Web-to-host integration is a prerequisite for e-business – and thus also for e-commerce.

A good example of e-business is just-in-time supply chain management between a manufacturer and its suppliers. This works by ensuring that most of the requisite queries, inventory checking, order processing, and funds transfer is done electronically, across the Web, b2b between the various companies. Often, the companies involved (especially those in the Fortune 5000) will have mainframes, but sometimes they won't, in which case the employees will need to be able to access certain mainframe applications being run by the other companies. Those without mainframes will not typically have the host emulation software required; nor will they have the experience to use conventional 'green-on-black' mainframe screens. The companies with mainframes therefore need to provide a transparent, 'no-hassle', user-friendly

scheme whereby the other partners can gain mainframe access. Web-to-host technology, typically in the form of 3270-to-HTML conversion or applet-based terminal emulation, is the only way to rationally and cost-effectively address this any-to-any access required for e-business.

The heavy toll exacted on data centre operations by Y2K preparations in terms of both expenditure and effort has left data centres behind when it comes to e-commerce and e-business initiatives. E-commerce efforts in particular were being implemented outside the bounds of the traditional data centre, while the data centre staff were bogged down in the throes of Y2K. The good news, however, is that the growing momentum around e-business, which means that by 2003 most companies will have no choice but to participate in e-business in some form, will force data centre professionals to quickly implement Web-to-host integration technology. This technology, where applicable, will also mean that companies can leverage their data centre applications and expertise to pursue e-commerce.

There are no technical reasons for mainframes to be out in the cold when it comes to e-commerce or e-business, especially with IBM ensuring that OS/390's built-in WebSphere Web- and Application-Server has all the necessary features, in an unparalleled scalable form, to facilitate e-commerce (eg payment gateway option and inclusion of the Secure Electronic Transaction (SET) protocol developed by VISA and MasterCard). And at least 55 other companies – including Attachmate, Microsoft, Novell, Eicon, Bluestone, Inprise, Neon, NetManage, Intelligent Environments, and ICOM – provide a plethora of germane Web-to-host integration technology ranging from 3270/5250-to-HTML conversion to full-blown, Java-based Application Servers bristling with object-oriented methodology for expediting application development and deployment.

Despite the distraction caused by Y2K, some mainframe stalwarts such as airlines, financial institutions, banks, and service companies already have a high-profile presence on the Internet. Large mainframe customers such as American Airlines, Charles Schwab, FedEx, and Fidelity Investments already permit the general public to conduct secure, revenue-generating financial transactions over the Web. However, in many instances, access to mainframe data is not direct. Although proven technology, complete with the necessary security

features to facilitate such direct access, is readily available, many corporations have so far postponed the provision of direct mainframe access by replicating some of the data on NT or Unix servers. This replication is generally regressive, since it results in mismatched data that has to be regularly synchronized and separately backed up via batch-oriented processes.

The remainder of this article looks at the current major trends in IBM networking and describes the pertinent technology, all of which is already available, that will enable mainframes to be better Web enabled so that they can play a pivotal role in e-commerce in the future.

## CURRENT IBM NETWORKING TRENDS

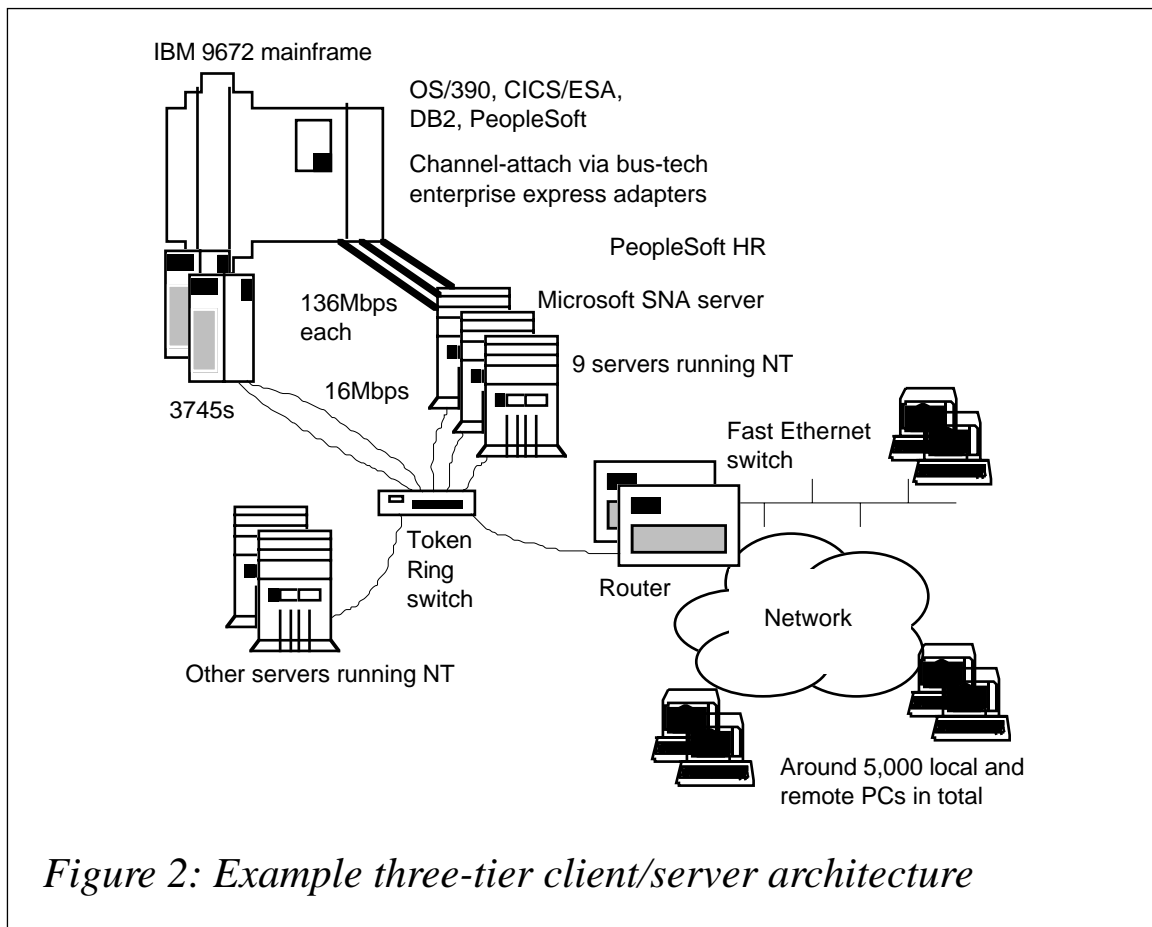
The key current IBM networking trends are as follows:

- Nearly all mainframe shops now have a production intranet, as well as a presence on the Internet in the form of a home page.
- Over 75% of mainframes running MVS or OS/390 already have TCP/IP installed, and this number will grow to at least 95% by 2001. OS/390 has TCP/IP as well as a full-function Web Server in WebSphere built-in.
- IBM's 2.6Gbps-capable OSA-Express, now available for Generation 5 and 6 System/390s, is by far the fastest and most efficient network-to-mainframe interface, and makes the mainframe into a highly compelling tn3270(E) and Web server.
- At least 70% of corporate data still resides in the data centre, rather than on Web, NT, or Unix servers.
- SNA-only and multiprotocol networks are being rapidly transitioned over to TCP/IP-centric networks, with DLSw being widely used to transport SNA end-to-end across the network, and nearly 11 million users rely on tn3270(E) to access SNA applications without using any SNA software at the client.
- The scope of APPN and its successor HPR will be restricted to the data centre – and in the case of corporations with multiple data centres, to certain inter-data centre interactions such as bulk data



transfer using ACF/VTAM Version 4 Release 4's High Performance Data Transfer (HPDT) option. IBM's Enterprise Extender capability, which is now available in OS/390, IBM Communications Server for NT, NetWare for SAA 4, and IBM 22xx bridge/routers as well as Cisco bridge/routers, which permits SNA/HPR traffic to be routed across an IP backbone, is an alternative to DLSw that has the edge over the now-entrenched DLSw in two specific scenarios:

- The first of these scenarios is if SNA-based routing dictated by an SNA application name is sought in a network which has multiple data centres.
  - The second scenario is if there is a real need for SNA LU 6.2-based Class Of Service (COS) Traffic Prioritization from the desktop to the application.
- Although the future of mainframe SNA applications is secure well beyond 2005, these mission-critical applications are being routinely augmented with ERP applications from SAP, PeopleSoft, IBM, and BAAN that are being deployed in a three-tier client/server architecture with the mainframe acting primarily as a data rather than an application server. One solution which is growing in popularity is to directly channel-attach the application server, which is invariably an NT platform, to the mainframe using PCI bus-based 17MBytes/sec (136Mbps) ESCON II adapters (eg IBM's Netfinity ESCON or Bus-Tech's EnterpriseExpress adapters). Using such an adapter, which has a price of around \$12,500, can in many cases reduce response times by an order of magnitude; say from 35 seconds down to 4 seconds. Figure 2 shows the actual channel-attached three-tier client/server configuration used by a large mid-West hospital chain in the USA, for its PeopleSoft Human Resources applications – where channel-attachment was key to ensuring acceptable response times.
  - Although the installed base has not yet shrunk perceptibly, Token-Ring – like SNA-centric networking – is now in decline, with most corporations having already decided to move to the much more economical and ubiquitous 100Mbps Fast Ethernet, despite IBM's, Olicom's, and Madge's eleventh-hour attempt of



coming up with a 100Mbps High-Speed Token-Ring (HSTR) alternative.

- Although over a million ‘seats’ of browser-based access to SNA solutions were sold both in 1998 and 1999, in some form or another, the prevalent transition at the desktop was not towards browser-based, ‘thin-client’ solutions. Instead, the major moves were from 16-bit to 32-bit desktop emulators – and from SNA-LAN Gateway-based access (eg Novell’s NetWare for SAA) to tn3270(E)-based, TCP/IP-centric access. But this is changing. Thin-clients are gaining renewed interest in 2000 – especially given their practicality and convenience when it comes to e-business.
- ATM, although not the ultimate networking panacea that IBM tried to make it in the 1993 to 1996 timeframe, is often being used at speeds of between 155Mbps and 622Mbps at the LAN backbone and LAN server level to increase LAN performance. In North America, the public carriers are now offering ‘low-speed’ ATM

WAN services, with speeds starting at T-1 rates, as direct, highly scalable, multimedia-capable alternatives to highly popular Frame Relay services.

- Encouraged by the availability of full-function Virtual Private Networking (VPN) capability on low-cost routers, such as IBM's 2212 and 2210, increasing numbers of corporations are actively exploring the possibility of using the Internet as a ultra-low-cost source of WAN bandwidth.

#### INTEGRATING SNA APPS WITH THE WEB AS A PRECURSOR TO E-COMMERCE

Browser-based access to SNA technology has now been around, in proven form and with support for printing, since 1998. The four primary advantages of browser-based access over those of conventional emulator approaches are as follows:

- It greatly minimizes the cost and effort of client software installation, maintenance, and updating by providing a server-based thin-client solution. In the case of applet-based solutions (eg IBM's Host On-Demand or Eicon's Aviva for Java), the applet can be dynamically downloaded from a Web Server. With today's browsers (ie Version 4 of Netscape Navigator and Microsoft's Internet Explorer), the applet can, moreover, be cached on the hard disk of a PC/workstation to remove the need for repeated downloads across the network. Automatic version checking each time the cached applet is invoked ensures that caching does not preclude users from immediately getting new versions of the applet that are installed at the server. This dynamic download, with caching, capability could save around \$200 per desktop per year in client software maintenance costs.
- 3270/5250-to-HTML conversion (eg Eicon's Aviva Web-to-Host Server or Novell's HostPublisher) is the thinnest of thin client solutions. The conversion is done on a data centre-resident server. All that the user needs is a standard Web browser. This type of host access is ideal for allowing the general public or e-business users ready, but secure, access to host applications. This thinnest of thin clients solutions eliminates all host access software-

related costs at the client side, and liberates and facilitates host access by ensuring that any authorized user with a Web browser can now interact with a mainframe or AS/400 application – provided, of course, that they have the relevant security clearances.

- It offers economical and easy options for dramatically rejuvenating and streamlining the very dated and anachronistic 3270 user interface of most SNA mission-critical applications. User interface rejuvenation is a must if SNA applications are to be opened up for public use over the Internet for e-commerce-related applications such as home investing, personal travel reservation, home banking, and on-line mail-order.
- It Web-enables SNA applications and allows them to be readily accessed, using just a standard browser, across an intranet, an extranet, or the Internet.

There are currently four very different ways to achieve browser-based access:

- 3270-to-HTML conversion – eg Novell’s HostPublisher or Eicon’s Aviva Web-to-Host Server.
- Applet-based tn3270(E) emulation – eg IBM’s Host On-Demand or Hummingbird’s HostExplorer Web.
- Application-specific gateways such as IBM’s CICS Web Interface.
- Programmatic solutions such as IBM’s Host Publisher Version 2 or Bluestone’s Sapphire/Web application server.

3270-to-HTML, the very first form of browser-based access to SNA, is ideally suited for use across the Internet in that it always provides some level of automatic user interface rejuvenation, and furthermore does not require an applet to be downloaded. Consequently, this technology, readily available at relatively low costs from multiple vendors, should be evaluated as a way of Internet-enabling existing applications. This approach revolves around having 3270 screens that are dynamically converted to HTML pages being sent out by a standard Web Server. Security *per se* is thus integrated, in that 3270-to-HTML benefits from the standard security features, such as Secure Sockets Layer (SSL), available on today’s Web Servers. SSL is a highly proven and widely-used security scheme that provides

authentication and arbitrates the use of end-to-end encryption. In addition to SSL, most of today's leading 3270-to-HTML solutions include session ID and sequence number facilities to ensure end-to-end session integrity, to preclude the possibility of 3270 'screens' inadvertently being shown to the wrong user.

In addition to session integrity safeguards, early 3270-to-HTML conversion offerings lacked certain functions such as support for function keys, cursor-select (ie light pen feature), emulation of input field attributes, and unsolicited host output. The good news is that products such as Novell's HostPublisher and Eicon's Aviva Web-to-Host Server have overcome these limitations. Consequently, 3270-to-HTML is a full-function, 'no-excuses' technology ideally suited for easily extending mainframe SNA applications onto the Internet.

In early 1998, applet-based emulation overshadowed 3270-to-HTML conversion by offering a broader repertoire of functions, including end-to-end integrity through the use of persistent TCP connections. Today, the gap in functionality is minimal, if it exists at all. A demarcation is therefore now emerging as to how these two different schemes should be best exploited. In general, 3270-to-HTML is the obvious choice for Internet access. On the other hand, applet-based emulation, which is in effect tn3270(E) emulation via a thin client, is ideally suited for intranet and extranet applications – where there is still a need for high-performance, 'green-on-black' 3270 emulation for 'heads down' applications such as data entry. The primary advantage of using an applet-based emulator in this context is the elimination of client software installation and maintenance costs as described above. Some kind of user interface rejuvenation is possible with all applet-based schemes, with the latest trend being the provision of 'auto-GUI' features such as IBM's ResQ!Net that will dynamically face-lift 3270 screens based on a pre-defined set of rules.

Both 3270-to-HTML and applet-based emulation have gained an increasing following as easy-to-implement, 'out-of-the-box' solutions for Web-enabling SNA applications. However, it is now very clear that the strategic means for Web-enabling SNA applications and leveraging them for e-commerce, post the Y2K crunch, is going to be the use of bespoke programmatic solutions.

## PROGRAMMATIC ACCESS – THE ULTIMATE SOLUTION

Most of today's venerable SNA applications will, by the year 2005, be front-ended, or be in the process of being 'end gamed', by a new breed of applications. Although this kind of claim has been being made about SNA for the last 20 years, this time the claim is not actually being made by detractors of SNA, but instead by those who intend to leverage the investment in SNA as long as possible. The current trend toward user interface rejuvenation of SNA applications is but a precursor to these applications being pushed into the background to act as transparent data servers to applications that will be performing expanded business logic functions in addition to interacting with the users via a Web-inspired user interface. The SNA applications and the data accessible through them will also become back-ends to the ERP and Online Analytical Processing (OLAP) applications that are now in vogue to help enterprises extract as much knowledge as possible from the data they possess.

Today's user interface rejuvenation techniques that make it possible to record and capture the keystrokes required to establish an SNA session and then navigate through various 3270 screens will be expanded so that this 'access logic' can be made into reusable JavaBeans. Many of the new 'front-end' applications will use such JavaBeans to obtain the mainframe data they require via the SNA applications. IBM's Host On-Demand JavaBeans that are now available pave the way for this new and exciting methodology.

Whichever way you look at it, programmatic access to SNA applications from browser-oriented clients will be the final chapter in the eventful, three-decade-long evolution of SNA access. User interface rejuvenation, even when it involves considerable screen consolidation and rationalization, still relies on the original SNA application to perform all of the business logic. Programmatic access takes you to the next logical step – the ability to synthesize the data accessible through, as well as the business logic inherent in, an existing mission-critical application or even multiple applications into the body of a brand new application; eg an e-commerce application.

Programmatic Web-to-data centre access implies some client software on the client PC/workstation, be it browser, applet, or application, that is interacting with a data centre-resident application or database

across the network. Some form of intermediary software, irrespective of whether or not it is strictly in the class of software deemed as 'middleware', is always going to be required to make this happen. Today there is a very wide and disparate range of options as to what this intermediary software can be. These options include:

- 3270 datastream-oriented schemes such as IBM's 'Host On-Demand' Host Access Class Library (HACL), the Host On-Demand JavaBeans, Attachmate's Enterprise Access Class Libraries (EACL), the Open Host Interface Objects (OHIO) endorsed by both IBM and Attachmate, as well as the old faithful HLLAPI interface, which is universally available with tn3270(E) and 3270 clients.
- LU 6.2-based program-to-program schemes through the APPC or CPI-C interfaces available with some ip3270 emulators such as Attachmate's EXTRA! and IBM's PComm.
- Traditional, pre-Web, middleware solutions such as IBM's MQSeries that can be used with browser-based or browser-invoked clients to realize programmatic Web-to-data centre access.
- New-wave, Web-oriented middleware solutions such as IBM's Host Publisher Version 2 or Bluestone's Sapphire/Web Application Server, which comes complete with high-end database access features and contemporary application development tools.
- ODBC- or DRDA-based capability, such as that available within Microsoft's Babylon or with Cisco/StarQuest StarSQL, for directly accessing data located with DB2-like relational databases.
- IBM VSAM file (ie 'flat file') resident data access schemes such as that available within Microsoft SNA Server Version 4.0.
- Application-specific solutions such as IBM's CICS Gateway for Java and the CICS Web Interface.

Programmatic solutions will differ significantly from Web enablement along the lines of 3270-to-HTML or applet-based emulation, in that they will require considerable amounts of programming effort. However, programmatic solutions are invariably the only way to ensure the totally seamless integration of SNA applications with new-wave Web applications. While 'out-of-the-box' solutions, with

rejuvenation capabilities, can produce dramatic results across the Web, in the end there's inevitably some hesitation or 'stutter' that betrays their jury-rigged nature. With so much at stake with e-commerce, and with competition fiercer than ever, corporations will want to present the best face possible to their customers. The slick interfaces sought to expedite e-commerce transactions will only be possible via programmatic solutions.

#### WHAT TO EXPECT IN THE PERIOD 2000 TO 2003

With Y2K now safely negotiated and consigned to history, the major networking themes for the coming few years are likely to be:

- A relentless consolidation towards TCP/IP-centric networking backbones, with the use and significance of intranets and extranets continuing to grow and mature.
- E-business and e-commerce becoming major business drivers that dictate the need for tight Web-to-host integration.
- The rapid adoption of browser-based technology to reduce networking costs, facilitate e-business, and, where applicable, to open up existing data centre applications for public consumption (eg travel reservation, home banking, on-line investing etc).
- An increase in the deployment of TCP/IP on mainframes, complete with mainframe-based TCP/IP management, to pave the way for a strategic TCP/IP-centric, end-to-end, mainframe-to-desktop networking fabric for the future, with the OSA-Express becoming the strategic means of attaching 21st century networks to new System/390 mainframes.
- An interest in evaluating the OS/390 WebSphere as a highly scalable and centralized Web Server and application server – in parallel with existing NT- or Unix-based Web Servers.
- A growing momentum in the exploitation of the 'near zero cost' Internet bandwidth for private corporate use for applications such as global remote access, telecommuting, remote office connectivity, and e-mail through the use of VPN or browser-based access solutions.



- A marked presence of Fast Ethernet within what were hitherto Token-Ring environments, as corporations opt for Fast Ethernet as they procure new PCs/workstations, move to new buildings, or upgrade their wiring closet equipment to handle the additional data volumes precipitated through the widespread use of graphic-intensive intranet applications.
- An initial evaluation of Web-oriented application servers as strategic tools for kick-starting e-commerce after the Y2K crunch.
- A continued battle between Java and ActiveX for domination, with Java offering platform independence (which is not that crucial as corporations standardize on Windows) and ActiveX touting tighter integration with Windows.

## SUMMARY

Y2K acted as a ‘speed bump’ and slowed down Web-to-host integration even though compelling technology had been readily available from over 50 credible companies since 1998. This meant that the data centre was not the initial focus of e-commerce activity. However, things are now changing. E-business is not only a prerequisite for e-commerce but it is also a superset of e-commerce embracing a much broader sphere of global commerce. The good news is that all the necessary technologies to enable the data centre to play an active role in e-business and e-commerce already exist and are proven. Furthermore, nearly a thousand companies around the world are already gainfully using this technology. In the short-term, browser-based access solutions will pave the way for SNA Web enablement with 3270-to-HTML offering the best solution for Internet applications, while applet-based emulation is the ideal choice for intranets. In the mid- to long-term, it will be programmatic access solutions that rule the roost and that will usher SNA applications written in the 1980s into the world of 21st century e-business and e-commerce. By then, SNA-only networks, Token-Ring, and ATM to the desktop will be legacy technologies. However, the conservatism and inertia inherent in IBM networking mean that you’ll still find Token-Ring networks, and possibly even the odd SNA-only one, ten years from now – if you look hard enough.

---

*Anura Gurugé*  
*Strategic Analyst (USA)*

© Xephon 2000

---

## Information point – reviews

### CANDLE CCR

Many software vendors offer newsletters highlighting their products in use at customer sites. Few reach the 1980s' rule of thumb goal for newsletters of 50% non-self-serving material, but *Candle Connections* is one of the better ones – see <http://www.candle.com/news/publications/connections>.

But it's *Candle Computer Report* (CCR) that really stands out in a class of its own – see <http://www.candlepubs.com>. Since it was first published in 1978, it has included major technical articles in each issue, typically on performance issues and/or technical architecture of major IBM, and even some non-IBM, systems software products. As you might expect, the focus is on areas where Candle has its own products, which means that non-mainframe topics have been added in recent years. But most articles don't even mention Candle's own products.

Obviously, not every issue has an article on TCP/IP or SNA, but the November/December 1999 lead article, 'Effects of E-Business on the Mainframe', looks at OS/390 from an e-business perspective, with considerable material on communications and network trends. And an article entitled 'The Emergence and Security of Virtual Private Networks' led the July 1999 issue, offering an in-depth technical discussion of VPN technology.

### ANALYST REVIEWS AND MANUALS

Another area of the Candle Web site that looks promising is the Analyst Reviews section, which is just getting started with a few reports. Divided into seven topics, it's a collection of studies by research firms on Candle products – see:

<http://www.candle.com/news/analystreviews.html>.

Vendors are beginning to offer, on the Internet, all the product manuals you're ever likely to need. Typically, you have the option of reading

them on-line or downloading them to your workstation. Candle is no exception:

<http://www.candle.com/productservices/download/documents/techdocs/index.html>

provides the entry point, along with instructions on use. Note that the free Adobe Acrobat Reader

<http://www.adobe.com/products/acrobat/readstep.html>

is required to read the manuals, whether you choose to read them on-line or download them to your workstation.

## CISCO

Despite the amount of information Cisco offers on its Web site, it's fairly easy to find it from the home page, <http://www.cisco.com>. In the Service & Support section, the Technical Documents link is the place to go for traditional documentation, from brochures to hot tip sheets and manuals. The first link you will see leads you to the Documentation Home Page, which begins with links that describe how to use the Documentation CD-ROM. (By the way, don't let 'CD-ROM' scare you off. The entire contents are available on-line on the Internet without even the need to have a CD-ROM drive or Adobe Reader on your workstation.)

As it says in *Welcome*, you can bookmark this page – <http://www.cisco.com/univercd/home/home.htm> – and go directly there at any time in the future if you prefer to avoid the navigation from the Cisco home page. But, if you do this, you'll miss all the other information along the way.

## PARALLEL SYSPLEX

Have you ever needed to know how much space you'll need in a parallel sysplex Coupling Facility (CF)? The Coupling Facility Structure Sizer Tool (CFSIZER) <http://www.s390.ibm.com/cfsizer> will do most of the calculations for you. For example, you enter the number of sessions and it calculates the CF structure size for VTAM generic resources, as well as displaying the Type, Name, INITSIZE,

and sample CFRM policy statements. VTAM MultiNode Persistent Session (MNPS) is promised soon, and may well be available by the time you read this.

#### IBM DICTIONARY

Have you ever wondered what an acronym stood for, what some cyberbabble meant, or how to spell a computer term? McGraw-Hill publishes *The IBM Dictionary of Computing*, but, for quick reference, IBM also offers its dictionary on-line, at <http://www.networking.ibm.com/nsg/nsgmain.htm>.

This seems to be a more up-to-date version than the one provided through IBMLink through its Help and WhatIs links. For example, a search through IBMLink for MNPS did not find anything, but the URL in the previous paragraph includes an MNPS entry under M.

#### DINOS WEB RING

A Web Ring may sound like a sinister underworld concept, but don't assume that it's just more evidence that the criminal element is the first to use the latest technology. A Web Ring is merely a centrally-managed structure where Web sites related solely by a common interest can link to each other. Once a site is accepted into a Web ring, the only major requirement is to place the ring's standard box on the home page of the Web site. The box is fairly large, usually contains an ugly animated logo, and provides a link mechanism to the Web Ring directory and other Web ring member sites.

The Dinos Web Ring bills itself as "a link between dinosaurs that have an interest in mainframe and other 'obsolete' technologies." See <http://www.webring.org/cgi-bin/webring?ring=dinos;list> for a complete list of sites.

#### THE SHARE TAPE

One Dinos site is the CBT Tape home page, where you'll find what's more commonly known as the SHARE Tape – see <http://>

www.cbttape.org. It gets that name because of the fact it's handed out free at SHARE conferences. Almost 500 copies of CBT423 were handed out in Anaheim in March.

CBT423 contains 472 files. Each file is one or more pieces of code that someone thought might be useful to others. There are five files with VTAM in their title.

You'll also find other tapes. A 295 file Overflow tape, the inevitable Updates tape to fix bugs found, the 455 file CBT249 from 1985, JES2 and JES3 tapes, and a couple of larger offerings, offered individually, such as UCLAmail. If you're looking for MXI, it's moved to another site:

[http://www.secltd.co.uk/mxi\\_download.htm](http://www.secltd.co.uk/mxi_download.htm)

## SHARE

Speaking of SHARE, that organization is now 45 years old and is still offering top-notch technical conferences. The SHARE Web site, <http://www.share.org>, lists all of the sessions from the most recent conference, and those from the next conference, if it's not too far into the future. Select Conferences from the left sidebar, then the conference you're interested in. From the right sidebar, under Agenda, select Online Version.

You'll see some of the Technical Tracks listed, and can select any of them to see all of the relevant sessions available. Further down the page, you can select sessions by day, session number, or project. But, near the top of the page, is the Search the Agenda box, which allows you to specify one or more keywords separated by AND, OR, or AND NOT. At March's conference in Anaheim, a search for VTAM listed 17 sessions, while 'sna OR tcp/ip' found 76.

If you've never attended SHARE, you really are missing an almost overwhelming amount of choice in technical sessions of interest. And don't let the titles fool you. "Selling Ducks on the Web" may sound like an old Marx Brothers movie, but e-commerce is the topic of this session on Ducks Unlimited's System/390-based Web site.

For someone new to NetView, what could be better than a session

entitled ‘NetView/390 for the New User’? And there are even a few chances to polish up your people skills. “When You Talk, Does Anyone Listen?” focuses on a form of communication that uses neither SNA nor TCP/IP protocol: speaking. It’s part of the PDEV (Personal Development) certificate program.

Finally, it should be pointed out that organizations, not individuals, are members of SHARE. Qualifying organizations must use an IBM computer system or operating system, and must send a representative to at least one SHARE conference a year to maintain their membership. Full details are available by clicking on Membership from the left sidebar on the home page and then the Become a Member link.

## GSE

In the US, GUIDE closed its doors and put its moral support behind SHARE. In Europe, the two merged to form GUIDE Share Europe (GSE). Its home page is at <http://www.gse.org>, and GSE UK has its own home page for the United Kingdom at <http://www.gse.org.uk>

As well as a few others of interest, GSE UK’s NetView Working Group is now Networks and Systems Management with its own home page at <http://www.gse.org.uk/wg/netview/nvxindex.htm>. You’ll also see links to the GSE UK Events Calendar at <http://www.gse.org.uk/calendar.htm>, which lists the locations and dates of each GSE UK working group meeting, as well as GSE conferences, symposiums, and other events.

## SUMMIT ON-LINE

If you can’t get to a conference, SUMMIT tries to provide an on-line equivalent – see <http://www.summitonline.com>. Although a few of the links to technical papers don’t work, there’s still plenty of great information here.

The White Papers link from the black bar near the top of the home page takes you to a choice of 10 categories, the first being Network Management. There are a total of 200 technical papers on the site to choose from.

Also from the home page, the Links link takes you to a page with an impressive number of links listed, especially in the Networking arena. Some of the sites being linked to are fairly amateur affairs, but HTML errors don't detract from the quality of the information presented.

---

*Jon E Pearkins*  
(Canada)

© Xephon 2000

---

## **Contributing to *TCP/SNA Update***

In addition to *TCP/SNA Update*, the Xephon family of *Update* publications now includes *CICS Update*, *MVS Update*, *VSAM Update*, *DB2 Update*, *RACF Update*, *AIX Update*, *Domino Update*, *MQ Update*, *NT Update*, *Oracle Update*, *SQL Server Update*, and *TSO/ISPF Update*. Although the articles published are of a very high standard, the vast majority are not written by professional writers, and we rely heavily on our readers themselves taking the time and trouble to share their experiences with others. Many have discovered that writing an article is not the daunting task that it might appear to be at first glance.

They have found that the effort needed to pass on valuable information to others is more than offset by our terms and conditions and the recognition they gain from their fellow professionals. Often, just a few hundred words are sufficient to describe a problem and the steps taken to solve it.

If you have ever experienced any difficulties, or made an interesting discovery, you could receive a cash payment, a free subscription to any of our *Updates*, or a credit against any of Xephon's wide range of products and services, simply by telling us all about it. For a copy of our *Notes for Contributors*, which explains the terms and conditions under which we publish articles, please write to the editor, Fiona Hewitt, at any of the addresses shown on page 2, or e-mail her at [fionah@xephon.com](mailto:fionah@xephon.com)

# TCP/SNA news

---

IBM has announced Version 6.1 of its SecureWay Communications Server for Windows NT, with new support for Windows 2000.

It has also announced Version 6.0 of its Tivoli NetView network management tool for centralizing the management of network hardware devices and servers.

For further information, contact your local IBM representative, or visit the Web site at [www-4.ibm.com/software/network/commsserver](http://www-4.ibm.com/software/network/commsserver).

\* \* \*

Tivoli has announced its Comprehensive Network Address Translator (CNAT), promising integration of multiple networks with different IP addresses, and increasing the reach of TCP/IP management tools like NetView.

For further information, contact:  
Tivoli Systems, 9442 Capital of Texas Highway North, Arboretum, Austin, TX 78759, USA.  
Tel: (512) 436 8000.  
URL: [http://www.tivoli.com/products/index/comp\\_net\\_add\\_trans/index.html](http://www.tivoli.com/products/index/comp_net_add_trans/index.html).

\* \* \*

MacKinney Systems has announced JES Queue Client for Printers (JQP), which prints reports from the JES output queue to network attached printers defined to a VTAM or TCP/IP LPD Daemon. It has also announced VTAM Virtual Printer (VVP), which routes output created by on-line

applications to the system spool.  
For further information, contact:  
MacKinney Systems, 2740 S Glenstone, Suite 103, Springfield, Missouri, 65804-3737, USA.  
Tel: (417) 882 8012.  
URL: <http://www.mackinney.com>.

\* \* \*

Computer Associates has begun shipping Version 2.0 of its NetworkIT network management software, now using the company's Neugents neural network tools to help predict network problems before they affect operations.

For further information, contact:  
Computer Associates International Inc, One Computer Associates Plaza, Islandia, NY 11788-7000, USA.  
Tel: (516) 342 5734.  
URL: <http://www.cai.com/products/networkit.htm>.

\* \* \*

3Com has announced the EtherLink Server 10/100 Mbps PCI NIC with 3XP processor, which off-loads key TCP/IP networking tasks, such as Internet Protocol Security, TCP/IP checksum, and TCP segmentation, from the host system to the NIC's on-board processor and encryption chip.

For further information, contact:  
5400 Bayfront Plaza, Santa Clara, CA 95052, USA.  
Tel: (408) 326 5000.  
URL: <http://www.3com.com/products/dsheets/400517.html>.



**xephon**