40 TCP/SNA update

December 2000

In this issue

TCP/SNA Update

Published by

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

North American office

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

Subscriptions and back-issues

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

Editorial panel

Editor

Fiona Hewitt

Disclaimer

Contributions

Articles published in *TCP/SNA Update* are paid for at the rate of £170 ($260) per 1000 words for original material. To find out more about contributing an article, please contact us and we will send you a copy of our *Notes for Contributors,* or you can download a copy from www.xephon.com/contnote.html.

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.

Printed in England.

Session management exit routine

There are many networks across America that legitimately connect into my (ie the TITAN's) network, and those networks in turn have many other networks that can legitimately connect into them. The problem is that a few networks that are connected to mine incorrectly forward requests for access to on-line applications to my network, after determining that nothing within their network satisfies the request. These requests for access to on-line applications, non-existent in my (ie the TITAN's) network as well, are propagated throughout my LPARs until they are eventually rejected by the LPAR that initially accepted the request. Each LPAR in turn issues appropriate error messages as it rejects each request. Since this is clearly a complete waste of my computer resources, I wanted to stop them from entering my system or, at least, prevent them from being forwarded on to my other LPARs.

My first course of action was to request help from one of the shops that was allowing these requests to be forwarded into my network in the first place. The first suggestion I received was that I should suppress the messages at my end of the connection. I rejected this on the grounds that automatically-generated requests could be continually rejected by my network and I would never be aware of them.

It's a relatively simple task to eliminate the unwelcome VTAM error messages from my system consoles – VTAM has a start-option to do this. Or rather, it has four start-options that are similar in nature. These options can be changed dynamically by using an F NET,VTAMOPTS command. The operands, along with their purpose, are as follows:

- SIRFMSG=NONE – suppress IST663/664/889 messages.
- ASIRfMSG=NONE – suppress IST890/896 messages.
- ESIRFMSG=NONE – suppress IST892/893 messages.
- FSIRFMSG=NONE – suppress IST894/895 messages.

In the end, the systems programmer at the shop I was talking to created a table containing all of the names of applications that his customers had attempted to access in my network, and stopped forwarding requests for them to me. However, this was not a completely satisfactory solution since misspellings were still forwarded. I therefore decided to create an exit that would block all requests not destined for my network at their initial entry point into my system, hence ISTEXCAA (below).

ISTEXCAA is a session management exit routine that can be used to control, manage, and authorize the establishment of sessions between an LU and an application, and any of the other permutations of connections that are possible. It is documented in *VTAM Customization* (LY43-0068). ISTEXCAA can be activated via a VTAM modify command:

```
F NET,EXIT,OPT=ACT,ID=ISTEXCAA
```

where NET is the name of the procedure that was used to start VTAM.

If this exit is modified and you choose to use the new code, you need to issue the following command:

```
F NET,EXIT,OPT=REPL,ID=ISTEXCAA
```

The exit is halted by entering the following command:

```
F NET,EXIT,OPT=INACT,ID=ISTEXCAA
```

Issue 'D NET,EXIT' in order to obtain the status of all VTAM exits.

Many function codes can be processed in this exit; I chose to process four:

- The begin function
- The initial authorization function
- The secondary authorization function
- The adjacent SSCP selection function.

Although I believe that it was the latter that made this exit work, I left the other code in since it was already in place – the processing uses the same code anyway.

The begin function is used to set the processing options that will be performed by this exit. Interestingly, the begin function is not entered unless the ACT OPTION is used to activate ISTEXCAA. However, another function code does get invoked so that similar processing can still be done. I opted to first ‘INACT’ and then ‘ACT’ ISTEXCAA in order to make changes to my code.

For the purpose of tracking the function codes processed by ISTEXCAA, I issued a WTO that contained each function processed. After completing my development of ISTEXCAA, I commented those instructions, but left them in place in case anyone else finds a future requirement for them.

Whenever a request is processed by ISTEXCAA that is either destined for processing by my network or originating from it, I permit the request to flow on. However, since dynamic CDRSCs and ADJSSCPs are allowed in my shop, VTAM performs what may be termed a ‘speculative’ search for an application request from one of my SNI partners and, at the time ISTEXCAA receives control, there is no proper DLU NETID in the Resource Identifier Control Vector(RIC) and a foreign NETID in the SLUs RIC. At that point in my processing, I locate the DLU name and then search VTAM’s control blocks in order to locate a matching name in one of the application program name segments or CDRSC segments. I skip my definition for TELEVIEW, a Computer Associates software product, since the definitions located there are not actually ‘application programs’. I bypass TSO definitions for the same reason. This also reduces search time. Note that this process duplicates work that is performed by VTAM. Also, subsequent releases of VTAM may change the format of Resource Definition Table Header Entries(RDTE) so that my code would no longer function properly, if at all. For me, that would be a temporary set-back – others can only answer for themselves.

Back to what I was saying: if my attempt to locate the application name in VTAM’s control blocks proves successful, the request is allowed to continue; otherwise, I issue an error message, IST000I, that contains the name of the SSCP, if available, of the originating and destination points, and the network identifier of each of those points, and then fail the request. Each network identifier is always eight bytes long. The direction of the request is indicated by either a greater than, ‘>’, or less than, ‘<’ sign.

Below is a sample of that message and the regular VTAM messages that would accompany it:

```
SSCP     NET-ID     SSCP     NET-ID
        -------- --------   -------- -------

IST000I OUTHOSTU OUTNETU  < BEVCPUA  PORTNET   REQUEST DENIED
IST663I CDINIT REQUEST FROM Q01 FAILED, SENSE=080A0000
IST664I REAL  OLU=PORTNET.BEVY602     REAL  DLU=OUTNETU.OUTTIP
```

ISTEXCAA must be linked with the RENT option. I stowed this exit in a dataset named SYS1.VTAMLIB. Afterwards, I had to remember to refresh that member since SYS1.VTAMLIB is in the LNKLST.

In order to prevent the propagation of erroneous requests from leaving my network, flowing into the networks of others and annoying them, the following entries have been made to my SYS1.VTAMLST dataset. Similar entries could be made by anyone who wants to restrict the search for illegitimate SNA requests to his network.

- ADJTITAN was included as the first entry in member ATCCONxx, where xx is the operand of the LIST keyword used to activate my network, ie SNET,,,(LIST=xx). Each ADJCDRM must also be defined in SYS1.VTAMLST, and the name of the member in SYS1.VTAMLST that contains it must have been specified in member ATCCONxx. Sample entries are shown below.
- Member ADJTITAN contains the names of SSCPs within my province, followed by a dummy entry.
  - ADJTITAN VBUILD TYPE=ADJSSCP
  - CDRMSSP1 ADJCDRM
  - CDRMSSP2 ADJCDRM
  - CDRMSSP3 ADJCDRM
  - CDRMSSP4 ADJCDRM
  - CDRMDUM CDRM

Member CDRMT1 contains:

```
  CDRMSSP1  VBUILD  TYPE=CDRM
                      NETWORK  NETID=NETITAN
```

```
  CDRMSSP1  CDRM  SUBAREA=4,ISTATUS=ACTIVE,CDRSC=OPT,CDRDYN=YES
```

Member ATCCONxx contains:

```
ADJTITAN,                                ===============>   First entry!
<=================
...
CDRMT1,
CDRMT2,
...
```

This exit currently runs under VTAM 4.2 and OS/390 2.5.

ISTEXCAA

```
TITLE 'ISTEXCAA - SESSION MANAGEMENT EXIT ROUTINE'
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*   THE PURPOSE OF THIS ROUTINE IS TO PREVENT ENTRY OF REQUESTS       *
*   DESTINED FOR OTHER NETWORKS FROM ENTERING THE TITAN'S NETWORK.    *
*   PROCESSING IS PERFORMED DURING THE BEGIN, INITIAL AND SECONDARY   *
*   AUTHORIZATION, AND ADJACENT SSCP SELECTION FUNCTIONS, FUNCTION    *
*   CODES, X'FE', X'ØØ', X'Ø1', AND X'Ø6', RESPECTIVELY.              *
*                                                                     *
*   CONTENTS OF REGISTERS AT ENTRY TO ISTEXCAA:                       *
*   R1    - ADDRESS OF A PARAMETER LIST, VARIABLE IN LENGTH AND       *
*           DEPENDENT UPON THE NATURE OF THE EXIT, THAT CONTAINS      *
*           ADDRESSES OF OTHER PARAMETERS.  THE FIRST THREE ADDRESSES *
*           IN A PARAMETER LIST ARE APPLICABLE TO ALL FUNCTIONS.      *
*           OFFSET  DESCRIPTION                                       *
*           Ø       ENVIRONMENT VECTORS ADDRESS                       *
*           4       FUNCTION CODE AND ADDRESS OF SESSION INFORMATION  *
*           8       ADDRESS OF USER DATA FIELD                        *
*           C       DON'T USE IT!                                     *
*           1Ø      DON'T USE IT!                                     *
*           14      DON'T USE IT!                                     *
*           18      DON'T USE IT!                                     *
*                                                                     *
*         FOR THE BEGIN FUNCTION:                                     *
*           C       ADDRESS OF OPTIONS FOR THIS EXIT                  *
*                                                                     *
*         FOR THE INITIAL AND SECONDARY AUTHORIZATION AND             *
*         AND ADJACENT SSCP SELECTON FUNCTIONS:                       *
*           C       ADDRESS OF CONTROL VECTOR FOR PLU RESOURCE ID     *
*           1Ø      ADDRESS OF CONTROL VECTOR FOR SLU RESOURCE ID     *
*           14      ADDRESS OF SESSION IDENTIFIER                     *
*           18      DON'T USE IT!                                     *
*                                                                     *
*           ...     .............................................     *
```

```
*          ...     ........................................     *
*          ...     ........................................     *
*                                                               *
*                                                               *
*   R13   - ADDRESS OF A STANDARD SAVE AREA                     *
*   R14   - ADDRESS OF A RETURN POINT                           *
*   R15   - ADDRESS OF ISTEXCAA'S ENTRY POINT                   *
         EJECT
*   CONTENTS OF REGISTERS UPON DEPARTURE FROM ISTEXCAA FOR      *
*   BEGIN FUNCTION:                                              *
*   R15   - RETURN CODE:                                        *
*           Ø - PROCESSING WAS SUCCESSFULLY PERFORMED           *
*                                                               *
*   INITIAL AUTHORIZATION FUNCTION:                             *
*   R15   - RETURN CODE:                                        *
*           Ø - CONNECTION REQUEST IS PERMITTED; NO SECONDARY EXIT   *
*           4 - CONNECTION REQUEST IS PERMITTED; INVOKE SECONDARY EXIT*
*           8 - CONNECTION REQUEST IS DENIED; DON'T REROUTE     *
*                                                               *
*   SECONDARY AUTHORIZATION FUNCTION:                           *
*   R15   - RETURN CODE:                                        *
*           Ø - CONNECTION REQUEST IS PERMITTED                 *
*           8 - CONNECTION REQUEST IS DENIED                    *
*                                                               *
*   ALL OTHER REGISTERS ARE RESTORED TO THEIR ORIGINAL CONTENTS.     *
*                                                               *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         EJECT
ISTEXCAA CSECT
         SPACE
ISTEXCAA AMODE 31
ISTEXCAA RMODE 24
         SPACE
         PRINT NOGEN
         SPACE
         USING ISTEXCAA,R12        ESTABLISH ISTEXCAA ADDRESSABILITY
         USING PSA,RØ              ESTABLISH PSA ADDRESSABILITY
         SPACE
         BAKR  R14,RØ              PRESERVE ENVIRONMENT AT ENTRY
         LR    R12,R15             PRIME BASE REGISTER
         LR    R11,R1              REPEAT GPR 1
         USING PPGSAFPL,R11        ESTABLISH PPGSAFPL ADDRESSABILITY
         SPACE
         L     R2,PPGSAFC          FETCH POINTER TO FUNCTION CODE
         SPACE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                               *
*        FORMAT AND DISPLAY CURRENT FUNCTION CODE.              *
*                                                               *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
         SPACE
*        GETMAIN R,LV=5Ø
*        LR    R5,R1
*        MVC   Ø(PPGWTOL,R5),PPGWTO
*        UNPK  13(3,R5),Ø(2,R2)    FUNCTION CODE
*        TR    13(2,R5),PPGTRANS-24Ø
*        MVI   15(R5),C' '
*        WTO   MF=(E,(5))
*        FREEMAIN R,LV=5Ø,A=(5)
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        ASCERTAIN IF PROCESSING IS TO BE PERFORMED FOR THE           *
*        CURRENT FUNCTION CODE.                                       *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
         SR    R8,R8               CLEAR A WORK REGISTER
         IC    R8,Ø(R2)            FETCH FUNCTION CODE
         SPACE
         CLI   Ø(R2),6             TEST IF ADJACENT SSCP SELECTION FUNC
         BE    PPGPROCS            IF SO, PROCESS REQUEST
         SPACE
         CLI   Ø(R2),2             TEST IF SECONDARY AUTHORIZATION
         BL    PPGPROCS            IF SO, PROCESS REQUEST
         SPACE
         CLI   Ø(R2),X'FE'         TEST IF ENTRY FOR INITIALIZATION
         BNE   PPGBYE              IF NOT, DEPART FROM THIS ROUTINE
         SPACE
         L     R2,PPGSEOA          FETCH POINTER TO EXIT OPTIONS
         OI    Ø(R2),PPGOSA        SET OPTION TO PROCESS SECONDARY AUTH
         OI    Ø(R2),PPGOPA        SET OPTION TO PROCESS PRINMARY AUTH
         OI    Ø(R2),PPGOCA        SET OPTION TO PROCESS ADJ SSCP SEL
         SPACE 2
PPGBYE   C     R8,PPGFØ            TEST IF INITIAL AUTHORIZATION CODE
         BNE   PPGALLOW            BRANCH IF NOT
         LA    R15,4               REQUIRE SECONDARY AUTHORIZATION CODE
         PR    R14                 BACK TO DUST
         SPACE
PPGALLOW SR    R15,R15             ALLOW REQUEST TO CONTINUE
         PR    R14                 BACK TO DUST
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        LOCATE THE 'VECTOR' CONTAINING THE NETWORK IDENTIFIER        *
*        OF THIS ROUTINE'S OPERATING ENVIRONMENT.  ITS IDENTIFIER     *
*        HAS A VALUE OF SIX.                                          *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
```

```
PPGPROCS ICM   R2,15,PPGSAEV       POINT TO VECTORS' HEADER
         BE    PPGBYE              EXIT IF NONEXISTENT
         SPACE
         CLC   PPGTWO,Ø(R2)        TEST FOR THE PRESENCE OF ANY VECTORS
         BE    PPGBYE              EXIT IF NONE ARE PRESENT
         SPACE
         LH    R3,Ø(R2)            TOTAL LENGTH OF PARAMETER LIST
         LA    R1,2                SIZE OF VECTOR HEADER
         SPACE
PPGVLOOP LA    R2,Ø(R1,R2)         POINT TO NEXT VECTOR IN PARM LIST
         CLI   1(R2),6             TEST IF VECTOR CONTAINS NETWORK ID
         BE    PPGHAVID            BRANCH IF SO
         SPACE
         SR    R3,R1               REDUCE TOTAL LNGTH BY LENGTH OF PARM
         BNP   PPGBYE              BRANCH IF ID OF NETWORK NOT AVAILABL
         IC    R1,Ø(R2)            FETCH LENGTH OF THIS VECTOR ENTRY
         B     PPGVLOOP            LOOP POWER!
         SPACE
PPGHAVID ICM   R1,1,Ø(R2)          FETCH LENGTH OF NETWORK IDENTIFIER
         BE    PPGBYE              EXIT IF HOST ID IS NOT AVAILABLE
         SH    R1,PPGTWO           REDUCE FOR EXECUTE INSTRUCTION
         BCTR  R1,RØ               GREAT REDUCTION PLAN
         L     R7,PSAATCVT         ADDRESS OF VTAM'S VECTOR TABLE
         SPACE
         EX    R1,PPGCLCID         ENSURE NETWORK IDENTIFIERS MATCH
         BNE   PPGBYE              EXIT IF NOT
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        ASCERTAIN IF THE PLU RESIDES IN THIS NETWORK.  IF IT         *
*        DOES, PERMIT THE REQUEST.  IF IT DOESN'T, THEN CHECK THE     *
*        DESTINATION OF REQUEST.  IF THE REQUEST IS INTENDED FOR      *
*        THIS NETWORK, PERMIT IT, OTHERWISE, FAIL IT.                 *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
         L     R3,PPGSAPLU         GET ADR OF PLU'S RESC ID CNTL VECTOR
         USING PPGRICL,R3          ESTABLISH PPGRICL ADDRESSABILITY
         SPACE
         CLI   PPGRKEY,X'19'       ENSURE THAT A VECTOR KEY IS PRESENT
         BNE   PPGBYE              EXIT IF UNKNOWN VECTOR KEY
         SPACE
         SR    R14,R14             CLEAR A VOLATILE REGISTER
         ICM   R14,1,PPGRVL        FETCH LENGTH OF VECTOR(-2)
         BE    PPGBYE              EXIT IF NONEXISTENT
         SH    R14,PPGTWO          REDUCE BY TWO(SKIP SSCP REROUT KONT)
         BE    PPGBYE              EXIT IF END OF VECTOR LIST
         SPACE
         LA    R1Ø,PPGRNET         POINT TO LU'S NETWORK IDENTIFIER
         CLI   PPGRLCPN,Ø          TEST IF NAME OF SSCP IS PRESENT
```



```
         BNE   PPGHAVCP            BRANCH IF 'TIS
         SH    R3,PPGEIGHT         ADJUST BASE REGISTER FOR ABSENT SSCP
         SPACE
         LA    R1Ø,PPGRNET         ADJ PTR TO LU'S NETWORK IDENTIFIER
         CLI   PPGRLNET,Ø          TEST IF NAME OF NETWORK IS PRESENT
         BNE   PPGHAVCP            BRANCH IF 'TIS
         SH    R3,PPGEIGHT         ADJUST BASE REGISTER FOR ABSENT ID
         SPACE
         LA    R1Ø,PPGRLUNM        POINT TO LU'S NETWORK IDENTIFIER
         CLI   PPGRLLUN,Ø          TEST IF NAME OF NETWORK IS PRESENT
         BNE   PPGHAVCP            BRANCH IF 'TIS
         SH    R3,PPGEIGHT         ADJUST BASE REGISTER FOR ABSENT ID
         SPACE
         LA    R1Ø,PPGRNAL         POINT TO LU'S NETWORK IDENTIFIER
         CLI   PPGRLNAL,Ø          TEST IF NAME OF NETWORK IS PRESENT
         BE    PPGBYE              EXIT IF NETWORK IDENTIFIER IS ABSENT
         SPACE
PPGHAVCP CLC   Ø(8,R1Ø),ATCNETID(R7) TEST IF ORIGIN IS MY NETWORK
         BE    PPGBYE              EXIT IF PLU IS WITHIN THIS NETWORK
         SPACE
         LR    R4,R1Ø              POINT TO PLU'S NETWORK IDENTIFIER
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        ASCERTAIN IF THE SLU RESIDES IN THIS NETWORK.  IF IT         *
*        DOES, PERMIT THE REQUEST, ELSE ISSUE AN ERROR MESSAGE        *
*        AND FAIL THE REQUEST.                                        *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
         L     R3,PPGSASLU         GET ADR OF SLU'S RESC ID CNTL VECTOR
         SPACE
         CLI   PPGRKEY,X'19'       ENSURE THAT A VECTOR KEY IS PRESENT
         BNE   PPGBYE              EXIT IF UNKNOWN VECTOR KEY
         SPACE
         SR    R14,R14             CLEAR A VOLATILE REGISTER
         ICM   R14,1,PPGRVL        FETCH LENGTH OF VECTOR(-2)
         BE    PPGBYE              EXIT IF NONEXISTENT
         SH    R14,PPGTWO          REDUCE BY TWO(SKIP SSCP REROUT KONT)
         BE    PPGBYE              EXIT IF END OF VECTOR LIST
         SPACE
         SPACE
         LA    R1Ø,PPGRNET         POINT TO LU'S NETWORK IDENTIFIER
         CLI   PPGRLCPN,Ø          TEST IF NAME OF SSCP IS PRESENT
         BNE   PPGGOTCP            BRANCH IF 'TIS
         SH    R3,PPGEIGHT         ADJUST BASE REGISTER FOR ABSENT SSCP
         SPACE
         LA    R1Ø,PPGRNET         ADJ PTR TO LU'S NETWORK IDENTIFIER
         CLI   PPGRLNET,Ø          TEST IF NAME OF NETWORK IS PRESENT
         BNE   PPGGOTCP            BRANCH IF 'TIS
```

```
         SH    R3,PPGEIGHT         ADJUST BASE REGISTER FOR ABSENT ID
         SPACE
         LA    R1Ø,PPGRLUNM        POINT TO LU'S NETWORK IDENTIFIER
         CLI   PPGRLLUN,Ø          TEST IF NAME OF NETWORK IS PRESENT
         BNE   PPGGOTCP            BRANCH IF 'TIS
         SH    R3,PPGEIGHT         ADJUST BASE REGISTER FOR ABSENT ID
         SPACE
         LA    R1Ø,PPGRNAL         POINT TO LU'S NETWORK IDENTIFIER
         CLI   PPGRLNAL,Ø          TEST IF NAME OF NETWORK IS PRESENT
         BE    PPGBYE              EXIT IF DESTINATION NETWORK UNAVAIL
         SPACE
PPGGOTCP CLC   Ø(8,R1Ø),ATCNETID(R7) ASSUME NAME IS ALWAYS EIGHT BYTES
         BE    PPGBYE              EXIT IF SLU IS WITHIN THIS NETWORK
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        LOCATE NAME OF DESTINATION LU                                *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
         L     R3,PPGSAPLU         GET ADR OF PLU'S RESC ID CNTL VECTOR
         TM    PPGRUI,X'4Ø'        TEST IF THIS IS THE TARGET RESOURCE
         BO    PPGTARGT            BRANCH IF NOT
         SPACE
         L     R3,PPGSASLU         GET ADR OF SLU'S RESC ID CNTL VECTOR
         TM    PPGRUI,X'4Ø'        TEST IF THIS IS THE TARGET RESOURCE
         BNO   PPGBYE              BRANCH IF NOT, SHOULD NOT OCCUR
         SPACE
PPGTARGT DS    ØH                  START OF COMMON CODE
         LA    R2,PPGRLCPN         POINT TO START OF ENTRIES
         SR    R1,R1               ZERO VOLATILE REGISTER
         SPACE
         IC    R1,Ø(R2)            LENGTH OF FIRST OF FOUR ENTRIES
         LA    R2,1(R1,R2)         POINT TO SECOND ENTRY
         SPACE
         IC    R1,Ø(R2)            LENGTH OF SECOND OF FOUR ENTRIES
         LA    R2,1(R1,R2)         POINT TO THIRD ENTRY
         SPACE
         IC    R1,Ø(R2)            LENGTH OF THIRD OF FOUR ENTRIES
         LA    R2,1(R1,R2)         POINT TO FOURTH ENTRY
         SPACE
         IC    R1,Ø(R2)            LENGTH OF FOURTH OF FOUR ENTRIES
         LA    R2,1(R1,R2)         POINT TO NAME
         SPACE
         ICM   R1,1,Ø(R2)          LENGTH OF NAME
         BE    PPHMVIFF            IF NONE, KILL REQUEST
         SPACE
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
```



```
*        ATTEMPT TO LOCATE AN RDT WITH AN ENTRY THAT MATCHES THE      *
*        NAME OF THE DLU IN THE RESOURCE IDENTIFIER CONTROL           *
*        VECTOR(RIC)                                                  *
*                                                                     *
*        PSA + X'4Ø8'(PSAATCVT) ====> ATCVT                           *
*        ATCVT + X'44Ø'(ATCCONFT) ====> CONFT                         *
*        CONFT + X'94'(CONVTHAA) ====> FIRST RDT ON A CHAIN OF RDT'S  *
*        RDT + X'7Ø'(RDTFORW) ====> NEXT RDT(RRN, RSW, RLS)           *
*        RDT + X'24'(RPRELEN) ====> OFFSET FROM BEGINNING OF THIS     *
*                                   RDT TO A CHAIN OF SUBORDINATE     *
*                                   ENTRIES.  EACH SUBORDINATE IS     *
*                                   CHAINED VIA AN OFFSET, FOUND AT   *
*                                   A DISPLACEMENT OF X'24' IN THE    *
*                                   CURRENT ENTRY, FROM THE CURRENT   *
*                                   ENTRY.                            *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE 1
         L     R7,PSAATCVT         ADDRESS OF VTAM'S VECTOR TABLE
         L     R9,ATCCONFT(R7)     ADDRESS OF VTAM CONFIGURATION TABLE
         SPACE 1
         L     R5,CONVTHAA(R9)     ADDRESS OF HOST CDRM DUMMY RDTE
         SPACE
PPHFORM  ICM   R15,15,RDTFORW(R5)  TEST IF THIS IS A DUMMY ENTRY
         BE    PPHMVIFF            BRANCH IF SO
         SPACE
         TM    RPRBITAN,8          TEST IF THIS IS THE LAST RDT ENTRY
         BO    PPGGO               BRANCH IS SO
         SPACE
         CLI   RPRENTRY(R5),RPRENTAH TEST IF APPLICATION HEADER ENTRY
         BNE   PPGAPLXD            BRANCH IF NOT
         CLC   RPRNAME(4,R5),=CL4'TELE' TEST IF TELEVIEW RESOURCE
         BNE   PPGAPLHD            BRANCH IF NOT
         B     PPGGO               BRANCH ANYWAY
         SPACE
PPGAPLXD CLI   RPRENTRY(R5),RPRENTRS TEST IF X-DOMAIN RESOURCE HEADER
         BNE   PPGGO               BRANCH IF NOT
         SPACE
         CLC   RPRNAME(3,R5),=CL3'IST' TEST FOR 'TERMINALS'
         BE    PPGGO
         SPACE
PPGAPLHD DS    ØH                  TARGET FOR BRANCH INSTRUCTION
         SPACE
         TM    RPRBITAN(R5),8      TEST IF THIS IS THE LAST RDT ENTRY
         BO    PPGGO               BRANCH IF SO
         SPACE 1
         ICM   R9,15,RPRELEN(R5)   FETCH OFFSET TO RDT
         BE    PPGGO               BRANCH IF AT END
         AR    R9,R5               POINT TO ENTRY
         EJECT
```



```
PPGLOOP  DS    ØH                   BRANCH ENTRY - ALSO
         CLI   RPRENTRY(R9),RPRENTRC TEST IF X-DOMAIN RESOURCE
         BE    PPHMATCH             BRANCH IF SO
         CLI   RPRENTRY(R9),RPRENTAP TEST IF APPLICATION
         BNE   PPGGO                BRANCH IF NOT
         SPACE
PPHMATCH CLC   1(8,R2),RPRNAME(R9) TEST IF NAME OF RDT MATCHES RIC LU
         BE    PPGBYE
         SPACE 1
         TM    RPRBITAN(R9),8       TEST IF THIS IS THE LAST RDT ENTRY
         BO    PPGGO                BRANCH IF SO
         SPACE 2
PPHNXRCC A     R9,RPRELEN(,R9)      OFFSET TO NEXT RCC
         B     PPGLOOP              PROCESS IT
         SPACE 1
PPGGO    DS    ØH                   ENTRY POINT
         ICM   R5,15,RDTFORW(R5)    ADDRESS OF NEXT RDTE
         BNE   PPHFORM              BRANCH IF NOT AT END OF RDTE'S
PPHMVIFF DS    ØH                   HANDLE IN NAME ONLY
         SPACE 1
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                   *
*        OBTAIN TEMPORARY VIRTUAL STORAGE IN WHICH TO STOW, FORMAT,   *
*        AND ISSUE AN ERROR MESSAGE.                                  *
*                                                                   *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
         GETMAIN R,LV=5Ø
         LR    R5,R1
         USING PPGWTOR,R5           ESTABLISH PPGWTOR ADDRESSABILITY
         MVC   Ø(PPGWTOLF,R5),PPGWTOF MESSAGE PATTERN TO OUTPUT AREA
         MVC   PPGPNET,Ø(R4) PLU'S NETWORK IDENTIFIER
         MVC   PPGSNET,Ø(R1Ø) SLU'S NETWORK IDENTIFIER
         SPACE
         L     R3,PPGSAPLU          GET ADR OF PLU'S RESC ID CNTL VECTOR
         TM    PPGRUI,X'4Ø'         TEST IF THIS IS THE TARGET RESOURCE
         BNO   PPGCKPS              BRANCH IF NOT
         MVI   PPGHYP,C'<'          SHOW PLU IS TARGET
         SPACE
PPGCKPS  CLI   PPGRLCPN,Ø           TEST IF NAME OF SSCP IS PRESENT
         BE    PPGDOTO              BRANCH IF NOT
         MVC   PPGPSSCP,PPGRCPNM    TRANSFER NAME INTO MESSAGE
         SPACE
PPGDOTO  L     R3,PPGSASLU          GET ADR OF SLU'S RESC ID CNTL VECTOR
         TM    PPGRUI,X'4Ø'         TEST IF THIS IS THE TARGET RESOURCE
         BNO   PPGCKSS              BRANCH IF NOT
         MVI   PPGHYP,C'>'          SHOW SLU IS TARGET
         SPACE
PPGCKSS  CLI   PPGRLCPN,Ø           TEST IF NAME OF SSCP IS PRESENT
```



```
         BE    PPGISSUE            BRANCH IF NOT
         MVC   PPGSSSCP,PPGRCPNM   TRANSFER NAME INTO MESSAGE
         SPACE
PPGISSUE WTO   MF=(E,(5))
         FREEMAIN R,LV=5Ø,A=(5)
         SPACE
         C     R8,PPGFØ            TEST IF INITIAL AUTHORIZATION CODE
         BNE   PPGR8               BRANCH IF NOT
         SPACE
         LA    R15,4               REQUIRE SECONDARY AUTHORIZATION CODE
         PR    R14                 BACK TO DUST
         SPACE
PPGR8    LA    R15,8               SESSION REQUEST FLUNKS
         PR    R14                 BACK TO DUST
         TITLE 'ESA CONTROL BLOCKS'
         SPACE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        CONSTANTS AND OTHER SUCH NONSENSE                            *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE 1
RPRNAME  EQU   Ø                   NAME OF RDTE ET AL
RPRENTAH EQU   2                   ENTRY IS AN APPLICATION HEADER
RPRENTRS EQU   7                   ENTRY IS A CROSS-DOMAIN RSC HEADER
RPRENTRY EQU   X'ØD'               ENTRY TYPE
RPRHDTYP EQU   X'ØE'               HEADER TYPE
RPRELEN  EQU   X'24'               LENGTH OF CURRENT ENTRY
RPRCURST EQU   X'3C'               CURRENT-STATE BYTES(SEE FSM)
RPRDESST EQU   X'3E'               DESIRED-STATE BYTES(SEE FSM)
RPRBITAN EQU   X'41'               FLAG BITS
RPRENTAP EQU   X'55'               ENTRY IS AN APPLICATION
RPRENTRC EQU   X'83'               ENTRY IS A CROSS-DOMAIN RESOURCE
         SPACE
RDTFORW  EQU   X'7Ø'               POINTER TO NEXT SEGMENT
         SPACE
RPXPUPTR EQU   X'68'               POINTER TO ASSOCIATED PU RDTE
         SPACE
CONVTHAA EQU   X'94'               POINTER TO VTAM RDT HEADER AREA
         SPACE
ATCCONFT EQU   X'44Ø'              POINTER TO VTAM CONFIG TABLE
         SPACE 3
PPGFØ    DC    F'Ø'
PPGONE   DC    H'1'
PPGTWO   DC    H'2'
PPGEIGHT DC    H'8'
         SPACE
PPGCLCID CLC   2(*-*,R2),ATCNETID(R7) ****** EXECUTE ONLY ******
         SPACE
ATCNETID EQU   X'82Ø'              DISPLACEMENT TO MY HOST'S IDENTIFIER
```

```
         SPACE
PPGTRANS DC    C'Ø123456789ABCDEF'
         SPACE 2
PPGWTO   WTO   'FUNCTION                   ',MF=L
PPGWTOL  EQU   *-PPGWTO
         SPACE 2
         DS    ØF
PPGWTOF  WTO   'ISTØØØI NO SSCP  LRLSEBGB - NO SSCP  LSNBCPPG  REQUEST C
               DENIED',MF=L
PPGWTOLF EQU   *-PPGWTOF
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        DSECT FOR A PARAMETER LIST OF THE PRIMARY AND SECONDARY      *
*        AUTHORIZATION FUNCTION.                                      *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
PPGSAFPL DSECT
PPGSAEV  DS    A                   ADR OF ENVIRONMENTAL VECTORS
PPGSAFC  DS    A                   ADR OF FUNCTION CODE & SESSION INFO
PPGSAUSR DS    A                   ADR OF USER'S DATA FIELD
         SPACE
PPGSEOA  EQU   *                   ADR OF EXIT'S OPTIONS
PPGOPA   EQU   128                 TAKE PRIMARY AUTHORIZATION EXIT
PPGOSA   EQU   64                  TAKE SECONDARY AUTHORIZATION EXIT
PPGOCA   EQU   2                   TAKE ADJACENT SSCP SELECTION EXIT
         SPACE
PPGSAPLU DS    A                   ADR OF CNTL VECTOR FOR PLU RESRC ID
PPGSASLU DS    A                   ADR OF CNTL VECTOR FOR SLU RESRC ID
PPGSASID DS    A                   ADR OF SESSION IDENTIFIER
PPGSANA1 DS    A                   DON'T USE THIS FIELD
PPGSAOLU DS    A                   ADR OF INFO VECTOR FOR OLU GATEWAY
PPGSADLU DS    A                   ADR OF INFO VECTOR FOR DLU GATEWAY
PPGSAOLS DS    A                   ADR OF SSCP VECTOR FOR ADJACENT OLU
PPGSADLS DS    A                   ADR OF SSCP VECTOR FOR ADJACENT DLU
PPGSANA2 DS    A                   DON'T USE THIS FIELD
PPGSANA3 DS    A                   DON'T USE THIS FIELD
PPGSANA4 DS    A                   DON'T USE THIS FIELD
PPGSAILU DS    A                   ADR OF CNTL VECTOR FOR ILU RESRC ID
PPGSAVXS DS    A                   ADR OF PARM LIST FOR VTAM EXIT SRVCS
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*   DSECT FOR A RESOURCE IDENTIFICATION CONTROL VECTOR.                *
*   LENGTHS OF NAMES ARE ASSUMED TO ALWAYS BE EIGHT - NEVER VERIFIED. *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
PPGRICL  DSECT
```

```
PPGRKEY  DS    X                    KEY OF VECTOR           - 19
PPGRVL   DS    X                    LENGTH OF VECTOR
PPGRRCNT DS    X                    REROUTE COUNT           - Ø
PPGRUI   DS    X                    USAGE INDICATOR         - 4Ø
         SPACE
PPGRLCPN DS    X                    LENGTH OF NAME OF SSCP  - 8
PPGRCPNM DS    CL8                  NAME OF SSCP
         SPACE
PPGRLNET DS    X                    LENGTH OF NAME OF NETID - 8
PPGRNET  DS    CL8                  NAME OF REAL NETID
         SPACE
PPGRLLUN DS    X                    LENGTH OF NAME OF LU    - 8
PPGRLUNM DS    CL8                  NAME OF LU
         SPACE
PPGRLNAL DS    X                    LENGTH OF NETID'S ALIAS - 8
PPGRNAL  DS    CL8                  NAME OF NETID'S ALIAS
         SPACE
PPGRLALU DS    X                    LENGTH OF ALIAS OF LU   - 8
PPGRALUN DS    CL8                  LU'S ALIAS NAME
         SPACE
PPGRCVK  DS    X                    KEY OF CONTROL VECTOR   - 1A
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        DSECT FOR A REJECTION MESSAGE                                *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE
PPGWTOR  DSECT
         DS    F
         DS    CL7
         DS    CL1
PPGPSSCP DS    CL8
         DS    CL1
PPGPNET  DS    CL8
         DS    CL1
PPGHYP   DS    CL1
         DS    CL1
PPGSSSCP DS    CL8
         DS    CL1
PPGSNET  DS    CL8
         TITLE 'OS CONTROL BLOCKS USED BY ISTEXCAA'
         YREGS
         SPACE 2
         IHAPSA
         SPACE
         END
```

Systems Programmer (USA) © Reserved 2000

3270 datastream basics

Even though you may not find an SNA or Bi-SynC (BSC) connection, or indeed any 3270 hardware, on today's desktops, on-line mainframe applications are still based on the 3270 datastream. No one could have imagined, when it was introduced in the early 1970s, that it was destined for such longevity. This article takes a close look at 3270 datastream basics, and analyses why it has played such an important role.

BLOCK MODE

The major disadvantages of the original 3270 technology were that:

- The attribute byte that preceded each field on the screen wasted screen real estate by occupying a blank spot on the screen.
- Screen size was fixed, although there were several sizes to choose from.
- It ran only in block mode.

'Block mode' refers to the fact that transmission from the terminal to the host occurs only when the user strikes specific keys – typically the *Enter* key, although *Program Function* (PF) and *Program Attention* (PA) keys are other common examples. For its time, the 3270 terminal system had a lot of intelligence, and was quite sophisticated in how it would respond to keystrokes from the user. Although most of the work was done by the 3270 terminal itself, the 3*x*74 controller also handled a few operations without interrupting the host.

Despite the tremendous flexibility provided by the 3270 datastream, it was no match for what the programmer could do if each keystroke went directly to the host. Character mode allowed complete control of what character was displayed for every keystroke. By exploiting this advantage, some minicomputer applications provided sophisticated user interfaces that weren't possible on the 3270. A good example was the host-based word processing system WORD-11, later marketed by Digital as DECWORD. At the end of 1970s, this ranked in the top three word processors in terms of functionality – no mean feat, considering that the other two were stand-alone single-purpose systems.

The downside of character mode was resource requirements. What users perceive as fast response time when they hit the *Enter* key is positively unacceptable when echoing a keystroke on the screen.

THE ORIGINAL ATTRIBUTE BYTE

As for attribute bytes, later improvements to the 3270 datastream eliminated the wasted screen real estate. Extended field attributes and character attributes were added, neither of which occupied a blank position on the screen.

It's also possible to not define any fields on the screen, creating what's known as an unformatted screen. But although no IBM programming course related to 3270s was complete without learning about this, I've never discovered any practical use for it outside the classroom. A few non-IBM full-screen editors may have offered it as an option, but no one actually used it more than once, after discovering how awful it was.

CLIENT/SERVER

In the early to mid-1990s, there was a lot of talk about mainframe-based client/server replacements for 3270 applications. Then, towards the end of the 1990s, the Year 2000 crisis froze almost all development in that area. As a result, there was little progress beyond screen scraping – trying to build a GUI appearance without changing the 3270 datastream. One notable exception, however, was ISPF, which provided a datastream format of its own to communicate between the mainframe server and its GUI client on the workstation.

Most large organizations standardized the desktop as part of their Y2K effort. This typically meant Windows NT 4 with WRQ's Reflection for 3270 terminal emulation. SNA was replaced by TCP/IP, with special versions of Telnet: tn3270 and tn3270e.

THE ORIGINAL 3270

When initially introduced, the 3270 datastream matched the hardware on which it was to be displayed:

- It was monochrome.
- Fields could be hidden, displayed, or intensified.
- Fields could be protected or autoskipped.
- Fields could also be defined as numeric, but this affected only data entry keyboards and, even there, still allowed entry of most symbols.

Upper-case is not in the list, even though very few of the first 3270s displayed lower-case. Lower-case was an expensive option, adding perhaps $1000 to the already high purchase price. (Even speakers cost extra. While working at my local university, I saw an IBM field engineer at my favourite 3270. He explained that he was removing the little speaker on each 3270 on campus. The university rented all their 3270s, and had decided to cut costs by cancelling the speaker option.)

An even rarer feature was the selector pen – you touched it to the screen, and the 3270 hardware could indicate the field on the screen that was selected. But if you couldn't afford this option, you could:

- Use the cursor positioning keys (arrows) to place the cursor in the desired field
- Hit the *Cursor Select* key.

Lower-case could be used freely in 3270 datastreams; upper-case-only 3270s displayed all lower-case letters in upper-case.

AT ITS SIMPLEST

The 3270 datastream began very simply – especially when you look strictly at the datastream and ignore the headers and trailers of the different communications protocols; SNA was typically used for remote terminals and BSC for local terminals.

From the host to the 3270 terminal, there are only three fields, in the following order:

- Command
- Write Control Character
- Data.

From the terminal to the host, there are also three fields:

- Attention IDentifier (AID)
- Cursor address
- Data.

Data is optional in some situations.

THE 3270 BUFFER

The 3270 datastream relies on the existence of a buffer, originally physically located within the 3270 terminal. Each position of the buffer refers to one position on the screen.

By far the most popular screen size of the 3270 is 24 lines by 80 characters, known as the Model 2. Position zero of the buffer determines what's displayed in line (row) one, column one of the 3270 display screen. Position 79 is column 80 of row one, and position 80 is column 1 of row two.

Field attributes:

- Are one byte in length.
- Indicate the start of a field.
- Control the characteristics of the field.
- Must be stored in the buffer.

As such, they actually occupy a position on the screen – at least in the initial 3270 implementation.

Remember, this was the beginning of the 1970s, before the microprocessor was being mass-produced. So-called intelligent terminals, often less sophisticated than the 3270, were being sold with big price tags. IBM was mass marketing the 3270 at affordable prices for the time. Improvements – extensions to the 3270 datastream definition – would have to wait until technology caught up, in terms of capabilities and pricing.

ALL THE SCREEN, EVERY TIME

All the applications I've seen, including commercial accounting packages, keep it simple. The entire screen is transmitted from host to 3270 display on each terminal write. Systems software products, such as ISPF, are much smarter, transmitting as little as possible. After all, most of the screen usually remains unchanged.

But just because you're running an application doesn't mean that the full screen write will actually reach the 3270 terminal. Nearly 20 years ago, screen optimization software became quite popular, and actually made a big difference in response time on remote terminals, although it would probably be imperceptible today. In those days, line speeds as low as 2.4Kbps were used to connect off-site users. And that was 2.4Kbps total throughput for the site, not for each 3270 terminal at the site.

I ran BMC 3270 Optimizer/CICS VSE and saw substantial improvements. On a tight budget, I had used a null modem to use the remote ports of a 4331 mainframe for local users. As for truly local 3270s, there was some controversy at the time over whether or not the CPU costs of all this optimization outweighed the reduction in CPU time because there was less network traffic to transmit.

Although the software did more, the three basic approaches to optimization were:

- Do not transmit nulls at the end of a field.
- Replace repeated characters with 3270 Repeat to Address (RA) orders.
- Use a soft copy of the 3270 hardware buffer to detect what's already on the screen and need not be retransmitted.

3270 DATASTREAM

So, what does the 3270 datastream actually look like? For older applications, a typical format would be:

- Only the original 3270 field attributes.
- The complete formatted screen contents transmitted for each output.
- Command Code of Erase/Write - EW = X'F5'.
- Write Control Character (WCC) byte with at least the Keyboard-Restore and Restore Modified Data Tag (MDT) bits set.
- Start Field order - SF = X'1D'.
- Field Attribute for the first field.
- Data for the first field.
- Start Field order for the second field, etc.

The screen design of many applications would begin:

- A page title occupying the first row of the screen.
- A field title beginning the second row.
- A field that can be updated beginning right after the title in the second row.

FIELD ATTRIBUTE

In detail, the field attribute for the first field (page title) would have the *Protect Field* bit turned on. The *Numeric* bit would also be turned on, because, when specified with the *Protect Field* bit, it indicates the *AutoSkip* feature: unless the cursor arrow keys are used, the user could not inadvertently position the cursor in the middle of this field. The MDT bit would be off, and bits 4-5 would also be off to indicate *Display* and not *Selector-Pen-Detectable*.

The result is bits 2-7 as 110000 (B'110000'). Bits 0-1 are used to turn the attribute byte into a graphic character, as shown in Figure 1, to avoid confusion with an Order. For this bit sequence, the hexadecimal value is F0 (X'F0').

Bits 2-7	Hex	Bits 2-7	Hex	Bits 2-7	Hex	Bits 2-7	Hex
000000	40	010000	50	100000	60	110000	F0
000001	C1	010001	D1	100001	61	110001	F1
000010	C2	010010	D2	100010	E2	110010	F2
000011	C3	010011	D3	100011	E3	110011	F3
000100	C4	010100	D4	100100	E4	110100	F4
000101	C5	010101	D5	100101	E5	110101	F5
000110	C6	010110	D6	100110	E6	110110	F6
000111	C7	010111	D7	100111	E7	110111	F7
001000	C8	011000	D8	101000	E8	111000	F8
001001	C9	011001	D9	101001	E9	111001	F9
001010	4A	011010	5A	101010	6A	111010	7A
001011	4B	011011	5B	101011	6B	111011	7B
001100	4C	011100	5C	101100	6C	111100	7C
001101	4D	011101	5D	101101	6D	111101	7D
001110	4E	011110	5E	101110	6E	111110	7E
001111	4F	011111	5F	101111	6F	111111	7F

Figure 1: Attribute bits to byte value

NEXT, THE DATA

The 79 bytes following the attribute are the values in the field (page title). This is immediately followed by another *Start Field* order (X'1D'), then the field attribute for the next field (field title that begins the second row), which would be the same as the last: X'F0'. The value of the field title immediately follows, complete with any trailing blanks or nulls to ensure the next field is positioned properly. Otherwise, the *Set Buffer Address* (SBA) order must be used, which requires three bytes.

The next two bytes are the *Start Field* order and the attribute field for the updateable field. The numeric, MDT, and protected bits would be off, and bits 4-5 set to B'01' for normal intensity and selector-pen-detectable. Alternatively, B'11' would be used for an invisible field, such as a password. From Figure 1, the resultant B'000100' would become X'C4'.

The current value of the field would follow. It can be changed by the user and transmitted back to the application for updating the database or file where the field is actually permanently stored.

LOTS MORE TO COME

Future articles will look at:

- Updating only portions of the screen.
- Extended field attributes.
- Character attributes.
- Structured fields.
- Inbound (terminal to host) datastreams.
- Perhaps most important, how the different APIs create the 3270 datastream, CICS and ISPF being two of the most popular environments with 3270 APIs.

George Walker
(Canada)

A practical glossary of Web-to-host terms

The following is a glossary of often used Web-to-host terms.

ActiveX. A reincarnation of Microsoft's Object Linking and Embedding (OLE) technology, geared specifically for Windows-specific Web and networking scenarios. Object-based and closely related to Microsoft COM. In the context of Web-to-host, ActiveX is often a direct competitor to Java applet-based solutions – although ActiveX solutions are always restricted to Windows-based clients using the Internet Explorer browser. Zephyr's Passport eClient 4.0 is an example of an ActiveX-based thin-client terminal emulator.

Apache. A popular open-source, public-domain Web server. The Web server component within IBM's cross-platform WebSphere family is based on Apache.

applet. A browser-invocable program, typically written in Java or ActiveX, which can be dynamically downloaded to the client system from a Web browser. An applet can be cached on the client system to remove the need for repeated downloads. Terminal emulators are now available in applet form – for example, IBM's Java-based Host On-Demand.

Application server. Server-side middleware rich in services (load-balancing, security, state management, etc) and usually with extensive object-oriented capability, to facilitate the development and deployment of new Web-oriented applications. Typically works with a Web server. IBM's WebSphere and BEA WebLogic are examples.

ASP. Microsoft's Active Server Page technology, first introduced with the Internet Information Server (IIS) 3.0 Web Server, and now a popular server-side scripting scheme for creating dynamic, highly interactive Web pages and applications.

autoGUI. Automatic rejuvenation of host screens, as with IBM's Host On-Demand Screen Customizer.

browser plug-in. A client application, usually not Java-based, that can be dynamically downloaded from a Web server when invoked from a Web browser – and then executed on the client. A browser plug-in is required to run Internet Explorer-centric ActiveX software with Netscape Navigator. Zephyr's Passport eClient 5.0 is an example of a browser plug-in thin-client terminal emulator that works with both Internet Explorer and Netscape Navigator.

Cache. The installation of downloaded applets on a client's hard drive, to eliminate the need for repeated downloads. Automatic version checking occurs with the server that originally downloaded the applet each time the applet is invoked, to ensure that users are notified when a newer version of the applet is available. (See also *smart cache.*) Static Web pages, previously displayed, may also be cached at the client or at an intermediary 'cacheing agent' to expedite subsequent accesses.

CIP. Cisco's Channel Interface Processor that permits a Cisco 7000/7500 class bridge/router to be ESCON or bus-and-tag attached to a mainframe – and can optionally act as a tn3270(E) server.

COM. Microsoft's Component Object Model – a Windows-specific object architecture that competes head-to-head with CORBA, the *de facto* industry standard.

Componentization. Customizing an applet's functionality to match the needs of a specific set of users, to eliminate the downloading of unnecessary code to the client. This cuts down applet download times.

COMTI. Microsoft's COM-based Transaction Integration – an extension to Microsoft Transaction Server (MTS) to enable it to support CICS, IMS, DB2, and 3270/5250 transactions.

CORBA. Common Object Request Broker Architecture. Object Management Group (OMG) standard for robust, transparent, network-type-independent interworking of objects distributed across heterogeneous platforms.

Corporate portal. A company-specific Web portal that provides selective access to the company's Intranet and information systems resources.

CPA. Cisco's Channel Port Adapter – a smaller version of the CIP for Cisco 7200 family bridge/routers.

DCOM. Distributed (ie client/server) version of COM.

DES. Data Encryption Standard. A pioneering 40-bit encryption algorithm initially developed by IBM in 1971, and subsequently adopted by the US security agencies. A 56-bit version is now available. DES is a public domain (ie royalty free) encryption algorithm that can be used with SSL and TLS security for Web-to-host access.

DHCP. Dynamic Host Configuration Protocol. A scheme to dynamically allocate IP addresses to clients on-demand.

DHTML. Dynamic HTML. A synergistic combination of HTML, style sheets, and JavaScripts that exploit the newest capabilities in Version 4 (and above) browsers, to allow the development of animated and interactive Web pages.

DLSw. Data Link Switching. Now a standard for safely transporting SNA/APPN traffic across TCP/IP backbones. Initially introduced by IBM on the star-crossed 6611 bridge/router.

DNS. Domain Name Server. Basic directory mechanism in IP to map resource names to IP addresses

e-business. Encompasses the entire realm of all electronic business-to-business (b2b) and business-to-commerce (b2c) dealings across the Web.

e-commerce. A subset of e-business, dealing with the entire process of buying and selling items over the Web.

EJB. Server-side, transaction-oriented extension to the JavaBean component model.

Enterprise Extender. IBM marketing-speak for routing HPR over IP.

G2G. Green-to-GUI. A cute term for user interface rejuvenation.

Green screens. Shorthand for ‘green-on-black’ screens – ie 3270/5250 screens.

Green-on-black screens. 3270 or 5250 screens.

HACL. IBM’s Host Access Class Library. Object-oriented, Java API implemented on Host On-Demand that is now also available as an API on other Web-to-host products, eg Hummingbird’s HostExplorer Web.

Host integration server. Server-side middleware, with extensive connectivity options to disparate host systems, for the creation of new e-business applications that rely on the manipulation and synthesis of data from multiple, existing mission-critical applications, eg IBM's Host Publisher Version 2.2.

Host on-demand. IBM's popular Java applet that provides tn3270(E)-, tn5250-, or VT-based host access.

Host publishing. Host datastream (eg 3270, 5250, VT) to HTML conversion, typically performed dynamically (ie on-the-fly), that enables host applications to be accessed directly from a standard Web browser. Also known as 3270/5250-to-HTML conversion. iE's ScreenSurfer is an example of a host publishing product.

HPR. High Performance Routing. The successor to APPN (and as such also to SNA).

HPT. Host Print Transform. An OS/400 function that does code translation and formatting to enable AS/400 print output to be spooled to PC printers.

HTML. HyperText Mark-up Language. The language used to describe the format and content of Web pages.

HTTP. HyperText Transfer Protocol. A connectionless, layer 4 protocol used on top of IP for data transfer between Web servers and Web browsers. Some applet-based host emulators now offer the option of using HTTP rather than tn, in conjunction with a Web server proxy, as a means of easily traversing firewalls using the existing firewall definitions for Web server access.

HTTP server. Another name for a Web server – primarily in 'IBM-speak'.

HTTPS. Secure HTTP, where security in the form of SSL is superimposed on top of HTPP on a transaction-by-transaction basis.

IIOP. Internet Inter-ORB Protocol. A standard for CORBA-based object interworking across IP-based networks.

IIS. Microsoft's ubiquitous Internet Information Server Web server.

IKE. Internet Key Exchange infrastructure, which permits encryption keys to be automatically and securely created, distributed, and refreshed according to the protocol requirements of IPSec.

Internet Explorer (IE). Microsoft’s Web browser.

ip3270/ip5250. A direct alternative to tn3270(E)/tn5250 that is supported by Microsoft’s SNA Server and Novell’s NetWare for SAA – whereby gateway-specific TCP/IP protocol can be used for SNA access between the client and the server. One of the key advantages of this approach, as opposed to ‘tn’, is that it supports LU 6.2-based transactions.

IP-assist. The TCP ‘off-load’ feature available on IBM’s OSA-Express adapter.

IPDS. IBM’s Intelligent Printer Data Stream. A cornerstone of IBM’s Advanced Function Presentation/Printing (AFP) initiative.

IPSec. IP Security. A network layer (ie layer 3) security that can be used to realize authentication, integrity, and data privacy between two IP entities. Heavily used to implement VPNs.

J2EE. Java 2 Platform, Enterprise Edition. The Java Software Development Kit, tools, APIs, and the Java runtime (ie execution environment) targeted at Java developers building enterprise-class, server-side applications.

Java. Popular, widely endorsed, platform-independent, object-oriented programming scheme introduced by Sun Microsystems circa 1995.

Java 2. Umbrella term introduced by Sun in December 1998 to refer, henceforth, to all subsequent Java initiatives and products.

Java application server. Application server written in Java, and as such platform independent, geared to facilitate the development and deployment of Java-based applications, applets, and servlets – eg BEA WebLogic.

Java platform. In essence what used to be called the JDK before Java 2. Consists of a Java Software Development Kit, tools, APIs, and a Java runtime.

JavaBeans. Client-side Java objects to facilitate the re-use of program components between different applications.

JavaScript. A Java-like, compact, object-based scripting language supported by modern browsers. Can be used to enhance the functionality and capabilities of Web pages. JavaScript is also widely used by host publishing and host integration solutions, within the context of Web pages, to increase the fidelity of host input functions – eg emulation of 3270 input fields.

JDBC. Java equivalent of ODBC.

JDK. Software development kit from Sun consisting of a Java compiler, a debugger, standard Java classes, and a Java runtime (ie JVM) for Unix. Nowadays referred to as a Java 2 Platform (as in J2EE) or a Java Software Development Kit (SDK).

JSP. JavaServer Pages. Java equivalent of Microsoft ASP methodology.

JUMBO frames. Optional enhancement to Fast/Gigabit Ethernet adapters (eg IBM's OSA-Express adapter) to permit 9,000-byte frames, as opposed to the standard 1,518-byte frames, to be transmitted to expedite data transfers.

JVM. Java Virtual Machine. A platform-specific environment for executing Java applets. Version 3 (and greater) of Netscape Navigator and Internet Explorer include JVM implementations so that Java applets can be executed by the browser. Of late, Java Runtime Environment is used in place of JVM.

LDAP. Lightweight Directory Access Protocol. An increasingly popular and important standard for directory management that is now being used by some Web-to-host products, eg Hummingbird's HostExplorer Web, for user authentication.

Lipstick on a bulldog. A rejuvenated GUI interface (ie the lipstick) that hides the dated 'green-on-black' interface of a host application and tries to disguise the vintage of the application.

ODBC. Open Database Connectivity. A popular standard for transparent database access.

OHIO. Open Host Interface Objects. An IBM and Attachmate inspired object-oriented host access API now being ratified as an industry standard.

OSA-Express. New network-to-mainframe adapter for System/390 G5/G6 models which can work at full-duplex speeds up to 2.66Gbps – and supports Gigabit Ethernet, Fast Ethernet, 155Mbps ATM.

Persistence. Emulation of an end-to-end connection over IP, as in an SNA session, to ensure client/server integrity and security in Web-to-host access scenarios (in particular, browser-to-host schemes) given that HTTP *per se* is a connectionless protocol.

Portal. An entry point (implemented as a Home page) for accessing Web-based information. America Online (AOL), Yahoo, and Excite are examples of Internet portals.

PPR/PPD. Page Printer Requester/Page Printer Daemon. IPDS over TCP/IP.

QDIO. Queued Direct I/O. IBM's turbo-charged network-to-mainframe memory transfer protocol that works with the OSA-Express adapters available for System/390 G5/G6 models.

RC4 or RC/4. A royalty-based, RSA encryption algorithm (originally developed by a company called Rivest) that is widely used with SSL for Web-to-host security. For example, it's one of the encryption algorithms supported by IBM's Host On-Demand.

Rejuvenation. Revitalizing host applications by providing them with a contemporary point-and-click graphical user interface (GUI) in place of their original, harsh, mainly textual, green-on-black screens.

RMI. Remote Method Invocation. Distributed object invocation, as in Java, that enables Java programmer to remotely invoke the methods of remote Java objects even if they're on a different host. A Java competitor to IIOP.

RSA. RSA Security. The leading provider of Web-oriented security technology, including encryption algorithms invoked by SSL.

Self service portal. A portal that permits Web browser users to easily perform their own inquiries and transactions, on-line, without having to interact with a telephone-based call centre.

Servlet. Server-side Java applet.

SLP. Server Location Protocol. An IETF standard written by IBM and Zephyr to facilitate optimum load balancing between tn3270(E)/tn5250 servers.

Smart cache. IBM improvement to conventional applet cacheing, pioneered with Host On-Demand 5.0, where the older version of an applet is activated while a newer version of it is being downloaded in the background.

SNASwitch. Cisco marketing-speak for IBM's Enterprise Extender.

SOAP. Simple Object Access Protocol. An IBM and Microsoft endorsed XML and HTTP-based protocol for accessing services, objects, and servers in a platform-independent manner.

SSL. Secure Sockets Layer. Widely used end-to-end, client/server security scheme developed by Netscape that deals with both user authentication and encryption facilitation. SSL *per se* does not perform the encryption. It activates the appropriate encryption algorithms such as those available from RSA Security.

Thin-client. In the context of Web-to-host, thin-client applications are those that can be dynamically downloaded from a Web server when invoked from a Web browser. Applets and browser plug-ins are examples of thin-clients. Microsoft's NT Terminal Server and Citrix's Terminal Server also provide thin-client solutions where the application executes on the server rather than on the client. Network computers (low-cost PCs with little or no hard drive capacity) that execute software dynamically downloaded from a server are also referred to, in some quarters, as thin-clients.

TLS. Transport Layer Security. The IETF variant of SSL which, over time, will gradually take the place of SSL. TLS is backward compatible with SSL 3.0.

tn. Shorthand for tn3270(E) and tn5250 – as in 'tn'-server,

tn3270(E)/tn5250. Strategic, proven, and widely-used industry standard for accessing SNA applications from TCP/IP clients across IP networks.

Triple-DES. Public-domain, 168-bit version of DES where the key is dynamically changed between 'steps' (or transactions).

UDDI. Universal Description, Discovery, and Integration. A nascent initiative by IBM, Microsoft, and e-business heavyweight Ariba to create a shared business registry on the Web to expedite and streamline b2b e-business processes.

URL. Universal Resource Locator. The 'address' of a Web site or Web page as specified to a browser in the format: http://www.xxx.yyy

VIPA. IBM OS/390 Virtual IP Addressing. A means of correlating multiple virtual IP addresses with a real IP-address to facilitate fault-tolerance via transparent fail-over.

VPN. Virtual Private Network. Encryption and tunnelling technology that permits Internet bandwidth to be securely and profitably exploited for intra-company data transfer. VPN in effect enables a corporation to create a secure intranet inside the public Internet.

WAP. Wireless Application Protocol. A once high-profile *de facto* standard to enable low-bandwidth mobile devices to effectively and efficiently browse and manipulate Web-based content.

WebSphere. IBM's strategic, cross-platform e-business application server, complete with a built-in Web server.

WLBS. Windows NT Load Balancing. Microsoft's preferred approach to overcoming the scalability limits of NT servers.

WML. Wireless Mark-up Language. A cross between XML and HTML, used with WAP, to describe the format and content of data exchanged with wireless devices.

X.509 certificates. Digital ID-cards (also known as digital certificates) used to authenticate Web users. Web-to-host products such as IBM's Host On-Demand are now starting to use these certificates to authenticate users in conjunction with established mainframe access control systems such as RACF.

XML. Extensible Mark-up Language. An increasingly strategic tag-based 'self-defining' mark-up language, within the growing Generalized Mark-up Language (GML) family, that is used to provide data with meaning, structure, and context. HTML, by contrast, deals just with the display, formatting, and user interaction aspects of Web pages. XML is becoming the preferred means for interchanging data between new e-business applications, since the data is in effect ‘self-descriptive’.

Zero footprint. Browser-only, thin-client, host-access solutions that don’t require any terminal emulation-related applets or software at the client in order to facilitate host interactions. 3270/5250-to-HTML conversion solutions are an example of a zero footprint host access scheme.

Anura Gurugé
Strategic analyst (USA)

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*, 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.

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

NetView and network management

When IBM announced NetView in May 1986, the new MVS and VM product combined NCCF, NLDM, and NPDA, with portions of VNCA and NMPF. Nearly everyone used some or most of these products to manage their VTAM SNA networks. They are:

- Network Communication Control Facility (NCCF)
- Network Logical Data Manager (NLDM)
- Network Problem Determination Application (NPDA)
- VTAM Node Control Application (VNCA)
- Network Management Productivity Facility (NMPF).

Unlike many of the workstation office suites of the early 1990s, NetView included a significant amount of glue to hold the distinct products together, in terms of code added for integration. And almost enough new features were added to make you feel that IBM had included another new product in the blend.

I remember the migration from the predecessor products to NetView as being a very positive experience on both VM and MVS. Admittedly, we had been a few releases back on those predecessor products and were upgrading both hardware and operating systems versions at the same time. I even enjoyed learning REXX to rewrite some of the VTAM LU definitions used by NetView in VM.

Even before NetView was available (21 November 1986, initially for MVS only), other products were announced bearing the NetView name. And soon, existing products were renamed to include the NetView name. 'Multiple Applications from a Single Screen' had been an MVS-only program offering that was now upgraded to a licensed product, also offered in VM, and renamed NetView Access Services, also known as NetView/Access, NetView AS, or NVAS.

ENTER TIVOLI

In March 1996, IBM bought Tivoli for US$743 million. As with the

Lotus acquisition the year before, IBM viewed Tivoli's corporate culture as an asset, and made little, if any, attempt to turn the employees into IBMers or to transfer products to IBM. In fact, quite the opposite occurred, as many major IBM software products became Tivoli products – including all NetView products.

In fact, it was three former IBMers who had formed Tivoli in the first place, in August 1989. The company made its mark quickly (remember TME 10?), and is best remembered for making it easy to manage environments with widely divergent products and software. A single workstation-based operator console controlled an entire environment. For example, in a network, the same operator actions in the same Graphical User Interface (GUI) would halt any device.

It was all done, of course, by partnering with all the major software and hardware players in each type of environment, eg network, database, etc. Each player would provide Tivoli with either documentation on existing interfaces to their products, or would work with Tivoli to create interfaces that Tivoli could use to provide the level of control and information required to fully manage the product from Tivoli's operator console.

'TME 10' was used to prefix nearly every product, including many that were acquired from IBM, such as ADSM. But this soon became unwieldy, and the TME acronym is now gone and many of the products have been given more sensible names. ADSM, for example, is now simply Tivoli Storage Manager. The previous TME 10 ADSM name looks compact until you have to spell it out: Tivoli Management Environment 10 Adstar Distributed Storage Management. Worse yet, try to explain where the '10' and 'Adstar' came from.

NETVIEW TODAY

Today, there are eight NetView products:

- Tivoli NetView
- Tivoli NetView Access Services
- Tivoli NetView Distribution Manager
- Tivoli NetView File Transfer Program

- Tivoli NetView for OS/390
- Tivoli NetView-IT Director Edition
- Tivoli NetView Performance Monitor
- Tivoli NetView Performance Monitor for TCP/IP.

That original 1986 product, then known simply as NetView, is now Tivoli NetView for OS/390. The product now known simply as Tivoli NetView manages TCP/IP networks, and runs on a variety of Unix and Windows NT/2000 platforms. Meanwhile, Tivoli NetView for OS/390 has expanded beyond SNA networks to also include TCP/IP. Both support SNMP-based devices.

To be absolutely correct, it's worth pointing out that Tivoli NetView for OS/390 is actually the integration of three IBM products:

- NetView for MVS/ESA
- NetView MultiSystem Manager (MSM)
- SystemView Automated Operations Network (AON).

This is worth keeping in mind, as you'll still see occasional references to the individual products, even in the titles of the current NetView manuals!

WHAT IS NETVIEW?

Tivoli NetView for OS/390 offers network management for both the mainframe and distributed environments, all the way to the desktop. If you read Tivoli's current datasheet for the product (http://www.tivoli.com/products/documents/datasheets/nv_os390_ds.html), you'll note that 'Enterprise Management from a Single Console' is the first capability highlighted. The promise of controlling and obtaining information on everything from the mainframe to the desktop from a single workstation application with a single GUI is very appealing. The fact that this goal was one that was shared by both NetView and TME 10 helps to make sense of IBM's decision to buy Tivoli and give it NetView.

The word 'console' is perhaps no longer the best way to describe what NetView offers. It conjures up pictures of the MVS operator console display – a 20-line view of a high-speed flow of often confusing messages rolling by at high speed. Whether they are answered by automated operations software or by a human console operator, the commands are cryptic, to help speed up the response of a mostly non-touch-typist world of operators.

That was never what the NetView console looked like, even when I first saw it in 1988. VTAM LUs were displayed and you moved the cursor to select one and drill down to see details (and devices) of the connection/path that that LU represented.

3270 TO GUI

No longer restricted to the 24x80 non-graphics 3270 screen of the late 1980s, connections can now be graphically displayed on workstations, allowing you to double click anywhere with a mouse to zoom in on that portion of the connection. Geographically dispersed networks can be displayed in the context of a world map shown behind the connection.

Alternatively, an entire network can be shown in the same tree form used by many Windows-based applications. Mouse click on a plus sign to the left of any item displayed to see the next layer of the hierarchy below that item. Click on a minus sign to collapse the level of detail for that item.

There are also all the features you would expect from a modern GUI-based application, including: tool bars, scroll bars, menu bars, drop-down menus, pop-up menus, buttons, and window sizing options.

Obviously, however, you can’t drag and drop a piece of hardware to have it physically moved from one side of the country to the other!

CHOICE OF UI

In fact, you have a choice of four user interfaces (also known in Tivoli talk as ‘management interfaces’):

- The default, and very GUI, Java-based Tivoli NetView Management Console.

- The Tivoli NetView Web browser console, allowing secure access from any Web browser anywhere.
- A Java implementation of 3270 console functions.
- The same text-based consoles that have always been available.

Yes, that means that, if you just stepped out of a time machine from the mid-1980s, or just left your job as a VTAM systems programmer at an MVS/370 site, you can still use the same pre-NetView product commands, and see a familiar console. In fact, you'll still see occasional references to the predecessor products. For example, in the *Command Reference manual*, in 'Chapter 3 – NetView Base Commands', parentheses in the title of each command still indicate the predecessor product from which the command came. How else would I remember that NCCF gave us the LISTSESS command? If you don't believe me, take a look for yourself at:

http://www.tivoli.com/products/index/netview_390/library

Select *Reference* from the left sidebar, then *Command Reference*, then *Chapter 3*.

ADD TCP/IP WITH SNMP

By combining NetView MultiSystem Manager (MSM) with NetView for MVS/ESA, Tivoli NetView for OS/390 moved beyond SNA-only network management to also include TCP/IP networks. Adding support for SNMP is at the heart of this change.

Simple Network Management Protocol (SNMP) was introduced by the Internet Activities Board (IAB) in 1988, just one year after the release of the Simple Gateway Monitoring Protocol (SGMP). Within two years, SNMP had effectively replaced SGMP and was widely available in vendor implementations of TCP/IP. This was the prerequisite first step for SNMP to be universally supported by all TCP/IP-based network components, most notably the hardware devices that are key to making it work.

As 'S' for 'Simple' in both protocols indicates, SGMP and SNMP were each proposed as stop-gap measures until something better came along. Common Management Information Protocol, CMIP (pronounced C-mip), was conceived as that something better. But even though CMIP provides improved security and better reporting of unusual network conditions, it was just too complicated for the network hardware of the time, and SNMP was already widely used. Today, network components still exist that support CMIP, including some with the IBM nameplate on them, but any that I have seen also support SNMP.

But SNMP has not stood still either. The original 1988 definition is now known as Version 1 (SNMPv1). Version 2 was completed in 1993, with, among other things, more granular error responses. In April 1999, Version 3 was issued, focusing on security, which was a major weakness of previous versions.

HOW SNMP WORKS

From the SNMP viewpoint, the world really is a simple place. Network managing stations send requests to, and receive responses from, the network elements (SNMP agents) they are managing. Information derived from the responses is stored in a virtual information store known as the Management Information Base (MIB).

The MIB defines the managed objects, such as interface statuses, routings, and packet counts, that may be used to monitor and control network elements in a TCP/IP-based network. The Structure of Management Information (SMI) defines the rules by which these managed objects are named and defined.

An excellent and more detailed technical introduction to SNMP can be found in Chapter 1 of the new (June 2000) IBM Redbook *Managing OS/390 TCP/IP with SNMP* (SG24-5866), from which some of the material above has been derived. It's accessible from *Redbooks On-line* at

http://www.redbooks.ibm.com

The direct URL is

http://www.redbooks.ibm.com/redbooks/SG245866.html

An Adobe Acrobat (.pdf) version of the manual is also available. It very precisely delivers a view of the paper version of the manual to your screen, but IBM has recently begun requiring an id and password to access it. They are readily available (no charge), but do require a one-time registration process.

HOW LOW CAN YOU GO?

Remember the late 1980s? SNMP came out at a time when workstations first appeared on every white-collar worker's desk. Every manager wanted an application to run his business, providing both processes and information analysis. That meant relational databases and a big growth spurt for mainframe hardware, especially networks.

All this was readily approved by executives, but they were less inclined to approve the kinds of increases in operations staff required to run the many much more complex applications that were being added. So, first we had automated operations, to allow existing staff to continue to man the much busier consoles. Then came network management, to keep the much larger network running smoothly.

I vividly remember the initial proposal of the concept that each network component could be centrally managed. I laughed about it with a peer at the time. "Yeah, right. That dumb lump of metal over there, affectionately known as a modem, with enough smarts in it to respond to a bit-string meaning 'Hi, how are you?' from a network management software package on the mainframe."

Today, with that in mind, I picked up my two-year-old ADSL modem, supplied by my local telco to give me cheap, high-speed Internet access, and sure enough, it has a mini-RS232 port on the back labelled 'management'. Even though the original manufacturer has since been taken over by Cisco, and the modem is no longer available in its original form, I was able to verify that it supports SNMP.

How low can you go? Using SNMP, network management software like NetView can inquire all the way down to the lowest level of network components, to determine just where the fault lies when a circuit fails. Using the same approach as automated operations uses to resolve mainframe console problems without human intervention – document scenarios and possible solutions – NetView can reduce operations manpower by requiring human intervention only when all possible automated solutions have been exhausted – such as a backhoe digging up a buried cable for which no back-up exists.

EXTENSIONS

Unfortunately, the ‘S’ for ‘Simple’ in SNMP also means that SNMP does not do everything that every vendor wants from a network management protocol, so there are the inevitable extensions. And Tivoli NetView for OS/390 includes support for many of them. It also works in co-operation with other TCP/IP-based management tools, such as Tivoli NetView for NT, Tivoli NetView for Unix, and HP OpenView.

SNA BEGINNINGS

Of course, it was SNA where NetView began. In fact, NetView component products like NCCF go so far back that many an operator will probably tell you that you cannot control an SNA network without NetView. From personal experience – over five years in the early 1990s in a NetView-free environment – I can attest that it is possible.

You can do almost anything with a VTAM VARY command, but it’s not a good idea if you have minimal typing skills or are pressed for time. With a console of its own, NetView was not encumbered by the fact that almost all the one-letter main console operator commands had been taken by the time VTAM arrived on the scene.

Pre-NetView component NCCF was simple to justify because of its ability to really reduce the operator keystrokes required to control an SNA network. Between long VTAM commands and the 8-character alphabet soup known as an LU (VTAM Logical Unit), a little effort by a systems programmer setting it up made a whole world of difference in the life of an operator. And it’s still true with NetView today.

NETVIEW FUNDAMENTALS

NetView, of course, does a lot more. It’s best known for its ability to take a large complex network and give you the kind of compact macro and micro views that you need to actively, even proactively, manage it:

- ‘Macro’ in the sense of a graphical overview of your network connections, with any potential problem areas highlighted. In other words, a picture of the overall health of the network. Plus, the ability to drill down for more detail.
- ‘Micro’ in the sense of an ‘exception view’, where, all in one place, any exception conditions anywhere across the network can be viewed. This also provides an excellent launch point for investigation of any of these exceptions.

Less obvious, but also very important, is the ability to automate much of the routine network management. Commonly occurring solutions can be implemented automatically, especially those that never cause other problems and are a response to a problem that is easy to identify programmatically.

If this sounds like the kind of automated operations that took over most large host operator consoles, you should not be surprised to also hear talk of lights-out operation in connection with NetView. Tivoli NetView for OS/390’s ‘unattended option’ allows the networks of remote OS/390 hosts to be centrally managed.

TIVOLI NETVIEW 6.0

As mentioned above, Tivoli NetView no longer refers to the OS/390 version. The client/server implementation runs on:

- IBM RS/6000 with AIX
- Sun Sparc 5 or greater with Solaris
- Intel Pentium with NT 4.0 or Windows 2000
- Digital Alpha with NT or Tru 64 Unix (formerly Digital Unix).

Tivoli NetView Mid-Level Manager (MLM) provides management functions to remote locations without the need to run dedicated network management systems in those locations. MLM runs on all of the above platforms, plus HP-UX.

You may see references to NetView for NT, NetView for Unix, NetView for AIX, or the other logical extensions based on the supported platforms. All refer to the same NetView 6 product, just the different operating system implementations of it.

NetView 6 supports only TCP/IP networks, but it's important to point out that the platforms listed above indicate where NetView will run, not what it can manage. Nor where it can be managed from, since MLM allows management from virtually any platform. And, of course, there's the Tivoli NetView Web console, which runs anywhere you can run a Web browser.

Tivoli strives to keep on top of the latest network management challenges. For example, NetView 6 supports Cisco's Hot Standby Router Protocol (HSRP) and unnumbered serial interfaces. Both technologies provide organizations with greater network flexibility, but can also turn into network management headaches.

POLICY-BASED MANAGEMENT

Tivoli NetView Smartsets allow you to group devices you wish to manage in a similar manner, and apply policies uniformly to all devices in the Smartset. For example, if all Cisco routers were grouped together, a policy could be set to collect CPU utilization for each every 15 minutes. NetView can discover any new Cisco router as it's installed on the network and automatically add it to the Smartset. The net result is that CPU utilization is automatically collected every 15 minutes from the new router.

Beyond full coverage of automated event response, NetView 6 has a very useful Router Fault Isolation feature. In the event of a serious network problem, Router Fault Isolation immediately focuses on the failing device and marks affected network regions as unreachable. In response, NetView reduces networking polling of the affected networks, thereby reducing overall event traffic. After all, there's nothing worse than making a bad situation (the original failure) worse by driving network traffic towards it.

TIVOLI IT DIRECTOR

Tivoli IT Director is aimed at small and medium-sized organizations, with up to several hundred employees. It provides a streamlined interface to a broad range of Tivoli products, including NetView. It's also sold through authorized resellers who can customize it for each organization, reducing the IT workload, especially during installation.

Tivoli NetView – IT Director Edition is a special version of Tivoli NetView 6.0 that runs on an Intel Pentium-based processor running Windows NT 4 Workstation or Server. Although it's ODBC-compliant, it does require SQL Server and Microsoft Internet Information Services (IIS).

PERFORMANCE MONITORS

Remember NPM? It's still around, but now in two flavours:

- Tivoli Network Performance Monitor (NPM)
- Tivoli Network Performance Monitor for TCP/IP (NPM/IP).

Both run on OS/390, have workstation client interfaces, and manage networks attached to an OS/390 host. NPM handles SNA and NPM/IP TCP/IP.

Although it is not a prerequisite for either, both NPM and NPM/IP can send alerts to and receive commands from Tivoli NetView for OS/390, and both were designed with it in mind. Workstation clients can be Windows 9*x* and NT, although NPM also supports OS/2.

ADSM

Remember ADSM (ADSTAR Distributed Storage Manager)? It began (under yet another name) with mainframe-based back-up/restore across most of the platforms you might expect to find in a large organization. ADSM Distribution Manager (DM) added software and data distribution.

Currently, the base product is Tivoli Storage Manager, and DM is Tivoli NetView Distribution Manager (NvDM). Tivoli is working hard to enhance its existing Tivoli Software Distribution product so that it will have all the features of NvDM. When it has finished, a migration path will be offered for existing NvDM users, and, presumably, a date after which NvDM support will no longer be available.

Tivoli Storage Manager, System/390 Edition runs on OS/390, MVS/ESA, and VM/ESA hosts, and can currently manage storage on AIX, Apple Macintosh, HP-UX, NetWare, OS/390 Unix System Services, OS/2, SCO UnixWare, Sequent PTX, SGI IRIX, Sun Solaris, Tru 64 Unix, Windows 32-bit Intel (Windows 9x/NT/2000), and Windows 32-bit DEC Alpha (Windows NT).

Beyond the mainframe, Tivoli Storage Manager Version 4.1 can also be run as a server on Windows NT/2000, AIX, HP-UX, and Sun Solaris.

It supports 39 client platforms.

DISTRIBUTION MANAGER

The current version of NvDM is Release 6.2, originally IBM NetView Distribution Manager for MVS (5685-016) when it was first announced in 1998. It's still available and supported, and can distribute software and data to a broad range of platforms. It uses:

- SNA LU 6.2 protocol to distribute software to VSE/ESA, System/36, AS/400, workstations with LANDP or PCNM, 468*x* and 469*x* Store System processors, DPPX/370, and RS/6000 with AIX.
- SNA LU 0 protocol to distribute to:
  - another directly attached NvDM.
  - 3174.
  - AS/400 with SystemView Managed System Service/400.
  - workstations running NetView DM for OS/2, NetView DMA for DOS, NetView DMA for Windows, or NetView DM for NetWare.
  - RS/6000 with NetView DM for AIX.

And it goes through an RS/6000 running NetView DMA for AIX to reach:

- HP-UX with NetView DMA for HP-UX
- Sun Solaris with NetView DMA for SunOS or Solaris.

The current version of Tivoli Software Distribution for OS/390 (5697-F03) is Version 3.6.1, which has been available for just over a year. As its name implies, it runs only on OS/390. It also requires recent versions of:

- Tivoli Management Framework for OS/390 (5697-D10)
- Tivoli Management Framework (5697-FRA)
- Tivoli Software Distribution (5697-SWD).

Note that the latter two products do not run on the mainframe, but can run on a variety of other platforms:

- RS/6000 AIX
- NetWare
- OS/2
- Solaris
- Windows 9x/NT/2000.

FTP

Tivoli NetView File Transfer Program is a high-performance ftp product to get data to and from the mainframe and other hosts. One side of the transfer must be an OS/390 host, and the other can be OS/2, AIX, or OS/400.

MULTI-SESSION MANAGER

Tivoli NetView Access Services allows multiple VTAM applications to be accessed from a single 3270 user session, without the need to end one before beginning another. Hot keys are provided to allow single keystroke switching between applications. Alternatively, users can choose from a menu of available applications.

The current version, 2.1.1, supports MVS and OS/390, and offers the ability to define terminals that will bypass the log-on screen.

A REXX API allows extensive customization to be performed. RACF is not a requirement; other security products can be used.

A LOT OF PRODUCTS

There are a lot of NetView products. But many of them can, and a few must, interface with other Tivoli products, not all of which bear the NetView name.

But then the same applies for IBM mainframe products. There are the requirements, the most obvious usually being the OS/390 operating system, and the optionals. For example, if you want to use the supplied REXX API within a product, you must have a version of TSO/E installed that is new enough to include REXX. It's not always easy to figure out, but definitely worth the effort before committing to any software product. Managers hate that kind of monetary 'oops'.

VM

NetView 2.3 is still available and is still supported for both VM/ESA (5756-051) and VSE/ESA (5686-055).

The last release of NetView Access Services that ran on VM was Version 1.3.2. It was still under the IBM name when availability ended in 1995.

As mentioned previously, the ADSM base product – most notably for back-up/recovery of a broad range of platforms – has been replaced by Tivoli Storage Manager, which will run on VM/ESA. It provides exactly the same functionality as ADSM, without the enhancements added to the MVS/ESA and OS/390 version. The best you'll find for distributing software is Release 2 of NetView Distribution Manager for VM/SP (5684-017), which was first released in 1988 and has not been available since 1997, but is still supported.

First available in 1994, Version 2.2 of NPM (5654-011) is still available for VM/ESA and requires VSE/VSAM. NPM for TCP/IP never ran on VM.

SOME WORDS OF WARNING

Even if you've managed to get a handle on all the different Tivoli NetView products, you're still faced with one major problem. Despite the fact that NetView is listed by Tivoli as a registered trademark, you'll see a lot of products, services, even companies bearing the NetView name in one form or another, and which have nothing to do with Tivoli. You'll even find netview.com, which is owned by a San Francisco area Web company called NETView Communications.

Note also that the Tivoli Web site does not provide complete information on all products, especially those recently acquired from IBM. Researching for this article, I found the site sometimes confusing, occasionally misleading, and perhaps even wrong (out of date) at times. IBM's Web site, especially *Announcements on IBMLink*, is a good place to verify information or get more information than Tivoli has to offer. And although I find IBMLink's *Sales Manual* not as complete as *Announcements*, it does provide a central resource for determining availability, support status, and replacement products – once you get used to its format.

Armand Minet
(Canada)

A REXX solution to a mixed Token Ring and Ethernet LAN environment

If you've ever worked in a LAN environment where both Ethernet and Token Ring topologies were supported, it probably wasn't long before you encountered the issue of converting MAC addresses from MAC bit order to canonical form. In an Ethernet (802.3) environment, adapter addresses appear in canonical or IEEE illustrative form. In a Token Ring environment, adapter addresses appear in MAC bit order form. Simply stated, for canonical form MAC addresses the most significant bit is on the right, while for MAC bit order addresses the most significant bit is on the left. Although some network devices and protocols can adjust for this, others cannot.

When we first supported client connections via routers, we used Proteon routers, which supported a FLIP command which would automatically convert the MAC address from MAC bit order to canonical form or vice versa. Then, as we began to support more routed connections, we migrated to Cisco routers. Although these offer a number of advantages, they have no FLIP command. This meant that, in environments that didn't automatically convert MAC addresses, we needed to perform the conversion manually. In fact, you still need to perform the conversion manually even where the networking equipment does automatically convert the MAC addresses. When tracing network connections, you may sometimes encounter some addresses that seem incorrect. These may simply be in a converted form.

The underlying technique for the conversion is relatively simple. All that you have to do is reverse the bits. This can be accomplished in three steps:

- Convert the hexadecimal address into binary format
- Reverse the order of the binary digits
- Convert back into hexadecimal.

We've tried various ways of accomplishing this, before settling on a solution that suits us.

1. 4000.3745.1088						(Original address – MAC bit order)
2. 40	00	37	45	10	88	(Separate)
3. 01000000	00000000	00110111	01000101	00010000	10001000	(Binary)

Figure 1: Separating the MAC address and converting pairs into binary

4. 00000010	00000000	11101100	10100010	00001000	00010001	(Reverse)
5. 02	00	EC	A2	08	11	(Convert/Hex)
6. 0200.ECA2.0811						(Converted address – Canonical form)

Figure 2: Rewriting digits in reverse order and converting back into hexadecimal

OUR FIRST METHOD

Let's start with the MAC address of 4000.3745.1088 (separating each group of four digits with a period to improve readability). First, we'll separate our MAC address into six pairs of hexadecimal digits and convert those pairs into binary (see Figure 1). Then, for each set of eight binary digits, we'll rewrite those digits in the reverse order. Then, we'll convert the newly ordered binary digits back into hexadecimal (see Figure 2).

The final step is to recombine the six pairs of hexadecimal digits into a converted 12-digit hexadecimal MAC address. Our converted address is 0200.ECA2.0811.

Although this process is not labour-intensive, it is completely manual and therefore prone to error. Many people have had difficulty deriving the correct answer by following these steps. We therefore needed to find a more error-free process.

FINDING A BETTER PROCESS

Our next process revolved around a conversion table like the one shown in Figure 3. The table method had the advantage of removing

N/c	C	N/c	C	N/c	C	N/c	C
00	00	40	02	80	01	C0	03
01	80	41	82	81	81	C1	83
02	40	42	42	82	41	C2	43
03	C0	43	C2	83	C1	C3	C3
04	20	44	22	84	21	C4	23
05	A0	45	A2	85	A1	C5	A3
06	60	46	62	86	61	C6	63
07	E0	47	E2	87	E1	C7	E3
08	10	48	12	88	11	C8	13
09	90	49	92	89	91	C9	93
0A	50	4A	52	8A	15	CA	53
0B	D0	4B	D2	8B	D1	CB	D3
0C	30	4C	32	8C	31	CC	33
0D	B0	4D	B2	8D	B1	CD	D3
0E	70	4E	72	8E	71	CE	73
0F	F0	4F	F2	8F	F1	CF	F3
10	08	50	0A	90	09	D0	0B
11	88	51	8A	91	89	D1	8B
12	48	52	4A	92	49	D2	4B
13	C8	53	CA	93	C9	D3	CB
14	28	54	2A	94	29	D4	2B
15	A8	55	AA	95	A9	D5	AB
16	68	56	6A	96	69	D6	6B
17	E8	57	EA	97	E9	D7	EB
18	18	58	1A	98	19	D8	1B
19	98	59	9A	99	99	D9	9B
1A	58	5A	5A	9A	19	DA	5B
1B	D8	5B	DA	9B	D9	DB	DB
1C	38	5C	3A	9C	39	DC	3B
1D	B8	5D	BA	9D	B9	DD	DB
1E	78	5E	7A	9E	79	DE	7B
1F	F8	5F	FA	9F	F9	DF	FB

N/c = Non-canonical; C = canonical

Figure 3: Conversion table (part one)

the binary conversion steps, and reduced the problem to a series of look-up operations.

Using the table, you start by writing down the 12-digit address on paper. Next, separate the address into six pairs or bytes. For each pair that you've written, locate that pair of hexadecimal characters in the conversion table above (in the non-canonical columns) and replace them with the converted pair in the canonical column to the immediate right. The last step is to recombine the six pairs of numbers into a converted 12-digit address. Figure 4 shows an example.

The table method reduces the number of steps and removes some of the more error-prone operations, making the new process more reliable.

20	04	60	06	A0	05	E0	07
21	84	61	86	A1	85	E1	87
22	44	62	46	A2	45	E2	47
23	C4	63	C6	A3	C5	E3	C7
24	24	64	26	A4	25	E4	27
25	A4	65	A6	A5	A5	E5	A7
26	64	66	66	A6	65	E6	67
27	E4	67	E6	A7	E5	E7	E7
28	14	68	16	A8	15	E8	17
29	94	69	96	A9	95	E9	97
2A	54	6A	56	AA	15	EA	57
2B	D4	6B	D6	AB	D5	EB	D7
2C	34	6C	36	AC	35	EC	37
2D	B4	6D	B6	AD	B5	ED	D7
2E	74	6E	76	AE	75	EE	77
2F	F4	6F	F6	AF	F5	EF	F7
30	0C	70	0E	B0	0D	F0	0F
31	8C	71	8E	B1	8D	F1	8F
32	4C	72	4E	B2	4D	F2	4F
33	CC	73	CE	B3	CD	F3	CF
34	2C	74	2E	B4	2D	F4	2F
35	AC	75	AE	B5	AD	F5	AF
36	6C	76	6E	B6	6D	F6	6F
37	EC	77	EE	B7	ED	F7	EF
38	1C	78	1E	B8	1D	F8	1F
39	9C	79	9E	B9	9D	F9	9F
3A	5C	7A	5E	BA	5D	FA	5F
3B	DC	7B	DE	BB	DD	FB	DF
3C	3C	7C	3E	BC	3D	FC	3F
3D	BC	7D	BE	BD	BD	FD	DF
3E	7C	7E	7E	BE	7D	FE	7F
3F	FC	7F	FE	BF	FD	FF	FF

N/c = Non-canonical; C = canonical

Figure 3 (continued): Conversion table

There are several variations of this table. We started with the above conversion table, which is a simple conversion chart of all 256 possible hexadecimal values. However, I preferred the table shown in Figure 5 because of its compact form. With this table, you use the first digit of the hex pair to locate the appropriate row, and the second digit to locate the proper column. The value located at the intersection of this row and column represents the converted number.

OUR FINAL SOLUTION

Although these table processes sufficed for a while, most of our network technicians eventually wanted a more automated process.

0200.ECA2.0811							(Original address – IEEE Illustrative form)
02	00	EC	A2	08	11		(Separate in pairs)
40	00	37	45	10	88		(Use the conversion table above)
4000.3745.1088							(Converted address – MAC bit order)

Figure 4: Example using conversion table

First character (row) **Second character (column)**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	80	40	C0	20	A0	60	E0	10	90	50	D0	30	B0	70	F0
1	08	88	48	C8	28	A8	68	E8	18	98	58	D8	38	B8	78	F8
2	04	84	44	C4	24	A4	64	E4	14	94	54	D4	34	B4	74	F4
3	0C	8C	4C	CC	2C	AC	6C	EC	1C	9C	5C	DC	3C	BC	7C	FC
4	02	82	42	C2	22	A2	62	E2	12	92	52	D2	32	B2	72	F2
5	0A	8A	4A	CA	2A	AA	6A	EA	1A	9A	5A	DA	3A	BA	7A	FA
6	06	86	46	C6	26	A6	66	E6	16	96	56	D6	36	B6	76	F6
7	0E	8E	4E	CE	2E	AE	6E	EE	1E	9E	5E	DE	3E	BE	7E	FE
8	01	81	41	C1	21	A1	61	E1	11	91	51	D1	31	B1	71	F1
9	09	89	49	C9	29	A9	69	E9	19	99	59	D9	39	B9	79	F9
A	05	85	45	C5	25	A5	65	E5	15	95	55	D5	35	B5	75	F5
B	0D	8D	4D	CD	2D	AD	6D	ED	1D	9D	5D	DD	3D	BD	7D	FD
C	03	83	43	C3	23	A3	63	E3	13	93	53	D3	33	B3	73	F3
D	0B	8B	4B	CB	2B	AB	6B	EB	1B	9B	5B	DB	3B	BB	7B	FB
E	07	87	47	C7	27	A7	67	E7	17	97	57	D7	37	B7	77	F7
F	0F	8F	4F	CF	2F	AF	6F	EF	1F	9F	5F	DF	3F	BF	7F	FF

Figure 5: Compact conversion table

Depending on where a particular problem occurred, they might not have access to the conversion table and might have to resort to a manual conversion. And when a technician was pressed for time, he was more likely to make an error, even using the table method.

Since the procedure for converting addresses is fairly simple, it ought to be relatively simple to provide an automated conversion tool. And a quick glance at Figure 5 suggests that a program using a translate table might be a perfect solution to this problem – after all, that's why translate tables are used in programs. I therefore started to code a translate table using the conversion tables shown above.

I thought that it would be an easy task to build the table and incorporate it into a program that provided the input and output logic. However, I soon noticed that some of the 256 possible values do not change when converted – in the above tables, check the converted values for 00, 66, 99, FF. I therefore began to think about the problem in terms of 16 hexadecimal values instead of 256, and constructed the chart shown in Figure 6.

The new chart made everything much clearer. Note that the values for 0, 6, 9, and F are represented the same even when reversed. Simply

Hexadecimal	Binary	Reversed binary	Reversed hexadecimal
0	0000	0000	0
1	0001	1000	8
2	0010	0100	4
3	0011	1100	C
4	0100	0010	2
5	0101	1010	A
6	0110	0110	6
7	0111	1110	E
8	1000	0001	1
9	1001	1001	9
A	1010	0101	5
B	1011	1101	D
C	1100	0011	3
D	1101	1011	B
E	1110	0111	7
F	1111	1111	F

Figure 6: Revised chart

stated, the problem of converting MAC addresses to and from canonical form has been reduced to translating 12 out of a possible 16 characters, and reversing the order of the hexadecimal digits.

This led me to consider a REXX solution for the problem. Most of our technicians have access to NetView. We also have MVS/TSO and VM/CMS environments and they both support REXX as well. Additionally, REXX has 'built-in' translate and reverse functions. Consequently, our final solution to the conversion problem is the following REXX EXEC.

```
/* REXX */
/*******************************************************************/
/*                                                                 */
/*    MEMBER-NAME: CFMAC                                           */
/*    Convert Non-canonical form MAC addresses into canonical form */
/*    or canonical into non-canonical form.                        */
/*******************************************************************/
trace off
arg mac
if mac = '' then signal help
lid = length(mac)
if lid ¬= 14 then signal endit
if substr(mac,5,1) ¬= '.' then signal endit
if substr(mac,10,1) ¬= '.' then signal endit
if lid = 14 then
  do
    m1 = substr(mac,1,4)
    m2 = substr(mac,6,4)
    m3 = substr(mac,11,4)
    mak = m1||m2||m3
  end
  cf = translate(mak,'84C2AE15D3B7','1234578ABCDE')

  cf1 = reverse(substr(cf,1,2))
  cf2 = reverse(substr(cf,3,2))
  cf3 = reverse(substr(cf,5,2))
  cf4 = reverse(substr(cf,7,2))
  cf5 = reverse(substr(cf,9,2))
  cf6 = reverse(substr(cf,11,2))
  cfm = cf1||cf2||.||cf3||cf4||.||cf5||cf6

say 'Translated form of 'mac' is 'cfm
exit
help:
```



```
say 'Translates between canonical and noncanonical form MAC addresses'
say 'CORRECT FORM: CFMAC <mac>'
say '              where <mac> is the MAC address (aaaa.bbbb.cccc)'
say 'DEFAULT:      None. Required.'
Exit
endit:
say 'SYNTAX ERROR: 'mac' IS INVALID (aaaa.bbbb.cccc)'
EXIT
```

After completing this REXX EXEC, I was surprised at how small it was. What's more, if we had wanted to be more spartan still, it could have been reduced to about 15 lines of code. However, we felt that it was beneficial to provide some error checking and help information. We also coded the REXX EXEC to require that the MAC addresses would be provided with a period '.' separating each four digits. We've grown accustomed to this convention, and it improves the readability of the addresses.

To install the REXX EXEC, place a copy in a dataset referenced by the DSICLD DDNAME for NetView or in a dataset referenced by the SYSEXEC DDNAME in your TSO log-on procedure. To invoke the EXEC, enter the following command in a NetView or TSO session:

```
CFMAC    4000.3745.2272
```

You should receive the following response:

```
Translated form of 4000.3745.2272 is 0200.ECA2.444E
```

Since the conversions to and from canonical form are inverse functions, if you invoke the REXX EXEC with the answer from the previous example as input, you will obtain the original MAC address from the first example. So, for

```
CFMAC    0200.ECA2.444E
```

you should receive the following response:

```
Translated form of 0200.ECA2.444E is 4000.3745.2272
```

Anthony Cieri
(USA)

Information point – reviews

The availability of IBM's mainframe manuals on the Internet is the topic this issue. The place to start is:

http://www.s390.ibm.com/library

For example, for the latest (eighth) edition of the *3270 Data Stream Programmer's Reference* (GA23-0059-07), you would:

- Click on the IBM *hardware* link.
- Click on the *Search Titles, names or doc numbers* link.
- Under *1 – Search Titles, names or doc numbers*, type '3270' in the box under *Find books with titles, names, or doc numbers containing:* and click the *Find* button.
- Click on the link for the single book title displayed, *3270 Data Stream Prgm. Reference*.

At this point, you'll see a BookManager-like user interface with the table of contents of the manual displayed, where each chapter and section is a link to that piece of the manual. You now have two choices:

- Read the manual on-line through your Web browser
- Download the manual to your workstation hard drive or LAN, and view it with BookManager.

NEW JAVA-BASED BOOKMANAGER

For as long as manuals have been available for viewing on the workstation, IBM has offered a free version of BookManager for this purpose. Until recently, it had the feel of being designed for Windows 3.1, but that has suddenly changed with the new IBM Software Reader, also known as Book Reader, and still free. It also comes with Bookshelf Organizer.

The only non-trivial downsides that I am aware of are:

- It runs in Java.

- It no longer fits on a diskette.

IBM Software Reader is a 4MB download from

http://booksrv2.raleigh.ibm.com/homepage/javaserv.html

But, before you do the download, be sure to install (at least) Version 1.3.0 of the Java Runtime Environment (JRE), a free 5MB download from

http://java.sun.com/j2se/1.3/jre/download-windows.html

Also, be sure to uninstall any previous versions of BookManager from your workstation.

TIPS ON USE

During the IBM Software Reader install, you will twice be asked to specify a directory: the first indicates where you will (or already have) stored BookManager (.boo) books, such as the one being downloaded in the next paragraph; and the second where any notes you create on the book will be stored.

On IBM's Web site, once you have the BookManager table of contents displayed for a manual you're interested in, you can download the entire manual by clicking the download icon on the top row of the Web page.

Bookmarking

Alternatively, you can bookmark the manual by specifying *Add to Favourites* in Internet Explorer while displaying the table of contents. For the 3270 manual, the URL saved is:

http://www.s390.ibm.com/bookmgr-cgi/bookmgr.exe/BOOKS/CN7P4000/CCONTENTS

though you would still get there if you do not specify the final '/CCONTENTS'.

You can also bookmark sections of the manual.

http://www.s390.ibm.com/bookmgr-cgi/bookmgr.exe/BOOKS/CN7P4000/C%2e0 gets you to Appendix C of the 3270 manual. You can add #TBLTBLUNIQ158 to that URL to get directly to Figure C-1, which shows the hexadecimal value to use for each possible sequence of bits 2-7 in a 3270 Field Attribute. I figured it out by looking at the HTML source while viewing Appendix C on-line:

<a name="TBLTBLUNIQ158">

but an easier way is to find a hypertext link that takes you directly to the table, such as the Table of Figures at the beginning of the manual.

ADOBE ACROBAT

Not all manuals are available in BookManager format; some are offered only in Adobe Acrobat format, also known as Portable Document Format or .pdf. The most popular, recently-published manuals are available in both formats.

With Acrobat documents, as with BookManager, you are given two choices:

- View the manual on-line, which can mean quite slow response time.
- Download the manual and view it directly from your hard disk or the LAN.

Unlike BookManager, whatever you choose, you must have the free Adobe Acrobat Reader installed on your workstation. The current version is available for download at

http://www.adobe.com/products/acrobat/readstep2.html

And it, too, no longer fits on a diskette, weighing in at over 5MB as a download.

BOOKMANAGER VERSUS ACROBAT

The choice between BookManager and Adobe Acrobat is mostly a matter of personal preference. Acrobat very accurately reproduces the paper manual on your screen. With BookManager, no attempt has been made to preserve the look and feel of the paper original, but efforts revolve around best use of the Web page.

Beyond personal preference, manuals in Adobe Acrobat format are more than twice the size, so require longer to download. Viewed on-line, BookManager provides better response time because much less is being loaded as you navigate.

Unfortunately, however, it's not as simple as finding the manual you want and choosing between Acrobat and BookManager format. For example, when you select *OS/390* from the System/390 Library home page, you must then choose between BookManager and Acrobat formats if you want to do a search.

The problem is that a few manuals are available only in one format. So, if you don't find a particular manual with one search, try searching for it again in the other format. In general, this 'look for both' approach is an important one to keep in mind.

Jon E Pearkins
(Canada)

March 1997 – December 2000 index

Items below are references to articles that have appeared in *TCP/SNA Update* since March 1997. References show the issue number followed by the page number(s). All these back-issues of *TCP/SNA Update* can be ordered from Xephon. See page 2 for details.

TCP/SNA news

Cisco has begun shipping three products designed to address the management needs of a mixed SNA/IP network:

- SNA View 3.0, which provides SNA/IP network troubleshooting and diagnostic capabilities via network and mainframe correlation.
- Maps 3.0, which provides dynamic views of both SNA and IP networks, allowing users to scale to manage larger SNA/IP network environments through improved host discovery and status propagation.
- The Internetwork Status Monitor for System/390, which provides management and visibility of Cisco devices from the mainframe.

For further information, contact:
Cisco, 5305 Gulf Drive, Suite 1, New Port Richey, FL 34652, USA.
Tel: (813) 817 0131.
URL: http://www.cisco.com.

* * *

IBM has announced its flagship zSeries 900 server replacement for the System/390. The z900 offers significant enhancements to TCP/IP communications through the use of high-speed interconnects (‘HiperSockets’), allowing TCP/IP traffic to travel between partitions with a single z900 at memory speed rather than network speed, creating a network in a box.

DB2 connectivity is improved with DB2 Server for VSE V7R1, by adding the capability to use DRDA with TCP/IP as well as SNA.

TCP/IP for VSE/ESA V1R4 adds new facilities required for e-business connectors, CICS Web Support, DB2 Server for VSE Version 7 Release 1, and VSE/POWER PNET. Also, the new TCP/IP for VSE/ESA includes accumulated service, functional and performance improvements, quality and diagnostics enhancements, and better documentation.

For further information, contact your local IBM representative or visit the Web site at http://www.s390.ibm.com.

* * *

Tivoli has announced its new NetView Performance Monitor for TCP/IP (NPM/IP) automated TCP/IP management software for monitoring the performance of OS/390 TCP/IP-based applications including FTP, TN3270, SAP, and new commerce applications.

IBM has also announced Version 5.0 of Tivoli Decision Support for OS/390, now offering support for TCP/IP for OS/390.

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.