45 TCP/SNA

March 2002

update

In this issue

TCP/SNA Update

Published by

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

North American office

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

Subscriptions and back-issues

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

Editorial panel

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

Editor

Fiona Hewitt

Disclaimer

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

Contributions

When Xephon is given copyright, articles published in *TCP/SNA Update* are paid for at £170 ($260) per 1000 words for original material. To find out more about contributing an article, please download a copy of our *Notes for Contributors* from http://www.xephon.com/index/nfc/css/nfc.

TCP/SNA Update on-line

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

Printed in England.

Converting TCP/IP addresses into 8-character terminal ID format

In an OS/390 TCP/IP (Communications Server for OS/390) environment, we're constantly dealing with TCP/IP addresses. These addresses are usually written in dotted-decimal notation, with four groups of decimal numbers separated by a period (.). In our current version of the TCP/IP architecture (Version 4), each group of decimal numbers can range from 1 to 255. Most TCP/IP commands require you to provide the address of the remote host where the server that you wish to access resides – for example, FTP, SNMP, or telnet. Coding software initialization files and configuring routers usually requires TCP/IP addresses to be provided in this same dotted-decimal notation.

Most TCP/IP application software uses a short-hand notation for storing TCP/IP addresses internally. This internal 32-bit representation is used to store the more familiar dotted-decimal form of the TCP/IP address. Many IBM-provided TCP/IP commands call the same C-function (inet_addr()) to perform this conversion. In fact, some mainframe security packages, which can be coded to restrict access based on the originating TCP/IP address, require that the TCP/IP addresses be provided in 32-bit format. Other mainframe TCP/IP applications issue error messages where the TCP/IP addresses are provided in the internal 32-bit format. Because of these inconsistencies, we need to be able to convert one format to the other.

The conversion is a relatively simple process. First, we start with a particular TCP/IP address, for example, 10.20.30.40. Next, we convert each decimal octet into hexadecimal, and finally remove the periods. The result, which is sometimes referred to as an 8-character terminal ID, is 0A141E28.

Starting with the internal 32-bit format, for example C1055609, we separate the 8-character terminal ID into pairs of hexadecimal digits:

C1 05 56 09

Next, we convert each hexadecimal pair into a decimal number. The largest hexadecimal number you can represent in two digits is FF or 255 in decimal:

193 05 86 09

Finally, we rewrite the address in dotted-decimal notation, omitting leading zeros where appropriate. The result is 193.5.86.9.

This article provides two sets of tools to perform this TCP/IP address conversion automatically, and a brief comparison of the methods employed in each. There are two Netview CLISTs, IPXTERM and TERMXIP, to perform the conversion to and from the 32-bit format respectively. These CLISTs are written in the standard NetView CLIST language. There are also two REXX EXECs that provide the same functions (named TERM2IP and IP2TERM). Note that, since REXX is an extremely powerful programming language, the two REXX conversion EXECs are a lot smaller than their NetView CLIST equivalents.

TERM2IP AND IP2TERM REXX EXECS

TERM2IP

First, let's consider the TERM2IP REXX EXEC, the smallest of the group. This EXEC takes advantage of two built-in REXX functions: the SUBSTR function and the 'convert hexadecimal to decimal' (X2D) function. These allow the TERM2IP EXEC to convert 8-character terminal IDs into dotted-decimal TCP/IP addresses in approximately 25 lines of REXX code (excluding comments). The TERM2IP EXEC performs the conversion in the same way as outlined above, separating the 8-character terminal ID into four pairs of hexadecimal numbers, converting each hexadecimal number into decimal, and then rewriting the results in dotted-decimal notation.

TERM2IP REXX EXEC

```
/* REXX */
/*********************************************************************/
/* START OF SPECIFICATIONS                                           */
/*                                                                   */
/*    MEMBER-NAME: Term2IP                                           */
/*    Convert hashed termid into tcp/ip address                      */
/*                                                                   */
/*********************************************************************/
trace off
```

```
arg trm
if trm = '' then signal help
lid = length(trm)
if lid <> 8 then signal endit
startc:
 q1 = SUBSTR(trm,1,2)
 q2 = SUBSTR(trm,3,2)
 q3 = SUBSTR(trm,5,2)
 q4 = SUBSTR(trm,7,2)
 o1 = x2d(q1)
 o2 = x2d(q2)
 o3 = x2d(q3)
 o4 = x2d(q4)
ipa = o1||.||o2||.||o3||.||o4
say 'Termid 'trm' has TCP/IP Address of 'ipa
exit
help:
say 'Rexx exec to convert TERMINAL IDs to TCP/IP addresses'
say 'CORRECT FORM: TERM2IP <termid>'
say '              where termid is the 8 character Terminal ID'
say 'DEFAULT:      None. Required.'
exit
endit:
say 'SYNTAX ERROR: Terminal ID 'trm' is INVALID'
exit
```

IP2TERM

The second REXX EXEC, IP2TERM, provides a utility for converting dotted-decimal TCP/IP addresses into 8-character terminal ID format. This REXX EXEC requires that the input be provided in dotted-decimal format – if the octets of the TCP/IP address are *not* separated with periods (.),the EXEC will end with an error.

Once again, we take advantage of some built-in REXX functions: this time, the PARSE function and the 'convert decimal to hexadecimal' (D2X) function. The PARSE function is used to check for the periods and to separate the supplied TCP/IP address into four octets. Next, we convert each decimal octet into a hexadecimal number. To ensure that the resulting terminal ID is precisely 8 characters in length, we have to convert each decimal octet into a two-digit hexadecimal number. If the result of the D2X function is only one character in length, we'll add a leading zero (0). This will occur if any octet in our TCP/IP address is less than 16. Finally, we concatenate the results to make an 8-character terminal ID.

IP2TERM REXX EXEC

```
/* REXX */
/*********************************************************************/
/* START OF SPECIFICATIONS                                           */
/*                                                                   */
/*    MEMBER-NAME: IP2TERM                                           */
/*    Convert TCP/IP addresses into hashed terminal IDs.             */
/*                                                                   */
/*********************************************************************/
trace off
arg trm
if trm = '' then signal help
lid = length(trm)
if lid = GT 15 then signal endit
if lid = LT 7 then signal endit

startc:
 parse var trm q1 "." q2 "." q3 "." q4
/*
 say '1st octet is 'q1
 say '2nd octet is 'q2
 say '3rd octet is 'q3
 say '4th octet is 'q4
*/
 o1 = d2x(q1)
 o2 = d2x(q2)
 o3 = d2x(q3)
 o4 = d2x(q4)
if length(o1) = 1 then o1 = '0'||o1
if length(o2) = 1 then o2 = '0'||o2
if length(o3) = 1 then o3 = '0'||o3
if length(o4) = 1 then o4 = '0'||o4
ipa = o1||o2||o3||o4
say 'TCP/IP address 'trm ' converts to a terminal ID of 'ipa
exit
help:
say 'Rexx exec to convert TCP/IP addresses to 8 character Terminal IDs'
say 'CORRECT FORM: IP2TERM <address>'
say '              where address is the TCP/IP address in dotted '
say '              decimal format (www.xxx.yyy.zzz) to be converted'
say 'DEFAULT:      None. Required.'
exit
endit:
say 'SYNTAX ERROR: 'trm' IS INVALID'
exit
```

Note that you can use these REXX EXECs to validate each other. If you start with a particular TCP/IP address and use the IP2TERM EXEC to obtain an 8-character terminal ID, you can check your results

by using the TERM2IP EXEC on the resulting 8-character terminal ID. You should arrive back at the original TCP/IP address.

TERMXIP AND IPXTERM CLISTS

Next, let's look at the Netview CLIST language tools for converting TCP/IP addresses. I've provided these CLISTs primarily for illustrative purposes, although they'll be useful for any NetView environment that doesn't have REXX.

Although not quite as succinct, the TERMXIP and IPXTERM CLISTs attempt to provide the same functionality as their REXX counterparts. Because the NetView CLIST language doesn't have many of the built-in functions that were used in the REXX EXECs, we have to alter our approach to the conversion process.

TERMXIP CLIST

Let's consider the TERMXIP CLIST first. Since the NetView CLIST language doesn't provide a function for converting between decimal and hexadecimal, we'll have some additional coding to do in order to perform the same tasks. However, we do have a SUBSTR function in the NetView CLIST language, so TERMXIP uses this function to parse the 8-character terminal ID into 8 individual hexadecimal digits. Next, in a rather 'brute force' method, the CLIST checks each digit for the hexadecimal values A through F. If it finds such a value, it converts it into its decimal equivalent. At this point, we have 8 decimal numbers that correspond to the 8 hexadecimal characters in the original terminal ID. For example, if we started with the 8-character terminal ID from our previous example, C1055609, we would now have the following series of decimal numbers:

12 1 0 5 5 6 0 9

Together, each pair of these numbers ultimately combines to form one octet of our TCP/IP address. To complete the conversion, TERMXIP multiplies the first number in each pair by sixteen. Since there's no native multiplication function in NetView, TERMXIP uses a loop to double the number in four consecutive instructions. This is only necessary for the first, third, fifth, and seventh numbers. Now, our series of numbers would be:

192 1 0 5 80 6 0 9

Next, we add the pairs of number together to yield:

193 05 86 09

Finally, we write the four octets in dotted-decimal notation. The resulting TCP/IP address is 193.5.86.9. Once again, we omit leading zeros where appropriate.

TERMXIP NetView CLIST

```
&CONTROL ERR
***
* CONVERT HASHED TERMID INTO TCP/IP ADDRESS
***
&IF .&1 EQ . &THEN &GOTO -HELP
&TRM = &1
&LID = &LENGTH &TRM
&IF &LID LT 8 &THEN &GOTO -HELP
&Q1 = &SUBSTR &TRM 1 1
&Q2 = &SUBSTR &TRM 2 1
&Q3 = &SUBSTR &TRM 3 1
&Q4 = &SUBSTR &TRM 4 1
&Q5 = &SUBSTR &TRM 5 1
&Q6 = &SUBSTR &TRM 6 1
&Q7 = &SUBSTR &TRM 7 1
&Q8 = &SUBSTR &TRM 8 1
&I = 1
-LOOP
&IF &Q&I EQ 'A' &THEN &Q&I = 1Ø
&IF &Q&I EQ 'B' &THEN &Q&I = 11
&IF &Q&I EQ 'C' &THEN &Q&I = 12
&IF &Q&I EQ 'D' &THEN &Q&I = 13
&IF &Q&I EQ 'E' &THEN &Q&I = 14
&IF &Q&I EQ 'F' &THEN &Q&I = 15
&I = &I + 1
&IF &I LE 8 &THEN &GOTO -LOOP
&I = 1
-LOOP2
&Q&I = &Q&I + &Q&I
&Q&I = &Q&I + &Q&I
&Q&I = &Q&I + &Q&I
&Q&I = &Q&I + &Q&I
&I = &I + 2
&IF &I LE 7 &THEN &GOTO -LOOP2
&IP1 = &Q1 + &Q2
&IP2 = &Q3 + &Q4
&IP3 = &Q5 + &Q6
&IP4 = &Q7 + &Q8
```

```
&IP1 = &CONCAT &IP1 '.'
&IP2 = &CONCAT &IP1 &IP2
&IP2 = &CONCAT &IP2 '.'
&IP3 = &CONCAT &IP2 &IP3
&IP3 = &CONCAT &IP3 '.'
&IP4 = &CONCAT &IP3 &IP4
&WRITE Termid &TRM has TCP/IP Address of &IP4
&EXIT
-HELP
&WRITE SYNTAX ERROR: "termid" IS INVALID
&BEGWRITE -END
VTAM DISPLAY TERMINALS COMMAND.
CORRECT FORM: TERM2IP <termid> where termid is the 8 character TERMID
DEFAULT:      None. Required.
-END
&EXIT
-CMD
TERM2IP <termid>  where TERMID is the 8 character TERMID
&EXIT
```

IPXTERM CLIST

The last of our conversion tools is the NetView CLIST IPXTERM. This CLIST is the largest example presented. In fact, the subroutine that converts from decimal to hexadecimal comprises more than half of the code in the IPXTERM CLIST. Once again, this is because the NetView CLIST language doesn't have many of the arithmetic and conversion functions that are present in REXX and we therefore need to use some more creative techniques to solve the problem.

The IPXTERM NetView CLIST uses a very similar parsing function to that used in the IP2TERM REXX EXEC. In NetView CLIST language, this function is called PARSEL2R (Parse Left to Right). It's used to check that the input TCP/IP address is in dotted-decimal notation and to separate the four individual octets. If the TCP/IP address is not provided in dotted-decimal format, the CLIST will end with an error. Next, the CLIST checks to see that none of the individual decimal octets exceeds 255. If any octet is greater than 255, again the CLIST will end with an error.

Once the TCP/IP address is parsed, the IPXTERM CLIST branches to a subroutine to convert the decimal values into their hexadecimal equivalents. This subroutine attempts to provide the same functionality as the REXX D2X function. It uses a series of comparisons and

variable substitutions to determine the largest multiple of 16 that can be subtracted from the octet. In doing so, it calculates the hexadecimal value of the decimal octets. It is coded to always return a two-digit hexadecimal value for every decimal octet that is passed to it. Each of the four octets is processed in a consecutive branch to the subroutine. It's a very cumbersome and unsophisticated process, but it does get the job done. Each time the IPXTERM CLIST returns from the subroutine, it concatenates the two hexadecimal digits to a terminal ID variable. After the fourth octet is processed, the variable contains a complete 8-character terminal ID.

IPXTERM NetView CLIST

```
&CONTROL ERR
***
* CONVERT TCP/IP ADDRESSES INTO 8 CHARACTER TERMINAL IDS
***
&IF .&1 EQ . &THEN &GOTO -HELP
&IPA = &1
&DEB = &2
&LID = &LENGTH &IPA
&IF &LID LT 7 &THEN &GOTO -HELP
&IF &LID GT 15 &THEN &GOTO -HELP
PARSEL2R IPA Q1 /./ Q2 /./ Q3 /./ Q4
&IF &DEB NE 'DEBUG' &THEN &GOTO -SKIP1
&WRITE First  octet is &Q1
&WRITE Second octet is &Q2
&WRITE Third  octet is &Q3
&WRITE Fourth octet is &Q4
-SKIP1
&IF &Q1 GE 256 &THEN &GOTO -HELP
&IF &Q2 GE 256 &THEN &GOTO -HELP
&IF &Q3 GE 256 &THEN &GOTO -HELP
&IF &Q4 GE 256 &THEN &GOTO -HELP
-OCTET1
&DEC = &Q1
&RETLBL = -OCTET1R
&GOTO -CNVD2X
-OCTET1R
&TERM = &CONCAT &TERM &HEX
-OCTET2
&DEC = &Q2
&RETLBL = -OCTET2R
&GOTO -CNVD2X
-OCTET2R
&TERM = &CONCAT &TERM &HEX
```



```
-OCTET3
&DEC = &Q3
&RETLBL = -OCTET3R
&GOTO -CNVD2X
-OCTET3R
&TERM = &CONCAT &TERM &HEX
-OCTET4
&DEC = &Q4
&RETLBL = -OCTET4R
&GOTO -CNVD2X
-OCTET4R
&TERM = &CONCAT &TERM &HEX
&WRITE TCPIP address &IPA converts to a Terminal ID of &TERM
&EXIT
-HELP
&WRITE SYNTAX ERROR: "address" IS INVALID
&BEGWRITE -END
Netview TCP/IP address to Terminal ID converter
CORRECT FORM: IP2TERM <address> where address is the TCP/IP Address
              in dotted decimal format (www.xxx.yyy.zzz). Each octet
              must be between Ø and 255.
DEFAULT:      None. Required.
-END
&EXIT
***
* Subroutine to Convert Decimal values into Hexadecimal values. The
* variable &DEC is passed to this subroutine along with a return label.
* The value of &HEX is calculated and then we go to the return label.
***
-CNVD2X
&X1Ø = A
&X11 = B
&X12 = C
&X13 = D
&X14 = E
&X15 = F
&IF &DEC GT 24Ø &THEN &GOTO -HEXF
&IF &DEC GT 224 &THEN &GOTO -HEXE
&IF &DEC GT 2Ø8 &THEN &GOTO -HEXD
&IF &DEC GT 192 &THEN &GOTO -HEXC
&IF &DEC GT 176 &THEN &GOTO -HEXB
&IF &DEC GT 16Ø &THEN &GOTO -HEXA
&IF &DEC GT 144 &THEN &GOTO -HEX9
&IF &DEC GT 128 &THEN &GOTO -HEX8
&IF &DEC GT 112 &THEN &GOTO -HEX7
&IF &DEC GT 96 &THEN &GOTO -HEX6
&IF &DEC GT 8Ø &THEN &GOTO -HEX5
&IF &DEC GT 64 &THEN &GOTO -HEX4
&IF &DEC GT 48 &THEN &GOTO -HEX3
&IF &DEC GT 32 &THEN &GOTO -HEX2
```

```
&IF &DEC GE 16 &THEN &GOTO -HEX1
&GOTO -HEXØ
-HEXF
&X1 = F
&DEC2 = &DEC - 24Ø
&GOTO -DIGIT2
-HEXE
&X1 = E
&DEC2 = &DEC - 224
&GOTO -DIGIT2
-HEXD
&X1 = D
&DEC2 = &DEC - 2Ø8
&GOTO -DIGIT2
-HEXC
&X1 = C
&DEC2 = &DEC - 192
&GOTO -DIGIT2
-HEXB
&X1 = B
&DEC2 = &DEC - 176
&GOTO -DIGIT2
-HEXA
&X1 = A
&DEC2 = &DEC - 16Ø
&GOTO -DIGIT2
-HEX9
&X1 = 9
&DEC2 = &DEC - 144
&GOTO -DIGIT2
-HEX8
&X1 = 8
&DEC2 = &DEC - 128
&GOTO -DIGIT2
-HEX7
&X1 = 7
&DEC2 = &DEC - 112
&GOTO -DIGIT2
-HEX6
&X1 = 6
&DEC2 = &DEC - 96
&GOTO -DIGIT2
-HEX5
&X1 = 5
&DEC2 = &DEC - 8Ø
&GOTO -DIGIT2
-HEX4
&X1 = 4
&DEC2 = &DEC - 64
&GOTO -DIGIT2
```

```
-HEX3
&X1 = 3
&DEC2 = &DEC - 48
&GOTO -DIGIT2
-HEX2
&X1 = 2
&DEC2 = &DEC - 32
&GOTO -DIGIT2
-HEX1
&X1 = 1
&DEC2 = &DEC - 16
&GOTO -DIGIT2
-HEXØ
&X1 = Ø
&DEC2 = &DEC
-DIGIT2
&IF &DEC2 LE 9 &THEN &X2 = &DEC2
&IF &DEC2 GE 1Ø &THEN &X2 = &X&DEC2
&HEX = &CONCAT &X1 &X2
&IF &DEB NE 'DEBUG' &THEN &GOTO -SKIP2
&WRITE Hexadecimal value of &DEC is &HEX
-SKIP2
&GOTO &RETLBL
```

Although these CLISTs are not very elegant coding examples, they do provide the same functions as their REXX analogues. Either or both sets can be used to provide a quick conversion between TCP/IP addresses and 8-character terminal IDs.

Anthony Cieri
(USA)

TCP/SNA Update on the Web

As a free service to subscribers and to remove the need to rekey the scripts, code from individual articles of *TCP/SNA Update* and complete issues in PDF format can be accessed on our Web site at:

http://www.xephon.com/tcpsna

You will be asked to enter a word from the printed issue.

TCP/IP response time analysis

In today's enterprise, application development has changed from single-tier/single-user applications to multi-tier applications, which exploit the best features of processing platforms ranging from OS/390 servers to PCs. The responsibility for application design and development has moved from the glasshouse out into the business units and the hands of business analysts who know what functionality they need. Applications are being created that expand the concept that "the network is the computer, and the computer is the network".

Not only are hardware platforms and business processes evolving, but change is also being driven at the pace of business competition. Figure 1 outlines the adoption rate of new application and business processes as corporations have moved from mainframe to Internet/intranet-based IT models. It shows clearly that the life-cycle of new applications, from conception, through use, to replacement is becoming shorter as the market becomes more competitive.

However, as Figure 2 shows, despite application design and implementation being moved out into the departments that have the need for business intelligence, the expectation of universal availability continues to grow.

Hardware and software manufacturers are designing systems that allow continuous availability. But at the same time as hardware is becoming more resilient, application design and development are being pushed into the hands of users who understand business

	1960s-'70s	**'70s-'80s**	**'80s-'90s**	**'90s-2000+**
Applications	Operational	Departmental	Cross-enterprise	Business-to-business and consumer
Number of users	Small	Small	Moderate	Consumers, all workers
Adoption/ effective life	10+ years	7+ years	5+ years	3+ years

Figure 1: Adoption rate of application and business processes

Expectations	Class-‘9s’	Outage	Examples
Always available	99.999%	5 min/year	OS/390 or z/OS parallel
Fault tolerant	99.99%	53 min/year	ES/9000 XRF
High availability	99.9%	8.8 hrs/year	Stratus ES/9000
General purpose	99%	88 hrs/year	HA Power Parallel
Campus LANs	90%	876 hrs/year	Alternative mainframes

Figure 2: Expectation of universal availability

practices better than application development standards. And however great the hardware, it can’t mask mistakes in application design and implementation. But effective system and network support can mitigate the impact of poor design choices.

THE PROBLEM

So, how do we make sure that end-user applications run with a minimum of resources and maximum effectiveness? The answer is to concentrate on the ‘network’ portion of the enterprise-computing environment.

THE ASSUMPTIONS

When OS/390 servers are combined with Unix and/or OS/400 mid-range systems, all being front-ended by Windows workstations, the network will be generally be structured using TCP/IP functionality. Applications designed on departmental workstations will need access to corporate data servers in the same, or different, departments. Business intelligence can be created either by discrete applications accessing source data directly, or through new code front-ending legacy applications and post-processing the resulting transaction data. New applications accessing source data directly will create more network traffic, as the required data is moved from the servers to the workstations for processing. However, programs accessing legacy transactions often have a higher run rate.

So, the first question that has to be asked of new applications is: “to post-process, or not to post-process?” Once the application data

requirements have been analysed, we can start to determine the impact of data transfer on the network resources between the user and the source data server.

In today's enterprise, the end user normally links his or her workstation to the corporation Intranet. LAN/WAN access is relatively simple, and allows a great deal of flexibility. The Intranet will have gateways that allow access to both external customers (Internet) and the corporate back-end processors (mainframe/mid-range). As corporate mergers continue, and as enterprises expand, the corporate Intranet grows increasingly more dispersed and complex. End users are no longer solely clustered in banks of cubicles, on the floors above and below the data centre. Instead, today's market allows users to work from home, or access corporate resources from remote locations, using the Internet. The challenge of managing the network, when you haven't been able to control either the application or the design of the network between data server and end user, can quickly become a nightmare of someone else's making.

NETWORK ISSUES

Network problems create unique 'opportunities' for technical staff, including:

- The detection of network problems before application impact is recognized.
- The acquisition of tools to monitor the entire TCP/IP network and its resources.
- The detection of resource shortages that create network bottlenecks.
- Creating strategies that allow the monitoring of network availability by segment.
- The determination of end-user response times at the application workstation.
- Combining existing tools on disparate platforms, with disparate operating systems, into a seamless network monitoring and management methodology.

- Defining Service Level Agreements, and managing service delivery to the end user.

Add to this set of challenges both the existing problem of dealing with the end-user community and the requirement to support and exploit new technology and business applications, and you have a recipe for problems that will exceed the scope of any one platform or department.

When things go wrong, you'll need to understand the visibility of network problems in order to prioritize the resolution process. One type of 'network problem' is the impact on application response times when new workloads are added to remote nodes without appropriate planning. As bandwidth requirements increase, the effect will be felt all along the path from the source workstation, through to each data server involved. An understanding of the path involved in transferring data is critical to resolving network bottlenecks.

POLITICAL ISSUES

As application response times slow, users become increasingly vocal about the impact of the network on their productivity. The complaints don't usually come from the culprits, but from everyone else along the affected path – a user spawning a new application, requesting large volumes of data for business analysis, will normally be very tolerant of slowed application response times. What that user doesn't understand is the impact the new application has on the rest of the workstations on the network segments that transfer server data to analyse.

If network response times remain slow, users will find other things to do with their idle time, and overall productivity will drop. Unfortunately, the human response to application performance degradation is a loss of concentration on the task at hand. Network bottlenecks will not normally cause a complete failure in an application; instead, they slow the business process to the point where people will spend more time trying to look busy than being busy. If your organization post-processes response time data, Service Level Agreements will start to be violated as network bandwidth becomes insufficient to handle the expected application data traffic.

New applications and new middleware solutions are being designed to take advantage of the ability to access the network directly. CICS

socket interfaces allow applications to connect to TCP/IP and enable the processing of Web-based transactions without the overhead of subsystem-controlled network service calls. This type of processing allows business planners to Web-enable new and legacy applications without moving to new application design platforms, and without the need to re-educate the existing corporate application programming staff. Middleware products such as MQSeries can use TCP/IP, which allows the incorporation of flexibility into application data transport design that would otherwise be unavailable using native network resources.

RESPONDING TO NETWORK PROBLEMS

As the new network structure comes into being, old methods of performance monitoring become less and less effective at resolving problems. It's fairly simple to detect a problem with response times within a CICS transaction: normal monitoring methods are quick to point out that a transaction is in a Socket Receive Wait; further investigation may show that the wait is related to a specific IP address, and the resource can easily be identified.

But once we know what server we're communicating with, and whether it responds to a PING, how do we tell where the bottleneck resides? To begin the diagnosis of the problem, we need to understand where we're coming from. From the perspective of a workstation-based application, we need to know several things:

- What is our port into the network (workstation IP address)?
- Within the network, how large is our local subnet?
- At what point do we exit the local network and enter into the realm of the Internet or another network within the enterprise?
- What is the bandwidth of any given section of our network path between the application front-end and the back-end servers?
- What is the current state of our network interface?

THE WORKSTATION NETWORK CONNECTION

We can start our investigation of the network path from the workstation,

or the processor that's running the user interface, to the business application by using the IPCONFIG command:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\>ipconfig
Windows 2000 IP Configuration
Ethernet adapter (CE5D8121-99AC-41D7-9907-AC87AD6AD6A5F6E):
      Connection-specific DNS Suffix   .  : landmark.com
      IP Address . . . . . . . . . . . . .      :
208.203.105.202
      Subnet Mask . . . . . . . . . . . . .   : 255.255.255.0
      Default Gateway . . . . . . . . . . .   :
Ethernet adapter Local Area Connection:
      Media State . . . . . . . . . . . . .   : Cable
Disconnected
PPP adapter Pipex Amsterdam Nokia:
      Connection-specific DNS Suffix   .  :
      IP Address . . . . . . . . . . . . .      :
213.116.100.20
      Subnet Mask . . . . . . . . . . . . .   : 255.255.255.255
      Default Gateway . . . . . . . . . . .   : 213.116.100.20
```

The LAN adapter that connects the machine to the network is detailed in the first section of this code. In this case, our workstation is connected to the target system, using VPN, via an Ethernet adapter with a specified IP address of 208.203.105.202.

The segment of the LAN that we're on can have up to 254 (2^8 - 2) different devices on it. This can be determined by examining the Subnet Mask value of 255.255.255.0. The 0 in the last place determines that up to 254 different addresses can be connected to the segment of the network that we have access to. The values of those IP addresses range from 208.203.105.1 to 208.203.105.254. The values of 0 and 255 are reserved and cannot be assigned.

Once we know the design of the target network, we then need to know how we get from our local segment to the target network segment. In this case, our workstation is disconnected from the local Ethernet segment, but we still have access to the outside world through our PPP (Point to Point Protocol) connection using a dial-up link.

The PPP connection allows the workstation to access network services in the same way that a local Token Ring or Ethernet connection would. As indicated above, we have an assigned IP address within the dial-

up service (213.116.100.20), along with a connection to the service provider's internal LAN. Notice that in this case the possible number of devices we can attach to the LAN segment that we've been assigned is only one – the Subnet Mask of 255.255.255.255 indicates that all addresses are reserved outside of our single connection.

THE HOST NETWORK CONNECTION

Now that we know where we're coming from, we need to know where we are attached to the system that will process our application requests. We can discover this using any of a number of TCP/IP monitoring products to view active network connections.

```
**Jobname: NASTCP  ******** Connection Summary *********Date: 12/Ø3/Ø1**
* Host ID: MVSSYSE                                       Time:  4:49:32 *
* Command: ___________________________________________ Cycle: MMSS    *
* LMRK118Ø8I - DATA HAS BEEN REFRESHED                 _Total _Right 1 *
* Deltas are currently displayed                                       *
* Interval Duration  : 6.13                                            *
* Displayed: 1 to 12 of 18    Sort Column#: Ø2 (1-9)    Order: D (A/D) *
* Valid line commands: P(Ping), T(Tracert), M(DMTU), D(Drop), N(Name-  *
*                                                          lookup)     *
*    Remote          Remote Local Conn      Conn     Task   Bytes Bytes *
* C  IP Address      Port   Port  State     ID       Name   In    Out   *
*    *               *      *     *         *        *                *
* _  213.116.1ØØ.2Ø  1Ø57 telnet Establish ØØØØØCF8 TCPIP   673   2548 *
* _  127.Ø.Ø.1       1Ø26   1Ø25 Establish ØØØØØØ13 TCPIP     Ø      Ø *
* _  127.Ø.Ø.1       1Ø25   1Ø26 Establish ØØØØØØ11 TCPIP     Ø      Ø *
* _  Ø.Ø.Ø.Ø            * telnet Listen    ØØØØØØ15 TCPIP     Ø      Ø *
* _  Ø.Ø.Ø.Ø            *    ftp Listen    ØØØØØØ47 FTPD1     Ø      Ø *
* _  Ø.Ø.Ø.Ø            *   9Ø21 Listen    ØØØØØØ49 FTPD5     Ø      Ø *
* _  Ø.Ø.Ø.Ø            *   1Ø27 Listen    ØØØØØØ1D SNMPQ     Ø      Ø *
* _  Ø.Ø.Ø.Ø            * sunrpc Listen    ØØØØØØ1E PORTM     Ø      Ø *
* _  Ø.Ø.Ø.Ø            *   7233 Listen    ØØØØØØ1Ø NASHUBP   Ø      Ø *
* _  Ø.Ø.Ø.Ø            *   1Ø28 Listen    ØØØØØØ3D OSNMPD    Ø      Ø *
* _  Ø.Ø.Ø.Ø            *   1Ø25 Listen    ØØØØØØØC TCPIP     Ø      Ø *
* _  Ø.Ø.Ø.Ø            *  1ØØØ7 Listen    ØØØØØØ4F BPXOINI   Ø      Ø *
** Help Information = PF1*******NASTCP  **** PF Key Assignments = PA1***
```

Here, we can see our workstation listed as the remote connection of 213.116.100.20 with a remote port of 1057 (see above for the IP address of the user workstation). This is hooked to the telnet port on the host server. To get more information about our connection, we can select the TCP/IP entry and display the details available:

```
**Jobname: NASTCP  ******* TCP Connection Detail *******Date: 12/Ø3/Ø1**
* Host ID: MVSSYSE                                      Time:  4:52:ØØ *
* Command: _____________________________________________ Cycle: MMSS    *
*                                                          _Next _Prev *
* Deltas are currently displayed          _Total _Ping _Tracert _DMTU *
* Interval Duration  : 6.13               _NameL _PkTrc _SkTrc  _Drop *
*  ----------------- TCP Connection Identification -----------------  *
*  Remote IP Address: 213.116.1ØØ.2Ø Remote Port      :  1Ø57          *
*  Local IP Address : 172.22.9.59   Local Port        :    23 (telnet) *
*  Connection Status: Established   Connection ID     : ØØØØØCF8       *
*  Task Name        : TCPIP         Target Application: LSCTPX2        *
*  LU Name          : TCPE29Ø1      Log mode           : SNX32ØØ2       *
*  Time since last act:     Ø.Ø2ØØ  Client UserID      :                *
*  ----------------- Connection Characteristics --------------------  *
*  Connection Options:
*
*  Send Max seg size:     146Ø      Opt Max seg size  :        Ø       *
*  Socket options   :       CØ      TCP select cond   :       ØØ       *
*  ------------ ------ Connection Activity ------------------------  *
*  Bytes in         :      673      Bytes out         :     2548       *
*  Packets in       :        5      Packets out       :        5       *
*  Avg bytes/pkt in :      134      Avg bytes/pkt out :      5Ø9       *
*  Retransmissions  :        1      Retransmission cnt:        1       *
*  Round-trip time  :      796      Round-trip variance:       92      *
** Help Information = PF1*******NASTCP  **** PF Key Assignments = PA1***
```

Our session is currently established, and the application that's being run is TPX. We're moving bytes both into and out of the host system, and are dividing those bytes into a set number of packets on the connection.

For the currently established session, there has been only one case where an error occurred that required a retransmission of the session data. Finally, our session is running with an average round-trip time of 796 milliseconds. Note that while it's useful to measure the round-trip time in order to determine Service Level Agreement compliance, it presents a problem as a practical measurement of network performance.

Users are willing to have response times ranging from sub-second to sub-hour, depending on what they're trying to accomplish. Most users understand that if they request large quantities of data for a particular business function, they will have to wait for a response from the server before they can continue. With workstations allowing users to run multiple applications during a single session, users are no longer

constrained by a terminal window being held waiting for data to arrive. If a delay occurs, users will switch sessions and then periodically check back to see whether they can proceed or not.

Users will normally complain when response times for data retrieval are inconsistent. If they run an application and each enter requires 10 seconds to complete, they soon learn to accept the 10 second delay. However, if the first enter takes 10 seconds, and the next takes 2 seconds, the next 7, the next 1, and so on, you can be sure that you'll get a phone call telling you that response times are slow, even though the response times on average are better.

To determine whether our session is 'consistent', we need to examine the field labelled 'round-trip variance'. Typically, we like to see a round-trip variance of less than 10% of the base round-trip time. As the round-trip variance increases, the user's perception of consistent response times will suffer, and user complaints will rise.

THE NETWORK PATH

Once you've determined the actual session response time, you can turn your efforts to optimizing the response time. To do this, you need to know how you're getting from the user workstation to the server host system. With TCP/IP connections, the path involved between nodes may not be immediately obvious, because of intranet or Internet routing requirements.

Note that from this point on, we're attempting to establish consistent response times, not necessarily optimal ones.

First, you need to understand the route taken when moving data from the source workstation to the target server system. The TraceRoute utility allows you to view the choices the network will make when determining your path:

```
**Jobname: NASTCP  ********** Tracert Utility **********Date: 12/Ø3/Ø1**
* Host ID: MVSSYSE                                       Time:  4:55:28 *
* Command: ____________________________________________________________ *
* TTCP52120I - TRACERT request completed                                *
*                                                                       *
*  Target Host Name :                                                   *
*  Target IP Address: 213.116.1ØØ.2Ø  Resolve Addrs to Host Names?: Y   *
*  Timeout (Seconds): Ø5              Maximum Hops      : 3Ø            *
```

```
*  Time-to-Live(TTL): 255                                                *
*  Interface Used  : 172.22.9.59  _List Link Name:OSALINK2 Primary?: Y *
*                                                                        *
*  Results           : Complete                                          *
* Hop Status   IP Address     Host name                  Trip Time (ms)*
* 1 Successful 172.22.254.254  gw-nb3-2-tokenring.landmar    6    6  6 *
* 2 Successful 208.203.105.1   fw.landmark.com               6    0  7 *
* 3 Successful 208.203.104.1   fw-cisco.landmark.com         7    7  7 *
* 4 Successful 157.130.39.133  510.Hssi3-0-0.GW3.TCO1.ALT   10   10 10 *
* 5 Successful 152.63.34.122   158.at-1-0-0.XR2.TCO1.ALTE   10   11 11 *
* 6 Successful 152.63.34.38    192.at-1-1-0.TR2.DCA6.ALTE   12   13 12 *
* 7 Successful 152.63.9.217    0.so-6-0-0.IR2.DCA6.ALTER.   12   12 12 *
* 8 Successful 146.188.13.46   so-1-0-0.IR2.DCA4.Alter.Ne   12   12 13 *
* 9 Successful 146.188.3.106   so-1-0-0.TR2.AMS2.Alter.Ne   92   92 91 *
*10 Successful 146.188.8.81    so-6-0-0.xr1.ams6.nl.uu.ne   92   92 92 *
*11 Successful 152.63.9.253    0.so-6-0-0.IR1.NYC9.ALTER.   18   17 17 *
*12 Successful 152.63.23.58    so-0-0-0.IR1.NYC12.ALTER.N   16   16 16 *
*13 Successful 146.188.3.101   so-0-0-0.TR1.AMS2.Alter.Ne   92   91 91 *
*14 Successful 146.188.8.77    so-5-0-0.xr1.ams6.nl.uu.ne   91   91 91 *
*15 Successful 212.136.184.133 195.at-1-0-0.cr1.rtm1.nl.u   93   93 93 *
*16 Successful 212.136.177.106 311.atm3-0.dr1.rtm1.nl.uu.   93   93 92 *
*17 Successful 213.116.1.3     tnt3.rtm1.nl.uu.net          95   96 96 *
*18 Successful 213.116.100.20 1Cust20.tnt3.rtm1.nl.uu.ne 3502 2618 3752*
*                                                                       *
** Help Information = PF1*******NASTCP  **** PF Key Assignments = PA1***
```

If you review the path from the server's first external router (172.22.254.254) to the user workstation (213.116.100.20), you can determine the devices that are involved in exchanging network packets. The first performance question to be resolved is: "how many hops has the message taken in transit?"

In the code above, the hop count is 18. If the application is going to provide consistent application response times, the hop count should always be as close to 18 as possible. In order to detect application slowdown and mitigate the impact that slowdown will have on a user community, you want to know as soon as possible when the hop count changes. This requires the automatic issuance of the TraceRoute command, followed by checking the current hop count against previously logged responses.

If the hop count changes, this indicates that the network is dynamically adapting the path used to connect the workstation to the server. This may be due to network load or to specific devices going offline. It's possible, in Intranet design, to specify static routing of messages for

priority applications. Depending on the amount of fluctuation in the network, you may want to consider using static as opposed to dynamic routing, but be aware of the complexity this will add to the design of your environment: you may gain speed by static routing, but you will lose flexibility.

In order to control the performance of network traffic, it's necessary to identify the links through which messages are travelling either by your own knowledge of the IP addresses involved, or by identifying the host names to which those addresses resolve. If you're using the Internet as part of your network path, you may be able to obtain priority routing from ISPs or their Area Providers, but this usually comes with an associated cost. Once you've identified the routing of message traffic, you need to understand the speed of the links between various network nodes. In the code above, hops 1-17 have round trip times that are under 100ms for each link. However, hop 18 suddenly slows to 3752 ms. This means that for each message from the user workstation to the host server and back, the network beyond the workstation link is running efficiently, but the last link is very slow.

In order to tune the network, you would need to investigate the link between the workstation and the rest of the network first, before trying to control hop counts or reduce path lengths. Identifying the slowest link and then concentrating tuning efforts in that area first will provide the best return for effort in eliminating network response problems.

The final step we'll discuss here has to do with breaking application data into network packets for transmission, and the subsequent reassembly of those packets at the end of the path. When you're moving application data across the network, you need to know what size packets the network expects, to optimize the transmission of data between the workstation and the server. In order to determine the best fit, you need to know what the maximum transmission unit is for the links between the workstation and the server. Shown below are the results of the DMTU utility between a data server and a user workstation:

```
**Jobname: NASTCP  ******* Discover MTU Utility ********Date: 12/Ø3/Ø1**
* Host ID: MVSSYSE                                        Time:  5:Ø5:46 *
* Command: ____________________________________________________________  *
* TTCP52120I - DMTU request completed                                    *
*                                                                        *
```

```
*  Target Host Name :                                                  *
*  Target IP Address: 213.116.100.20  Resolve Addrs to Host Names?: Y  *
*  Timeout (Seconds): 05              Maximum Hops      : 30           *
*                                                                     *
*  Interface Used : 172.22.9.59  _List Link Name: OSALINK2 Primary?: Y *
*                                                                     *
*  Results        : Complete      Path MTU      :    1499            *
* Hop Status   IP Address      Host Name                  Hop MTU      *
* 1 Successful 172.22.9.59     MVSSYSE.landmark.com         2000       *
* 2 Successful 172.22.254.254  gw-nb3-2-tokenring.landmar   2000       *
* 3 Successful 208.203.105.1   fw.landmark.com              1500       *
* 4 Successful 208.203.104.1   fw-cisco.landmark.com       1500       *
* 5 Successful 157.130.39.133  510.Hssi3-0-0.GW3.TCO1.ALT   1500       *
* 6 Successful 152.63.34.122   158.at-1-0-0.XR2.TCO1.ALTE   1499       *
* 7 Successful 152.63.34.122   158.at-1-0-0.XR2.TCO1.ALTE   1499       *
* 8 Successful 152.63.34.38    192.at-1-1-0.TR2.DCA6.ALTE   1499       *
* 9 Successful 152.63.9.217    0.so-6-0-0.IR2.DCA6.ALTER.   1499       *
*10 Successful 146.188.13.46   so-1-0-0.IR2.DCA4.Alter.Ne   1499       *
*11 Successful 146.188.3.106   so-1-0-0.TR2.AMS2.Alter.Ne   1499       *
*12 Successful 146.188.8.81    so-6-0-0.xr1.ams6.nl.uu.ne   1499       *
*13 Successful 212.136.184.141 195.at-7-0-0.cr1.rtm1.nl.u   1499       *
*14 Successful 212.136.177.106 311.atm3-0.dr1.rtm1.nl.uu.   1499       *
*15 Successful 213.116.1.3     tnt3.rtm1.nl.uu.net          1499       *
*16 Successful 213.116.100.20  1Cust20.tnt3.rtm1.nl.uu.ne   1499       *
*                                                                     *
*                                                                     *
** Help Information = PF1*******NASTCP  **** PF Key Assignments = PA1***
```

Note that while we're in the OS/390 environment, the maximum transmission unit (MTU) is 2KB, but after we leave through the host firewall and go out through the router into the Internet, the MTU drops to 1499 bytes. If you were planning to move large volumes of transaction data across this kind of path, it would be better to specify packets based on the Internet sizes rather than on the host network capacity. Otherwise, every time you send a message from the server to the workstation, you'll create fragments of packets once you leave the host. The connection would then be sending 1499 bytes in one frame and 501 bytes in the next frame, and re-assembling the frames in the workstation. With a large volume of transactions, you would end up with half of the network packets using only 1/3 of the network capacity. Remember, for each frame of data, there's a significant portion of the actual frame that is made up of control information to allow the packet to arrive at its intended destination. It's inefficient to duplicate that control information for small fragments of a larger message.

CONCLUSIONS

While network performance tuning requires a combination of disciplines and hardware knowledge, the building blocks of optimal performance are consistent. Of course, each platform, application, and type of network introduces its own complications. But if you aim for consistent rather than optimal performance, the whole user community will benefit.

You will always be able to provide measured improvement to network response times. What you need to ask is, what is the required effort to create that improvement, and what benefit will be realized by the user community? Hopefully, in this case, the benefit realized will be worth all the effort.

Aaron Cain
Pconsulting (UK)

Intrusion detection in z/OS – recent security enhancements in IBM Communications Server

There was a time when network administrators thought a firewall would protect the data centre and the corporation from malicious attacks. Then, they realized that most attacks were coming from within – from contractors selling information for profit, or from disgruntled employees. Now, with worldwide embracement of the Internet, the situation is changing again – an increasing number of these attacks now come from outside. Particularly vulnerable is an open system such as a public Web server that's placed outside the internal network. In the US, the Federal Bureau of Investigation's Computer Security Institute reported that the total number of attacks nearly doubled between 1998 and 1999, and that 70% of all attacks in the year 2000 originated from the Internet.

Intrusion is a broad term encompassing many undesirable activities. The objective of an intrusion may be to acquire information that a person is not authorized to have (information theft). It may be to cause business harm by rendering a network, system, or application unusable (denial of service). Or it may be to gain unauthorized use of a system as a stepping-stone for further intrusions elsewhere. Most intrusions follow a pattern of information gathering, attempted access, and then destructive attacks, and detecting this kind of behaviour is a vital part of intrusion detection.

This article compares the various intrusion detection types and assesses the recent security enhancements in z/OS IBM Communications Server (Version 1, Release 2).

INTRUSION DETECTION SYSTEMS

Intrusion detection systems (IDSs) are generally either network- or host-based approaches to recognizing and deflecting attacks. These products look for suspicious events or traffic patterns that might indicate an attack. Known specific patterns are categorized into attack signatures. When these patterns are looked for in network traffic, it's called network-based IDS; when an IDS looks for attack signatures in

log files, it's host-based. Both approaches have strengths and weaknesses, and each is complementary to the other. A corporation should deploy both intrusion detection approaches.

Once an attack has been detected, a management system should be notified of the attack so that some action can be taken. An effective IDS response should include a variety of options to notify, alert, and respond to the attack. These responses vary by management system, but should at a minimum include administrator notification, connection termination, and session recording for forensic analysis and evidence collection.

NETWORK-BASED IDS

Network intrusion detection systems are based on the real-time recognition of attacks appearing in network traffic. Network IDS typically uses a 'sniffer', placed at strategic points in the network to monitor and analyze traffic in real-time as it travels across the network. Sniffers are network devices operating in promiscuous mode that obtain copies of packets directly from the network media, regardless of their destination (normally devices only read packets addressed to them). Sniffers use pattern matching to try to match a packet against known attack signatures.

Most companies initially deploy network-based intrusion detection, because it's commercially available and implementation is relatively easy. The advantages of network-based IDS include the following:

- The installation of a sniffer does not cause any disruption to the network or degradation to network performance. Individual devices, including hosts, on the network can be (and usually are) unaware of the sniffer's presence.
- All packet headers are examined for malicious and suspicious activity. Because the network media provides a reliable way for a sniffer to obtain copies of raw network traffic, there's no obvious way to transmit a packet on a monitored network without it being seen. However, a skilled attacker can evade detection by exploiting ambiguities in the traffic stream – for example, he might fragment packets to confuse the sniffer, knowing the IP target host will reassemble the attack.

- Attacks are reported as they occur, allowing fast, predetermined responses. Network-based IDS generally includes extensive alerting, including sending pages to network administrators. A denial of service (DoS) attack should initiate an automatic sending of a TCP reset, to terminate the attack before it crashes or damages the targeted host. (Host-based systems usually don't recognize an attack or take action until after a suspicious log entry has been written. By this time, critical systems could already be compromised.)
- A network-based IDS investigates the content of the payload in real-time, looking for commands or syntax used in specific attacks. This means that information that may lead to identification and prosecution can be obtained before a hacker has time to manipulate host log files to cover their tracks.
- The equipment is strategically deployed at critical access points (in front or behind a firewall, in the network, or in front of a host) to view network traffic. Careful selection of detection points can limit the amount of equipment required.
- Network-based IDSs can be placed outside of a firewall where they can detect attacks intended for protected resources. An effective firewall rejects these attacks, masking host-based IDS from information necessary to evaluate and refine security policies.
- A network-based IDS is not dependent on detection by the host operating system, whereas a host-based IDS must continue to function correctly when the host is under attack to generate meaningful results.

HOST-BASED IDS

Host-based intrusion detection began in the early 1980s before networks were as prevalent, complex, and Internet-connected as they are today. In those slower times, a periodic review of audit logs for suspicious activity was sufficient. Intrusions were rare, and after-the-fact analysis was sufficient to prevent future attacks.

Today's host-based intrusion detection systems use 'scanners'. Scanners are not real time, and they work best by evaluating statistical

anomalies, or known attacks. Host-based IDS remain a powerful tool for understanding previous attacks and determining methods to defeat them in the future.

Most host-based IDSs scan audit logs provided by the operating system, detecting attacks by watching for suspicious patterns of activity on the computer system. At the time events are logged, an IDS compares the new log entry with attack signatures to see if there is a match. If so, the system (hopefully) responds with administrator alerts and other calls to action. Some host-based detection systems run, at regular intervals, checksums against key system files and executables, looking for unexpected changes. Other host-based IDSs listen to port activity, and alert administrators when specific ports are accessed. This last type of detection brings an elementary level of network-based intrusion detection into the host-based environment.

Although it's typically slower to react to an attack than network-based intrusion detection, host-based intrusion detection provides certain advantages:

- Host-based detection services have a lower cost of entry – often only a few hundred dollars or less (IBM includes IDS in z/OS Communications as a no-charge feature). Network-based IDS deployments are expensive – a single detection system often costs thousands of dollars.
- Host-based intrusion detection resides on existing network infrastructure, including file servers, Web servers, and other shared servers. This makes them cost-effective.
- Most types of encryption make network-based intrusion detection difficult or impossible. Host-based IDSs do not have this limitation – when an operating system sees incoming traffic, the data stream has already been decrypted.
- Host-based systems can detect attacks from within – attacks that do not cross the network and cannot be seen by network-based IDS – for example, attacks coming from the keyboard of a data centre server.
- Switches allow a large network to be managed as multiple small network segments. This granularity can make it difficult to

identify suitable locations for the deployment of network-based IDS (although traffic mirroring and administrative ports on switches can help). Host-based intrusion detection provides visibility in a switched environment as it can reside on as many hosts as needed.

- Host-based IDSs monitor specific system activities:
  - User and file access can be monitored, including file access, changes to file permissions, attempts to install new executables, and attempts to access privileged services.
  - Activities that are normally executed only by an administrator can also be monitored – for example, the addition, deletion, or modification of user accounts.
  - Improper or questionable change can be detected as soon as it is executed – for example, policy changes that affect what the host tracks in the logs, changes to key system files and executables, or attempts to overwrite vital system files.
- Host-based IDSs use logs containing events that have occurred, enabling them to measure the success of an attack with greater accuracy and fewer false positives (false alerts) than can network-based systems. (The combination of the early warning provided by network IDSs, together with the host verification of an attack, makes an excellent intrusion detection complement.)
- Although host-based intrusion detection doesn't offer real-time response, it can come extremely close. Host-based systems receive an interrupt from the operating system when there's a new log file entry. If this new entry is processed immediately, rather than checked at predefined intervals (as with older systems), the time between attack recognition and response can be significantly reduced. In many cases, an intruder can be detected and stopped before damage is done.

INITIAL z/OS IDS IMPLEMENTATION

Before we look at what host-based intrusion detection means for z/OS Communications Server, it's worth noting that IBM almost never

manages to get it right in the first release of a function. Generally, it takes several releases (read that as several years) to provide a useable and robust implementation. Its IDS implementation is no exception – missing is a management system.

What exacerbates this omission is (my understanding that) the IBM Communications Server development position is that an IDS management system is not within its province. This means that, unless another IBM development group or business partner intercedes to provide IDS management, the current offering is of little value:

- Most glaring in this implementation is the lack of a viable method of informing the network administrator that an attack is underway. The feeble attempt of notification – the sending of event messages to the local console – is totally inadequate. Perhaps they forgot that most companies try to operate 'lights out' with no one around to monitor the local console.
- z/OS IDS requires that a security policy be created, defining what constitutes an attack and what preventative measures should be taken. This security policy is stored centrally in a LDAP repository and from there downloaded to all applicable hosts. Because IBM has yet to provide a policy manager to make the setting and controlling of any policy (eg quality of service), maintaining security policy can quickly become a formidable task.
- There is an extensive and constantly growing list of known intrusion signatures. But missing from the z/OS IDS description is a method of updating the supported signature listings (read that as APARs required). Hopefully, the next release will address this inadequacy.

The events that are detected in this release are: scans, attacks against the stack, and flooding (both TCP and UDP). When an intrusion is detected, the defence mechanisms are packet discard and limiting the number of available connections. As for reporting, a trace of the intrusion can be started and notification of a detected scan via console message or SYSLOGD message sent to the console.

- z/OS defines a scan, the prelude to an attack, as the access by a computer to multiple unique resources (eg ports or interfaces)

over a specified period of time. Both the number of unique events and the interval of time can be user specified. Computer-driven 'fast' scans (generally completed in under five minutes) and intermittent access 'slow' scans (often done over several hours) are recognized. The scan event types supported are: ICMP scans, TCP port scans, and UDP port scans. An exclusion list of legitimate scanners, such as network management, can significantly reduce the false reporting of intrusions.

- z/OS servers have defended themselves from many attacks against the TCP/IP stack for some time. However, z/OS IDS provides new capabilities to control the recording of intrusion, and includes controls to detect and disable uncommon or unused features that could be used in an attack:
  - *Malformed packet events* detect packets with incorrect or partial header information.
  - *Inbound fragment restrictions* detect fragmentation in the first 256 bytes of a datagram.
  - *IP protocol misuse* detects the use of IP protocols that aren't normally used.
  - *IP option misuse* detects the use of IP options that aren't normally used.
  - *UDP perpetual echo* detects traffic between UDP applications that unconditionally respond to every datagram received.
  - *ICMP redirects* detect the receipt of ICMP redirect to modify routing tables.
  - *Outbound RAW sockets* detect z/OS RAW socket applications crafting invalid outbound packets.
  - *TCP SYNflood flood events* detect a flood of SYN packets from 'spoofed' sources.
- Elementary TCP flooding control, introduced in Communications Server for OS/390 Version 2 Release 10, limited the number of inbound connections by port or client. Refinements allow you to specify the number of connections, by application, from the same

port (eg telnet and tn3270), and to set connection limits for all hosts or for an individual host. There's also a 'fair share' algorithm to ensure that no individual exceeds a specified percentage of available port connections.

- Without UDP queue limits, a stalled application or a flood against a single port could consume all the available buffer storage. In the past, a UDP application queue limit could be applied globally to all queues. New UDP traffic regulation allows setting (four) queue lengths, based on the type of application. This enables differentiation among applications that consistently receive data at higher rates than can be processed from fast applications with bursty arrival rates.

CONCLUSION

The host-based IDS provided within z/OS isn't intended, nor is it functionally rich enough, to operate by itself. Host-based IDS is a valuable tool that supplements network-based IDS and should be employed when possible. However, given the inadequacies in this initial IBM release, you should consider waiting until – either through in-house development or by joining up with a network-based IDS provider – future releases provide the network management functions z/OS IDS requires to become a valuable function.

Richard Tobacco
(USA)

Interested in writing an article, but not sure what on?

We've been asked to commission articles on a variety of TCP/SNA-related topics. Contact the editor, Fiona Hewitt, for a list of topics we'd like articles written on.

Network certification

Certification really took off in the 1990s, and by the end of the decade employers were listing it among the required qualifications for their technical networking posts.

But there are now so many certifications, and certification terms and acronyms, that it's almost impossible to keep track of what they all mean. Although the list below isn't exhaustive, it does include the main ones you're likely to encounter. Note that the list includes some certifications with little or no networking component, so that if you run across a certification, you'll be able to find out easily whether or not it includes networking.

The list is presented in glossary format, in alphabetical order. Within the definitions, terms in bold print are those you can find elsewhere in the glossary and that provide additional information.

CERTIFICATION GLOSSARY

A+ – **CompTIA** certification indicating entry-level expertise in personal computer technology.

AATP – Microsoft Authorized Academic Training Provider Program.

ACP – Associate Computing Professional. **ICCP** certification for new entrants into the field of computing.

ATS – Associate Technology Specialist. From The Chauncey Group International. Based on the IT Skill Standards authored by **NWCET**.

BCS – British Computer Society.

BCS Professional Examination – Equivalent to a UK honours degree. Administered by the **British Computer Society**.

CBK – Common Body of Knowledge. A term used by certification organizations to refer to the scope of subject areas that the exam is intended to test.

CCCS – Cisco Cable Communications Specialist. Can support and deploy Cisco cable two-way data services. Includes proficiency in DOCSIS, DVB, RF, and Cisco IOS. See **Cisco** for more details.

CCDA – Cisco Certified Design Associate. Can design routed and switched networks involving LAN, WAN, and dial access for smaller networks. See **Cisco** for more details.

CCDP – Cisco Certified Design Professional. Can design routed and switched

networks involving LAN, WAN, and dial access for larger networks. See **Cisco** for more details.

CCIE – Cicso Certified Internetwork Expert. The CCIE Routing and Switching can install, configure, and operate networks in highly complex environments with specific protocols. The CCIE Communications and Services can provide end-to-end network solutions for the service provider market including Core, Pop, and access technologies. See **Cisco** for more details.

CCIP – Cisco Certified Internetwork Professional. Can plan, design, implement, or operate New World service provider networks. See **Cisco** for more details.

CCNA – Cisco Certified Network Associate. Can install, configure, and operate LAN, WAN, and dial access services for small networks. See **Cisco** for more details.

CCNP – Cisco Certified Network Professional. Can install, configure, and operate LAN, WAN, and dial access services for larger networks. See **Cisco** for more details.

CCP – Certified Computing Professional. **ICCP** certification replacing the Certified Computer Programmer (CCP), **CSP**, and **CDP**.

CDE – Certified Directory Engineer. **Novell** certification indicating expertise in directory-enabled solutions.

CDIA+ – Certified Document Imaging Architech (technical architect). **CompTIA** certification indicating mastery level expertise in document imaging technology.

CDP – Certified Data Processor. See **CCP**.

Certification – Approval by a vendor or professional organization of an individual's competence in a specified area of expertise. Many require passing **CBK** exams.

Certification role – **IBM** certification concept. See **Professional Certification Program** for details.

Certified Advanced Technical Expert – See **Professional Certification Program**.

Certified Developer – See **Professional Certification Program**.

Certified Developer Associate – See **Professional Certification Program**.

Certified Enterprise Developer – See **Professional Certification Program**.

Certified for e-business Professional – Focuses on all the key capabilities required to develop an e-business strategy, select specific technologies and products, and design an implementable solution to execute that strategy. Offered as a Solution Advisor, Solution Designer, or Solution Technologist. See **Professional Certification Program**.

Certified Instructor – See **Professional Certification Program**.

Certified Solution Developer – See **Professional Certification Program**.

Certified Solutions Expert – See **Professional Certification Program**.

Certified Specialist – See **Professional Certification Program**.

Certified Systems Expert – See **Professional Certification Program**.

CIP – Certified Internet Professional. Obsolete **Novell** certification dropped in favour of the **CIW**.

CIPS – Canadian Information Processing Society.

Cisco – Cisco Systems. Large US network hardware and software company, also known for its certification program. All certifications begin with a C (for Cisco) and end with a letter indicating the level of expertise: A for Associate, P for Professional, and E for Expert. S is a Qualified Specialist and is offered in only a single level, indicating a specific area of expertise. See also **CCNA**, **CCNP**, **CCIE**, **CCDA**, **CCDP**, **CCIP**, **CCCS**, **CSS**, **CISA**, **CISAE**, **CISS**, **CSDS**, **CSSS**.

CISA – **Cisco** Internet Solutions Specialist (**CISS**) – Architecture Essentials.

CISAE – **Cisco** Internet Solutions Specialist (**CISS**) – Applications Essentials.

CISS – Cisco Internet Solutions Specialist. Validates skills and knowledge in developing Internet business solutions. The focus is on four key areas of systems planning: applications, tools, operating systems, and networks. The CISS has the technical expertise to integrate applications with the underlying network architecture, providing a foundation of knowledge to accelerate e-business deployment. See **Cisco** for more details.

CISSP – Certified Information Systems Security Practitioner. **ISC2** certification.

CISSS – **Cisco** Internet Solutions Specialist (**CISS**) – Solutions.

CIW – Certified Internet Webmaster. Endorsed by both the Association of Internet Professionals and the International Webmasters Association. Offered in six levels of certification in four Internet tracks: Designer, Developer, Administrator, and Master Web Site Manager.

CLEI – Certified Lotus End-user Instructor. **Lotus** certification indicating experience, expertise, and teaching ability with Lotus-authorized end-user courseware. cf **CLI**.

CLI – Certified Lotus Instructor. **Lotus** certification indicating experience, expertise, and teaching ability with Lotus-authorized courseware. cf **CLEI**.

CLP – Certified Lotus Professional. **Lotus** certification that is superior to the **CLS**, in terms of level of technical expertise.

CLS – Certified Lotus Specialist. **Lotus** certification indicating a basic level of technical expertise.

CNA – Certified Novell Administrator. **Certification** for network administration and support staff in **Novell** environments.

CNE – Certified Novell Engineer. Full range of **Novell** network support capabilities.

CNI – Certified Novell Instructor.

CNS – Certified Novell Salesperson.

CompTIA – Computing Technology Industry Association. Boasts more than 8,000 computing and communications companies as members around the world, providing vendor-neutral standards in certification, e-commerce, customer service, and workforce development. CompTIA certifications are recognized by the plus sign ('+') at the end: **A+**, **CDIA+**, **i-Net+**, **Network+**, **Server+**, **Linux+**, **IT Project+**, **e-Biz+**, **CTT+**.

CSDS – Cisco SNA/IP Design Specialist. Designing Cisco SNA/IP Integration solutions and focus on accelerating the transition of SNA networks to TCP/IP-based networks with Cisco solutions. See **Cisco** for more details.

CSP – Certified Systems Professional. See **CCP**.

CSS – Cisco Security Specialist. See **Cisco** for more details.

CSSS – Cisco SNA/IP Support Specialist. Installing and supporting Cisco SNA/IP Integration solutions and focus on accelerating the transition of SNA networks to TCP/IP-based networks with Cisco solutions. See **Cisco** for more details.

CTEC – Microsoft Certified Technical Education Centres. Full service, non-Microsoft training organizations.

CTT+ – Certified Technical Trainer. **CompTIA** certification acquired from The Chauncey Group International in July 2001.

e-Biz+ – **CompTIA** certification for e-Business. Purchased in February 2001 from Prometric-Thomson Learning, where it was part of the Gartner Institute Certification Program.

ECDL – See **European Computer Driving Licence**.

European Computer Driving Licence – European-wide qualification which enables people to demonstrate their competence in computer skills. **BCS** runs the qualification in the UK on behalf of the ECDL Foundation.

IBM – See **Professional Certification Program** and subsidiaries **Lotus** and **Tivoli**.

ICCP – Institute for Certification of Computing Professionals. Certifications are currently the **ACP** and **CCP**, and formerly the **CDP** and **CSP**.

IISSCC – See **ISC2**.

inCERT – Bi-monthly IBM newsletter on its **Professional Certification Program**.

i-Net+ – **CompTIA** certification indicating a base level of expertise in Internet technologies.

Information Systems Examinations Board – **Certification** arm of the **BCS**.

IPA – Acronym for the French translation of **ISP**.

ISC2 – International Information Systems Security Certification Consortium (IISSCC). The acronym is for ISC squared, which arguably becomes IISSCC. See also **CISSP**.

ISEB – See **Information Systems Examinations Board**.

ISP – Information Systems Professional of Canada. **CIPS**-sponsored certification.

IT Project+ – **CompTIA** certification for Computing Project Management. Purchased in February 2001 from Prometric-Thomson Learning, where it was part of the Gartner Institute Certification Program.

jCert – Java Certification. For developers in a Java environment. The Certification Initiative for Enterprise Development was incorporated as jCert Initiative in 1999 by BEA Systems, Hewlett-Packard, **IBM**, **Oracle**, Sun Microsystems, and Sybase.

LCA – Linux Certified Administrator. Linux and GNU certification indicating the ability to perform as a power user who would feel comfortable staffing a Linux Help Desk or performing as a System Administrator.

Linux+ – **CompTIA** certification indicating vendor-neutral Linux knowledge.

Master CNE – **CNE** plus integration tasks.

MCDBA – Microsoft Certified Database Administrator. Requires the skills to implement and administer **Microsoft** SQL Server databases.

MCP – Microsoft Certified Professional. Requires the skills to successfully implement a **Microsoft** product or technology as part of a business solution in an organization.

MCP+I – Microsoft Certified Professional (**MCP**) plus Internet. Indicates those who can install and configure server products, manage server resources, extend servers to run CGI scripts or ISAPI scripts, monitor and analyse performance, plan security, and troubleshoot problems.

MCP+SB – Microsoft Certified Professional (**MCP**) plus Site Building. Requires the skills to plan, build, maintain, and manage Web sites using **Microsoft** technologies and products.

MCSA – Microsoft Certified Systems Administrator. Requires the skills to implement, manage, and troubleshoot existing network and system environments based on the **Microsoft** Windows server operating systems and Windows .NET Server platforms.

MCSD – Microsoft Certified Solution Developer. Requires the skills to design and develop leading-edge business solutions with **Microsoft** development tools, technologies, platforms, and the Microsoft Windows architecture.

MCSE – Microsoft Certified Systems Engineer. Requires the skills to analyse the business requirements, and design and implement the infrastructure for business solutions based on **Microsoft** server software.

MCSE+I – Microsoft Certified Systems Engineer plus Internet. Requires the skills to enhance, deploy, and manage sophisticated intranet and Internet solutions based on **Microsoft** server software.

MCT – Microsoft Certified Trainer. Qualified instructors who are certified by **Microsoft** to deliver Microsoft Official Curriculum (MOC) and Microsoft Developer Network (**MSDN**) training courses to IT professionals and developers.

Microsoft – Certifications include **MCDBA**, **MCP**, **MCP+I**, **MCP+SB**, **MCSA**, **MCSD**, **MCSE**, **MCSE+I**, **MCT**, and **MOUS**.

MOC – **Microsoft** Official Curriculum.

MOUS – **Microsoft** Office User Specialist.

Network+ – **CompTIA** certification in networking technologies.

Novell – Once the dominant LAN software company. Current certifications include **CDE**, **CNE**, **CNI**, and **CNS**.

NSF – National Science Foundation. Independent US government agency responsible for promoting science and engineering through programs that invest over $3.3 billion per year in almost 20,000 research and education projects in science and engineering. See also **NWCET**.

NUI – **Novell** Users International.

NWCET – NorthWest Centre for Emerging Technology. Commissioned by the **NSF** to define skills standards as needed by the major US computing users: employers, higher education, and government agencies. See also **ATS**.

OCP – See **Oracle Certification Program**.

Oracle – A database and ERP software company. See **Oracle Certification Program**.

Oracle Application Developer – One of four tracks of the **Oracle Certification Program**.

Oracle Certification Program – Consists of four tracks: **Oracle Database Administrator**, **Oracle Application Developer**, **Oracle Database Operator**, and **Oracle Java Developer**. And three tiers: **Oracle Certified Associate**, **Oracle Certified Professiona**l, and **Oracle Certified Master**.

Oracle Certified Associate – The lowest level of Oracle certification. See **Oracle Certification Program**.

Oracle Certified Master – The highest level of Oracle certification. See **Oracle Certification Program**.

Oracle Certified Professional – The middle tier of Oracle certification. See **Oracle Certification Program**.

Oracle Database Administrator – One of four tracks of the **Oracle Certification Program**.

Oracle Database Operator – One of four tracks of the **Oracle Certification Program**.

Oracle DBA – See **Oracle Database Administrator**.

Oracle Java Developer – One of four tracks of the **Oracle Certification Program**.

Oracle University – The education arm of **Oracle**.

Professional Certification Program – **IBM** certification available to those outside the company. Indicated in two parts, together known as a certification role: (1) an expertise level and (2) a product or skill area. There are three expertise levels. Level 1 is the lowest and consists of **Certified Specialist** and **Certified Developer Associate**. Level 2 is **Certified Solutions Expert**, **Certified Systems Expert**, **Certified Developer**, **Certified Instructor**, and **Certified for e-business**. Level 3 is **Certified Advanced Technical Expert**.

Recertification – The process required to renew a time-limited **certification**. For example, most **Cisco** certifications are valid for only three years before recertification.

Server+ – **CompTIA** certification covering PC hardware issues.

SME – Subject Matter Expert. See **CBK**, **certification**.

Sun Certified – Sun certification that includes: Sun Certified Programmer for Java 2 Platform, Sun Certified Developer for Java 2 Platform, Sun Certified Web Component Developer for the J2EE Platform, Sun Certified System Administrator for the Solaris Operating Environment, Sun Certified Network Administrator for the

Solaris Operating Environment, Sun Certified Data Management Engineer, Sun Certified Back-up and Recovery Engineer, and the Sun Certified Storage Architect.

TCC – See **Tivoli Certified Consultant**.

TCEC – See **Tivoli Certified Enterprise Consultant**.

TCP – See **Tivoli Certificated Professional**.

TCSE – See **Tivoli Certified Solutions Expert**.

Tivoli – **IBM** subsidiary with its own certification program. See **Tivoli Professional Certification Program**.

Tivoli Certificated Professional – One of four **Tivoli** certifications: **Tivoli Certificated Professional**, **Tivoli Certified Consultant**, **Tivoli Certified Enterprise Consultant**, and **Tivoli Certified Solutions Expert**.

Tivoli Certified Consultant – **Tivoli** certification reflecting mastery in a single area of Tivoli software for the large enterprise.

Tivoli Certified Enterprise Consultant – A higher level certification than the **Tivoli Certified Consultant**, recognizing either mastery of a broad group of subjects or in-depth expertise in a single complex area of Tivoli Enterprise software.

Tivoli Certified Solutions Expert – **Tivoli** certification covering its software for small/medium businesses.

Tivoli Professional Certification Program – **Tivoli**'s certification program. See also **Tivoli Certificated Professional**, **Tivoli Certified Consultant**, **Tivoli Certified Enterprise Consultant**, and **Tivoli Certified Solutions Expert**.

TPCP – See **Tivoli Professional Certification Program**.

Chris Bruns
(Canada)

Leaving? You don't have to give up *TCP/SNA Update*

You don't have to lose your subscription when you move to another location – let us know your new address, and the name of your successor at your current address, and we will send *TCP/SNA Update* to both of you, for the duration of your subscription. There is no charge for the additional copies.

XML – a 'first-cut' tutorial

XML, or eXtensible Mark-up Language, permits incompatible systems, perhaps belonging to different organizations, to interchange data in a meaningful and productive manner – hence the claim that it is the *lingua franca* for e-business. XML provides a common, platform- and programming language-independent scheme for sharing data – in an unambiguous and consistent manner. Although it was formalized by the World Wide Web Consortium (W3C) only in February 1998, by the year 2000, XML was already being hailed as the great unifying technology for the Web – the 'next big thing'. Everything to do with the Web now has some XML connotation. Take the new Web Services, for example – modular, self-contained, self-describing e-applications that can be published, readily located, and easily invoked across the Web. Everything about Web Services is so XML-centric that some technical people already refer to them as 'XML Web Services'.

IBM's support for XML is extensive. WebSphere Application Server Version 4 has many XML-related capabilities, including uncompromised support for all the XML-based Web Services enabling protocols: SOAP, WSDL, and UDDI. In addition, WebSphere Host Publisher 2.2 offers 'on-the-fly', bi-directional host-to-XML data conversion, thus enabling screen-mode input/output to be easily represented in XML form. XML support can also be found in WebSphere Studio, WebSphere Business Integrator, WebSphere Partner Agreement Manager, WebSphere Transcoding Publisher, and the WebSphere Portal Family – with the latter also embracing XML Web Services. DB2 Version 7 has an XML capability, as does MQSeries Integrator (now WebSphere MQ Integrator). The new COBOL compiler supports XML, and there is an XML Toolkit for z/OS and OS/390. Clearly, IBM expects XML to play a pivotal role in the future of e-business, and Microsoft obviously feels the same: Excel 2002, in Office XP, has an XML spreadsheet file format that allows users to import, export, and analyze XML data. XML-based data interchange is also possible with SQL Server 2000, Access 2002, and BizTalk Server 2000.

WHAT XML IS ALL ABOUT

XML is a meta-mark-up language for documents – in particular documents containing structured information. Most documents have some level of structure, in that the information they contain is made up of content (eg text, graphics) and context (eg headings, tables, captions). XML's strength is its ability to describe the context of the data relative to a document. HTML, by contrast, does not deal with this context and focuses instead on how to display content (see below for more about this).

Data (ie content) is included in an XML document as strings of text. The data is bracketed by XML text mark-up which sets out to describe that data – that is, to give it context. The basic building blocks of an XML document are called 'elements', where an element is a specific unit of data along with the XML mark-up describing that data. The XML mark-up, in much the same way as HTML, is in the form of tags, with the tags appearing within angled-brackets. An XML element is thus delimited by two tags: a start-tag and an end-tag. The element *per se* will typically consist of these two tags with text (ie data) in the middle. In some cases, there can be more XML mark-ups between the original start and end tags. XML thus takes a flat stream of text which represents data of some sort, and transforms it into a set of self-describing objects that can be easily and consistently manipulated by recipients.

Since XML documents are always in text form, they can always be read and deciphered by humans. However, in order for an application to be able to successfully process a XML document, it needs to know what the tags mean. This is where the rub comes with XML: because the meaning of a tag can differ significantly between different organizations, countries, and industry sectors, an application really needs to know what each tag means within a specific 'application domain'.

So, just having a well-formed XML document doesn't guarantee that it can and will be correctly interpreted by any and all applications. An analogy widely used in the mid-1980s to explain the need for networking protocols and architectures is relevant again here: if you attempt to make a direct-dial phone call between London and Moscow, you will, with luck, get a connection; but there's no guarantee that

you'll be able to hold a meaningful conversation unless both of you happen to know a common language. To its credit, XML provides two main schemes to facilitate this mutual understanding: Document Type Definitions (DTDs) and XML schemas. In some special cases, it's possible to have a DTDless XML document, provided the elements are structured in some type of self-explanatory manner. DTDs and XML schema will be described later.

XML is derived from the Standard Generalized Mark-up Language (SGML), which became an ISO standard (ISO 8879) in around 1985. It's the standard for defining descriptions of the structure of different types of electronic document, and has been widely used by the US military, the US government, and the aerospace industry over the last decade. SGML, by design, is very detailed, powerful, and complex. It was too unwieldy to be easily adopted for the Web and e-business. XML is 'lite' SGML, retaining enough of the SGML functionality to make it useful but removing much of the optional and redundant features which make SGML so convoluted.

CONTRASTING HTML WITH XML

The fundamental distinction between HTML and XML is that HTML defines data presentation (or data rendering) whereas XML defines the meaning of the data. This is best illustrated with an example. Let's start with some sample HTML code that could be found within a document containing contact information for an individual:

```
<p><b>Mr. Anura Guruge</b>
<br>
Principal
<br>
i-net guru
<br>
4 Varney Point Road, Left
<br>
Gilford, NH Ø3249
<br>
USA
<br>
anu@wownh.com</p>
```

In this HTML example, the <p> tag represents the start of a new paragraph, the <b> tag calls for bold text, and the
 tag specifies a line break. Note that HTML doesn't require or define a </br> tag.

This would not be permissible in XML. In XML, all open tags have to be explicitly closed. This HTML code, when rendered by a Web Browser, would show ‘Mr. Anura Guruge’ in bold with the rest of the information underneath it, in separate lines as per the breaks dictated by the
 tags.

The problem with this HTML is that it only describes the layout for the data contained in the document; there’s no description as to what the data means. Although a human could interpret this, it would be difficult for an application to determine what all these fields meant unless it was explicitly programmed to look for contact information in this type of format. A good example of the potential for ambiguity with HTML documents can be seen by doing a search for a key word such as ‘bill’ using any of the popular Web search engines, such as Google.com. Such a search will return thousands of entries covering people’s names (eg Bill Gates), laws (eg Bill of Rights), theatre play lists, and monetary notes.

With XML, the contact information shown above will be organized between tags that set out to describe the data. So, one possible XML representation for some of this data might look like:

```
<contact_info>
<name>
<salutation>Mr.</salutation >
<first-name>Anura</first-name>
<last-name>Guruge</last-name>
</name>
<title>Principal</title>
<company>i-net guru</company>
<address>
<street>4 Varney Point Road, Left</street>
<city>Gilford</city>
                  .......
</address>
<e-mail>anu@wownh.com</e-mail>
</contact_info>
```

The first thing to note is that the XML representation shown here is arbitrary. You could describe this data in other ways using tags with different names. That is what the ‘extensible’ part of XML refers to. XML is a meta-mark-up language. This means that XML doesn’t contain a fixed set of tags and elements that has to be used by everybody. Instead, with XML you can define the elements you want

as you go along, and design your own customized mark-up languages for limitless different types of document. This means that you can have elements specific to organizations and industries. But the key thing to remember is that you'll need a DTD or XML schema that describes what's expected and acceptable within a specific XML document – especially if you want that XML document to be processed by applications written by others.

As well as using different tag names and possibly a different hierarchy (eg break down the address further into <unit number> and <street name1> etc), another immediate variation that would be possible with the above XML would be to use XML attributes. XML elements can have attributes – where an attribute is a name-value pair that's attached to an element's start tag. The name and value making up a particular name-value pair will be separated from each other by an equals sign and optional whitespace. The value of a name-value pair is enclosed in either single or double quotation marks. Thus, it would be possible to say:

```
<contact_info>
     <entry name="Anura Guruge">
         <address>  ....
                 </address>
     </entry>
</contact_info
```

Contrasting the HTML with XML it's immediately obvious that the XML doesn't provide any formatting guidelines. However, it's also apparent that a standard browser would be able to interpret the XML structure and display the data portion of an XML document: Microsoft Internet Explorer 5 and Netscape 4.*x* can both display XML documents.

There are two main ways to add formatting information to XML documents: eXtensible Stylesheet Language (XSL) and XHTML. These will be discussed later, following a more detailed look at a more complex XML document.

AN XML REPRESENTATION OF A 3270 SCREEN

XML can be used to describe any kind of data – including data interchanged via screen-based terminal input/output such as that on a 3270 or 5250. Within the SNA world, a major use for XML will be that

of facilitating host data interchange with new e-applications, with screen data from host applications being represented in XML. The following XML document describes a 3270 screen using an XML definition scheme favoured by host access vendor Aviva Solutions (www.avivasolutions.com). The screen being described in XML is the one shown in Figure 1.

Lines 3 to 5 of the XML representation on page 48 illustrate the use of attributes within XML elements; eg, size width=80, height=24 – in other words, a standard 24x80, 327*x* Model 2 screen. The document also exploits one of the few 'shortcuts' permitted in XML: the notion of 'empty elements'. An empty element is one that does not contain content, and attributes are not considered content *per se*. Empty elements can be terminated with a simple /> tag without a name, hence the /> tags on lines 3 to 5.

The description of the screen itself starts with the <Screen> tag on

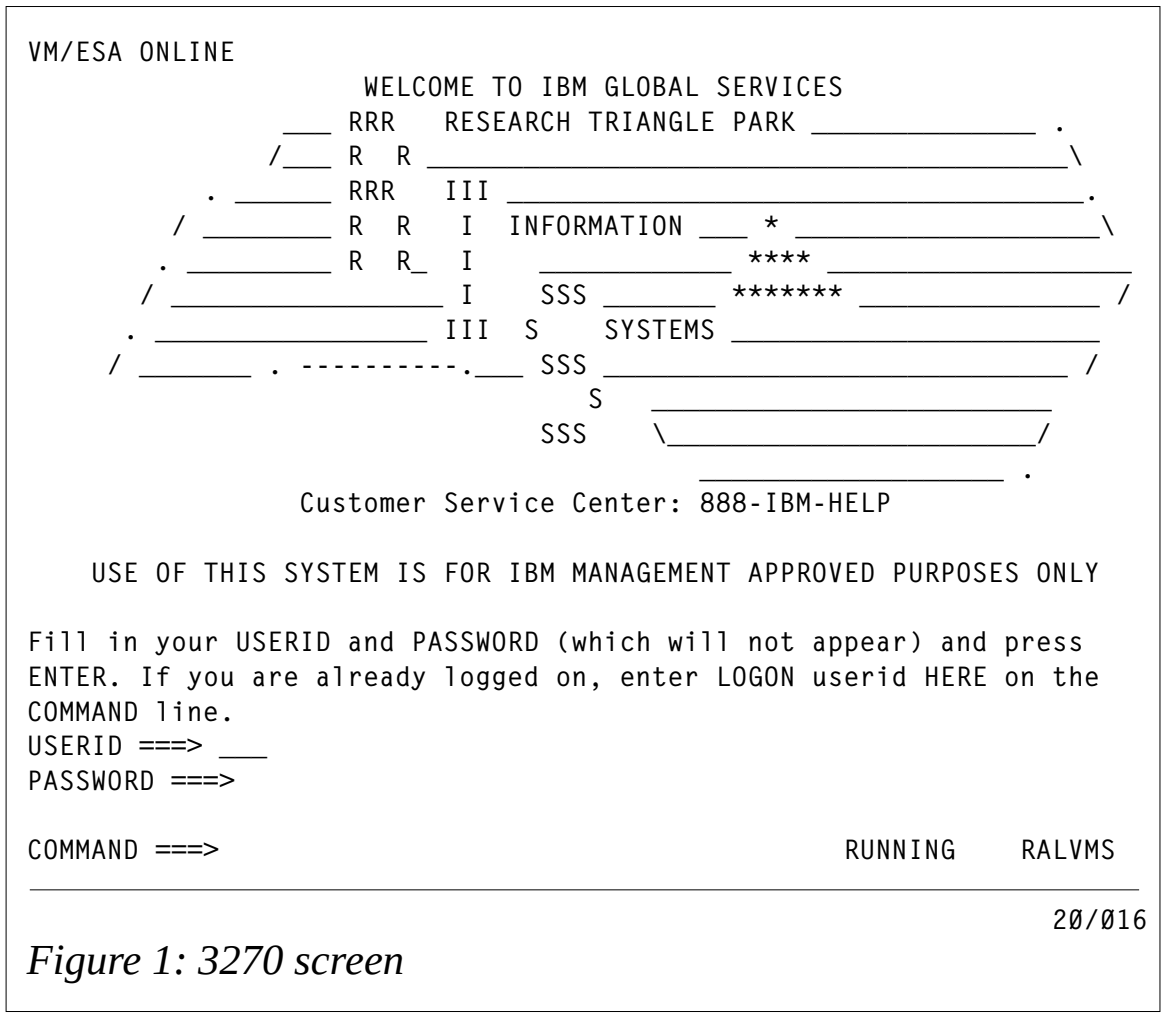

Figure 1: 3270 screen

line 6. The </screen> tag appears on the penultimate line of this document. Each field of the screen is then described by a ‘field’ element which has its own set of attributes. The overall specification of this screen is self-explanatory, except that the values assigned for the various foreground and background colours are arbitrary and proprietary.

The first line of this XML document includes the optional, but invariably included, XML declaration, specifying an XML version number and the encoding scheme being used. In this case, the encoding is the default: UTF-8, the variable length encoding of the Unicode character set which is a superset of ASCII. Another attribute that could have appeared on this XML declaration statement is ‘standalone=’. The default, if it’s omitted, is standalone=“no” (as here). The ‘standalone=’ attribute refers to whether an external DTD is required in order to be able to successfully process the XML document in question. A value of standalone=“no” signifies that an external DTD is likely to be required in order to ascertain the proper values for parts of the XML representation of the 3270 screen.

```
<?xml version="1.0" encoding="UTF-8"?>
<XmlTerminalData3270>
<Response code="0"/>
<Size width="80" height="24"/>
<Cursor col="2" row="21"/>
<Screen>
<Field row="1" col="2" length="78" protected="true">
<Segment offset="0" length="78" foreground_color="#FFFFFF"
background_color="#000000"/>
<Content>VM/ESA ONLINE
</Content>
</Field>
<Field row="2" col="1" length="1280" protected="true">
<Segment offset="0" length="1280" foreground_color="#0000FF"
background_color="#000000"/>
<Content>                              VV          VV MM          MM
VV          VV MMM        MMM
VV          VV MM M      M MM
VV          VV MM  M    M  MM
VV        VV  MM   M M    MM
VV      VV   MM    M     MM
EEEEEEEEEEEE   SSSSSSSSSS       MAAAAA
EE           VV SS     MM  SS     AA   AA
EE            VVSS     MM         AAM    AA
EE              VSS     MM        AAMM     AA
EEEEEEEEEE      SSSSSSSSSS    AAAAAAAAAAAA
```

```
EE                        SS  AA          AA
EE                        SS  AA          AA
EE              SS        SS  AA          AA
EEEEEEEEEEEE    SSSSSSSSSS    AA          AA
</Content>
</Field>
<Field row="18" col="2" length="79" protected="true">
<Segment offset="0" length="79" foreground_color="#0000FF"
background_color="#000000"/>
<Content>
</Content>
</Field>
<Field row="19" col="2" length="55" protected="true">
<Segment offset="0" length="55" foreground_color="#FFFFFF"
background_color="#000000"/>
<Content>                     McGill University Computing Centre</
Content>
</Field>
<Field row="19" col="58" length="23" protected="true">
<Segment offset="0" length="23" foreground_color="#0000FF"
background_color="#000000"/>
<Content>                       </Content>
</Field>
<Field row="20" col="2" length="49" protected="true">
<Segment offset="0" length="49" foreground_color="#FFFFFF"
background_color="#000000"/>
<Content>                             VM/ESA 2.4.0 slu 0001</Content>
</Field>
<Field row="20" col="52" length="29" protected="true">
<Segment offset="0" length="29" foreground_color="#0000FF"
background_color="#000000"/>
<Content>                              </Content>
</Field>
<Field row="21" col="2" length="8" protected="false">
<Segment offset="0" length="8" foreground_color="#00FF00"
background_color="#000000"/>
<Content>        </Content>
</Field>
<Field row="21" col="11" length="70" protected="true">
<Segment offset="0" length="70" foreground_color="#0000FF"
background_color="#000000"/>
<Content>
</Content>
</Field>
<Field row="22" col="2" length="8" protected="false">
<Segment offset="0" length="8" foreground_color="#00FF00"
background_color="#000000"/>
<Content>        </Content>
</Field>
<Field row="22" col="11" length="70" protected="true">
<Segment offset="0" length="70" foreground_color="#0000FF"
```

```
background_color="#000000"/>
<Content>
</Content>
</Field>
<Field row="23" col="2" length="138" protected="false">
<Segment offset="0" length="138" foreground_color="#00FF00"
background_color="#000000"/>
<Content>
</Content>
</Field>
<Field row="24" col="61" length="20" protected="true">
<Segment offset="0" length="20" foreground_color="#0000FF"
background_color="#000000"/>
<Content>RUNNING   VM1       </Content>
</Field>
</Screen>
</XmlTerminalData3270>
```

Just to highlight XML's flexibility, another XML representation of the first few lines of this same 3270 screen is shown below, but this time using an XML scheme proposed by IBM. Although there are some obvious similarities, there are also some significant differences. Note, for example, that very different techniques are used to describe the display fields: Aviva Solutions describes the display fields using elements called 'content', while IBM refers to the entire screen via an element called 'content'. The display fields are specified, as data (ie text) within field elements. The beauty (and possibly also the bane) of XML is that both these XML representations are perfectly valid and legitimate, provided that there are corresponding DTDs to define what to expect within these screen definition documents.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xml_host_data>
  <session>
    <id type="1">N5FU3IQAAAAAACJFGNPAAAA</id>
    <host port="23">ralvms</host>
    <size>2</size>
    <codepage>037</codepage>
    <sessionnumber>2</sessionnumber>
    <reconnect>false</reconnect>
  </session>
  <screen cursor="1536">
    <content>
      <field position="2" length="13" protected="true" numeric="false"
hidden="false" reset="false" modified="false">VM/ESA ONLINE</field>
      <field position="16" length="67" protected="true" numeric="false"
hidden="false" reset="false" modified="false">
```

```
</field>
      <field position="84" length="20" protected="true" numeric="true"
hidden="false" reset="false" modified="false">                    </
field>
      <field position="105" length="73" protected="true" numeric="true"
hidden="false" reset="false" modified="false">WELCOME TO IBM GLOBAL
SERVICES                                              </field>
      <field position="179" length="4" protected="true" numeric="true"
hidden="false" reset="false" modified="false">___ </field>
      <field position="184" length="5" protected="true" numeric="true"
hidden="false" reset="false" modified="false">RRR  </field>
      <field position="190" length="22" protected="false"
numeric="false" hidden="false" reset="false" modified="false">RESEARCH
TRIANGLE PARK</field>
```

XSL, XSLT, AND XHTML

Once you have an XML document, people invariably want to be able to display it on various devices, with Web Browsers and intelligent cell phones being the most popular at the moment. But it doesn't end here. A lot of work is currently taking place on VoiceXML for voice-based applications such as the voice response systems used by call centres. XML documents are typically displayed on a browser or cell phone by associating a style sheet with the XML document. In some cases, this can be achieved by using standard Cascading Stylesheets (CSSs). XSL, however, is the more strategic approach given that CSS uses a non-XML syntax. It is hoped that the next generation of browsers will contain sufficient XML functionality to enable them to reasonably display most XML documents without the need for auxiliary stylesheets.

XSL defines the format for an XML document. XSL is divided into two parts: XSL Formatting Objects (XSL-FO) and XSL Transformations (XSLT). XSL-FO is deemed to be an XML application that can be used to describe the layout of a page in terms of blocks of text, graphics, and horizontal lines. Most people, however, don't create XSL-FOs; instead, they write an XSLT stylesheet that transforms the XML in your document into the corresponding XSL-FOs.

XSLT is also called an XML application, and specifies how one XML document can be transformed into another. XSLT works through the use of XSLT stylesheets (sometimes referred to as XSLT documents). XSLT stylesheets contain templates and are XML documents in their

own right. XSLT works by comparing the elements in an input XML document being converted with the templates appearing in the stylesheet. When it finds a match, it creates a corresponding output according to what's specified in the template. You can have multiple stylesheets for the same XML document: one for displaying the document in HTML form within a browser and another for displaying it in some type of WML form on an intelligent phone. Internet Explorer 5.5 has a built-in XSLT processor that enables it to accept XML documents and corresponding stylesheets and process the necessary XSLT transformations on the fly. If the browser doesn't support XSLT transformation, a separate XSLT processor will have to be used – for example, the 'open-source' Apache Software Foundation's Xalan (in Java or C++) from www.apache.org.

XHTML is an XML-based variant of HTML 4.0, and is an official W3C recommendation. It reformulates HTML 4.0 to ensure that it meets XML's syntax requirements. There are three DTDs available that describe HTML 4.0 in terms of XML. The big thing with XHTML is that it imposes strict discipline on HTML. For a start, unmatched tags are not allowed. So all start tags have to have matching end tags. All attribute values now have to appear within quotes. XHTML also enforces case sensitivity in the case of tags, deeming <p> valid, and <P> invalid. Most of today's browsers, irrespective of their vintage, can adequately render most XHTML documents.

DTDs AND XML SCHEMA

A DTD is a formal description of a particular class of XML document in XML declaration syntax. XML borrows the concept of DTDs from SGML. A DTD specifies what structures are permissible within an XML document, spelling out what names are to be used for the different types of element, where they may occur, the attributes that can be assigned to elements, and how they all fit together.

Every element that can appear within an XML document needs to be declared within that document's DTD with an element declaration statement. The basic structure of an element declaration statement is as follows:

```
<!ELEMENT element_name (content_description) ['?' | '*' | '+']>
```

where ‘?’, ‘*’ and ‘+’ are wild-card references: ‘?’ indicates that the preceding name (or associated content description) can occur zero or one times; ‘*’ denotes zero or more times; and ‘+’ one or more times.

So a simple DTD for a XML document containing contact information may look as follows:

```
<!ELEMENT person (name, company*)>
<!ELEMENT name (salutation?, first_name, middle_name*, last_name)>
<!ELEMENT salutation (#PCDATA)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT middle_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT company (#PCDATA)>
```

The term ‘#PCDATA’, often seen in DTDs, refers to Parsed Character Data – essentially a string of text. If an element is defined as being of type #PCDATA, it cannot have sub-elements (or child elements). So, for example, if you define an element ‘phone_number’ as being <!ELEMENT phone_number (#PCDATA)> you will, within the corresponding XML document, have to define the phone number as a string of text; you couldn’t subdivide it into separate elements corresponding to ‘country_code’, ‘area_code’, ‘extension’, etc.

A DTD would typically be stored as a file (with the ‘.dtd’ suffix), separate from the XML documents it describes. It could also be included within the XML document it describes. The location of the DTD that describes a given XML document is specified within that document via a Document Type Declaration, such as:

```
<!DOCTYPE contact_info SYSTEM "http://www.wownh.com/dtds/
contactinfo.dtd">
```

The <!DOCTYPE > declaration usually follows the <?xml version= ...?> statement that begins an XML document. If the DTD is to be included within the XML document, it’s embedded as a part of the Document Type Declaration statement as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE person [
    <!ELEMENT person (name, company*)>
    <!ELEMENT name (salutation?, first_name, middle_name*, last_name)>
    ....
]>
<person>
  <name>
```

```
    <salutation>Mr.</salutation>
    <first_name>Anura</first_name>
.....
.....
</person>
```

As mentioned above, XML's success depends on a common understanding of what an XML document represents at both ends of a 'transaction', and DTDs are one way in which this understanding can be achieved. There is now a relatively rich body of public DTDs for various industry sectors and applications: eg loan processing within the financial sector, property description for real estate, etc. (See www.xml.org/xmlorg_registry. See also www.schema.net and www.biztalk.org.)

DTDs just specify the structure of an XML document. Since the roots of DTDs go back to SGML, their strength is in describing conventional text documents, and they don't have a mechanism for expressing the content of elements in terms of data types. This means that a DTD cannot be used to specify numeric ranges for an element, to define limitations on what can occur, or checks on the text content. What's more, the syntax of DTDs is relatively complex, and, ironically, they're written in their own special syntax rather than in XML. Because of this, DTDs are now being usurped by XML schemas (see the W3C recommendation at http://www.w3.org/TR/xmlschema-0/).

XML schemas are written in standard XML, and can provide a far more rigorous description of the contents of an XML document in a modular, typed, and object-oriented manner. Since they permit data types to be rigorously specified, developing a good XML schema will require more thought and work than was required to create a DTD – especially if the XML document being described contains complex data types. As with DTDs, industry- and application-specific XML schemas are already available, and many new ones are being defined.

THE BOTTOM LINE

Just before we conclude, we need to consider some of the things that XML is not and cannot do, just to dispel any lingering misconceptions. XML is not a programming language, a database schema, or a networking protocol. XML documents can and are transported across

networks using standard, widely-used protocols such as FTP, HTTP(S), SOAP, and SMTP. Databases, eg DB2 Version 7, will support XML data and even permit its data to be described in XML form, but the database itself will not be in the form of a large XML document.

XML addresses what has been the holy grail of networking, right from the very early days: universal data interchange that transcends platforms, networking protocols, and programming languages. Now, with the Web providing global connectivity, XML provides what SNA, OSI, and others always aimed for – consistent data interchange capability, without barriers.

XML is the most strategic and widely adopted standard of its type. Most people associated with the technical aspects of the Web and e-business cite it as the next big thing – with XML-based Web Services just adding more fuel to this claim. And, since there appears to be no real alternative to XML on the horizon, all the indications are that XML will be the great unifying force that pushes e-business to its next level.

Anura Gurugé
(USA)

Looking for a specific article?

If you keep hoping for an article on a particular topic, but we never publish one, please let us know what the subject is. If it's likely to be of interest to other subscribers too, we'll commission it and publish it in *TCP/SNA Update*.

Visit the *TCP/SNA Update* Web site

http://www.xephon.com/tcpsna

and follow the link to Opportunities for TCP/SNA specialists.

Web-based terminal emulators – reflection for the Web

So far, in our series on TCP/IP-based 3270 mainframe terminal emulation software, we've concentrated solely on Windows-based products; this issue, we move on to Web-based products. Web-based software is more difficult to test because it requires software installation privileges on a Web server. In fact, many of these products are solely Web-based in the sense that their modules are installed only on the Web server, and not on the client workstation or mainframe host.

The movement from Windows- to Web-based software is a major trend in most software areas. Although it sounds more complicated, it's actually cheaper to support from the vendor's perspective – it can be very difficult to write Windows-based software that will run:

- On all versions of Windows.
- With every other piece of software, and combination of versions of software.
- With every piece of hardware.

Plus, Web-based software will run on Mac, Linux, AIX, Unix, and other workstation platforms – anywhere that a Web browser will run. Admittedly, most require a Java Virtual Machine (JVM) these days, but that, too, is common to recent browser versions on all platforms.

A recent Web-to-Host study by The Tolly Group indicates that total cost of ownership (TCO) reductions of 60% are possible by switching from Window-based to Web-based for IBM mainframe and iSeries 400 hosts. Web-based also makes it practical to provide host access (typically integrated into an application) almost anywhere, because it eliminates the problem of installing software on every workstation as would be required with Windows-based software.

Our look at Web-based 3270 terminal emulators begins with WRQ's Reflection for the Web. With the menu bar enabled (not the default), it provides the same look and feel as Reflection for Windows (see *TCP/SNA Update* 41, March 2001 pp 14-20).

The Reflection home page is accessible from WRQ's corporate Web site at http://www.wrq.com; once there, you'll find a Web-Based Emulation link to a page with a brief description of the product, as well as lots of other links:

http://www.wrq.com/products/reflection/fortheweb

Click the *Product evaluation* link in the section of the left sidebar labelled Additional Information, and you're taken to a Web page that offers four ways to evaluate the product:

http://www.wrq.com/products/evaluate/download/rvchoice.html

The four choices are:

- A preconfigured Reflection for the Web session.
- Custom sessions on WRQ servers.
- Download a 60-day trial, complete with documentation, hosted on your own Web server.
- The same as the above, but mailed on a CD-ROM instead of downloaded.

A link is also provided to Technical Note 1359 which provides:

- Additional information on each choice and how to choose between them.
- Some installation and customization instructions.
- Popular features and their use.
- Links to a few of the most relevant Technical Notes, including 9982, which lists all Reflection for Web technical notes, and 1528 with Frequently-Asked Questions (FAQs) regarding the download procedure.

Each Technical Note has its own URL formed by replacing NNNN with its number:

http://support.wrq.com/techdocs/NNNN.html

http://support.wrq.com/techdocs/1359.html

DOWNLOADING THE EVALUATION VERSION

I chose option three. Because this is a time-limited edition of the full server-based product, the download is huge: 24.5MB. It can be accessed directly from:

http://www.wrq.com/products/evaluate/download/rweb_dwnld.html

There are four choices of language, but, because Reflection for the Web is written in Java, only one choice of platform (Java). I chose English, Java, and was then prompted for my e-mail address and other registration information. When I hit the Send button, a long Agreement page appeared, containing an export notice, an end-user licence agreement, and a limited warranty. (The latter was particularly amusing, since it covered me for 90 days even though the software expired in 60!)

When I pushed the Accept button, the Evaluation Software Download Page was displayed, beginning with the version being downloaded (Version 4.50 for Java in English), a description of how the installation process works, and a note explaining why 56-bit encryption is used instead of the 168-bit you'll receive if you buy the product. A Download table followed, indicating which of three packages should be installed, depending on your computer environment.

How the Installation Process Works, at the top of the Web page, states:

> The package that you download depends on the Java capabilities of the computer on which you are installing the components.

This can refer to the Web server (optional metering and security components) or client workstation (terminal emulation component). The administration component can be run on either.

The packages range in size from 20-24.2MB. I needed the largest, rv450ejw.exe, as it is for Windows-based Java installers, and the download took 2.7 minutes, averaging 1.6Mbps on a 2.5Mbps ADSL line through a hardware firewall.

EXTRACTION AND READING

I opened the folder where I stored the download , and double-clicked on the downloaded file to execute it. A WinZip Self-Extractor

dialogue box immediately came up, asking for confirmation of the current folder as the place to unzip the files. Since I didn't have WinZip installed on the server, I pushed the Unzip button rather than the Run WinZip button. After it reported that five files were unzipped, I pushed the Close button and found five new files in the folder:

- readme.html
- rwebemulation.exe
- rwebmeter.exe
- rwebproxy.exe
- webstation.exe

The final step on the download page was to read the readme.html file. Double clicking on it opened up a new Internet Explorer window with a very long Web page entitled *Reflection for the Web Installation Guide*. Like all good designs of this type of single Web page approach, the beginning of a table of contents (toc) is visible without scrolling, and each toc entry is a link to further down the Web page.

The first toc entry is an Overview, which identifies the four components (the four .exe files above) by name:

- Administrative WebStation – required
- Terminal emulation – required
- Usage metering – optional
- Security proxy server – optional

The Installing Reflection Components section of readme.html began by explaining where each of the four components could be installed. The Administrative WebStation can be installed on a single administrator's client workstation or on a Web server to allow access from several workstations by one or more administrators. The Terminal Emulation component is normally installed on a Web server. A tip in the same section offers an intriguing alternative:

> For testing Reflection for the Web, you may choose to install this component onto your local hard drive to simulate an installation on to a Web server.

The next subsection, Installation Options, explained that each of the four .exe files listed above, corresponding to the four components of Reflection for the Web, are actually Windows-based installers, presumably each with all of the files and programs for a component embedded and compressed inside an .exe file. Each component is installed by executing it, by double-clicking.

INSTALLING ADMINISTRATIVE WEBSTATION

The Administrative WebStation seems to need to be installed first, as it's used to define the connection information for the host (mainframe) needed by the Terminal Emulation component. Double-clicking webstation.exe started the InstallShield Wizard. Everything was fine until the default Destination Folder displayed as blank, a by-product of WRQ's use of a Windows wrapper on a platform-independent Java installer. When I pushed the Change button a Select Directory dialogue box appears, which defaults to C:\Program Files, and allows you to browse, but not to create a new directory. Instead, you can type \Reflection after C:\Program Files in the Folder field and hit the OK button and then the Next button. A Confirm dialogue box appears with Yes and No buttons, and reads:

> The directory "C:\Program Files\Reflection" does not exist. Do you want to create it now?

I hit Yes and the installation continued, taking about a minute and a half, listing each file as it was stored in the new folder. The final wizard page was labelled Information, had a single button, Exit, and read:

> The Reflection for the Web Administrative WebStation is now installed. To run the WebStation, launch webstation.html in your web browser. If you have installed Reflection for the Web on a Windows computer, you can also run the WebStation from the Start menu.

True to its word, Start/Programs/Reflection for the Web/Administrative WebStation launched Internet Explorer and opened C:\Program Files\Reflection\webstation.html, which displayed a logo, but that quickly changed to C:\Program Files\Reflection\html\default_ie.htm. If you can't wait the four seconds for the logo to disappear, there is a link you can click: Skip to Administrative WebStation.

ADMINISTRATIVE WEBSTATION HOME

The Administrative WebStation Home page offers three links for the most common tasks:

- Create terminal sessions
- Upload sessions to a Web server
- View Reflection activity reports.

Two sections follow: first-time users and upgrading.

The left sidebar offers the following choices, which are also repeated as links across the bottom of the Web page: Home, search, welcome, how to, deployment director, reports, advanced, and resources.

CREATE TERMINAL SESSIONS

The Create terminal sessions link was the obvious choice, which led to a Web page entitled Deployment Director and subtitled Terminal Session. The left sidebar's Deployment Director item was broken down into four subsections:

- Terminal session (highlighted)
- Security archive
- File upload
- Default settings.

The Web page text read:

> Use the Terminal Session section of the Deployment Director to create, modify, or deploy a terminal session. After you create the Reflection files, use the File Upload tool to put them on your Web server.

Immediately below the text, part of a dialogue box is shown with six tabs: session, emulation, applet, configuration, browser, and Web page.

However, it wasn't clear how to start whatever it is that displays those six tabs. I had the feeling that I was already seeing what I would get when I clicked on the Terminal Session left sidebar menu item. But I

clicked on the menu item anyway, having moved my mouse over it and seen a different URL listed:

file:/C:/Program Files/Reflection/html/tools/terminal_session.html

Clicking on the menu item did in fact reveal the same Web page I was viewing. On a whim, I clicked the Add button shown on the Session tab below and it worked! These are not screen shots of portions of a dialogue box, but the dialogue boxes themselves embedded into the Web page.

When I hit the Add button, an Add New Session dialogue box appeared with OK, Cancel, and Help buttons, and three fields:

- New session name
- Session type (default is HP)
- Enable secure connection (with empty check box).

The session name is arbitrary, and I chose the name of the company providing the test mainframe. The session type is selected from a drop-down list, of which IBM 3270 and IBM 3270 Printer were the only mainframe options. Help indicated that the secure connection option required that the security proxy server be installed and configured. After hitting the OK button, the session name showed, and was highlighted, in the large Session field on the Session tab. The value of the other two fields, session type and security status, were shown to the right.

I clicked on the Emulation tab, and the newly-created session's name appeared at the bottom as: Current session. There was a Save Sessions button to the right of it and a Help button right beside it. When I pushed Help, another browser window opened up with detailed information on each field. A good thing, too, as the first field, Host Name, had me remembering a similar field in some Windows-based 3270 emulators that you normally leave blank. In this case, it is either the numeric IP address (eg 161.87.23.1) or the domain name of the host. The only other field I changed was Transport, to tn3270E from the default tn3270. Excluding those that were greyed-out, the other fields were:

- Port = 23
- Automatically connect to host = check mark

- Model = Model 2 – 24x80 Extended
- TN association = blank (only used for print sessions).

CONFIGURE A SESSION

I hit the Save Sessions button and clicked the Applet tab, but nothing looked important enough to bother investigating. The Configuration tab displayed mostly greyed-out fields, but the Configure button in the Modify configuration section looked important and was described as: Open and configure a terminal session. When I hit the Configure button, a Reflection for Web window appeared that showed as initializing and got stuck at 85% until I hit the OK button on the dialogue box that told me that I was using a time-limited evaluation version of the software. My test mainframe then came up almost immediately, although the text was shifted right on the VTAM log-on screen by about 20 columns, and remained that way as I logged on.

When I maximized the window, it became clear what had happened. The default window was significantly wider than its vertical size, and the image was merely being centred. Maximized, it looks great, filling the window. The default 3270 Enter key is the PC <-' Enter key.

SUPPORTING BROWSERS

Back to the Terminal Session of the Deployment Director: the Browser tab allows you to specify which browsers will be running Reflection. For this Windows installer, the defaults are IE and Navigator/ Communicator 4.*x* on the Windows platform. I removed the check mark from Nav and suddenly many of the other choices were no longer greyed out. Pushing the Help button explains why:

> Because of changes in Netscape browser design, Netscape 6 requires a different web page to run Reflection sessions than older Netscape browsers.

Java Plug-in remained greyed-out as a choice of platforms for IE and Nav 4.*x*, but a Java Plug-in button sat alone in the Options section, not greyed out, though Help text indicated that it should be:

> If you select the Java Plug-in as your browser choice, use the Java

Plug-in button in the Settings section of this tab to configure settings for the version of the Java Plug-in you use.

THE SESSION WEB PAGE HTML

I pushed the Save Sessions button and clicked the Web Page tab. Just above View and Save Web Page buttons, the text reads:

> Review and save the HTML code that will start the Reflection session.
>
> Terminal session Web page
>
> This web page runs a terminal session. When security is enabled, the page requires the security archive that is cached locally by the web page saved in the section below.

When I hit the View button, a View Page Details: Terminal Session dialogue box appeared that listed the HTML for the Web page on the left and had two buttons on the right: Copy to Clipboard and Hide Comments. There were OK, Cancel, and Help buttons at the bottom of the dialogue box. I pushed the OK button, then the Save Web Page and Save Sessions buttons on the Web Page tab. The Copy to Clipboard button could have been used to add the HTML to an existing Web page.

FIGURING OUT FILE UPLOAD

The next step was to run the File Upload tool to store the newly-created Reflection files onto the Web server. It even gives you a direct link to it.

Clicking on the link displayed the File Upload component of Deployment Director on a Web page clearly designed for at least 1024x768 resolution; at 800x600, a horizontal scroll bar appears.

The text at the top of the page indicated that ftp was used to perform the upload. Further down the page there were two tips:

- For information about where to save files on the server, click the Help button, and then click the link to Upload Destinations for Reflection Files.

- For information about connecting to the Web server using ftp, go to the troubleshooting topic named File Upload Tool Cannot Connect to the Web Server.

I pushed the Help button, and a new Web browser window opened. I clicked on the Upload Destinations for Reflection Files link, and the newly-displayed Web page explained that everything depends on whether you have a Basic or Secure Terminal Session:

> The destination of the files you upload depends on the type of session you created. Use the information on this page to determine which files you should upload and the directory where they should be saved in the terminal emulation component installation on the web server.

The choice was clear once I recalled that I chose not to enable security through a proxy server. But the text also raised concerns that I had not yet run the installer for the terminal emulation component, especially when the Basic Terminal Session section had a one-row table with the following contents:

- File Type – Session web page (*.html), Configuration file (*.config)
- Default Local Location – <Java home folder>\reflectionweb_tle\upload\session\
- Target Server Location – <terminal emulation component>\session\

Just above was a section entitled About the Java Home Folder:

> By default, Reflection files are stored locally in the folder that the browser considers the user's Java home folder. For Microsoft Internet Explorer on Windows, the Java home folder might be C:\Winnt\Java or C:\Windows\Java; for Netscape Navigator or Communicator on Windows, the home folder might be C:\Netscape\Users\YourName; and for Internet Explorer on the Mac OS, the MRJ home folder might be Hard Drive:System Folder:Preferences.

Leaving Help and clicking the Connect button displayed a Connect to Server dialogue box with a single section entitled Login information

with four fields, all blank: server name, user name, password, and Reflection root directory.

I pushed the Help button, and the following description was provided to clarify the Reflection root directory field:

> Specifies the path to a home (default) directory for the installation of the terminal emulation component on your server. If you completed the Server tab in the Preferences dialogue box, the Reflection root directory that you entered is displayed in this box.
>
> When Reflection opens a connection to the server, it automatically sets the working directory to the specified root directory. The files and folders in the Reflection root directory appear in the Server files section of the File Upload window.

The Preferences dialogue box refers to the three tabs that you see when you hit the button to the right of the Connect button you pushed to get here. The Server tab sets the default values for server name, user name, and Reflection root directory.

At this point, several important points became clear:

- Now was the last possible moment to install the Reflection terminal emulation component.
- The documentation had glossed over some necessary ftp and Web server set-up.

In fairness, two points should be made:

- Much later, I did find instructions that told me to install the terminal emulation component immediately after the Administrative WebStation.
- The File Upload Tool does check for the presence of the terminal emulation components, which will identify most ftp and Web server set-up problems.

WHAT YOU NEED TO KNOW ABOUT WINDOWS 2000 SERVER

At this point, it became clear that I would need a good basic understanding of ftp and Web server set-up. Here are the most important points.

The default Windows 2000 ftp and Web sites do not take you to the same place. The default ftp site is intended for its traditional role of hosting files for Internet visitors to access via ftp, rather than Reflection's use of ftp as a way to upload HTML files to the Web site.

Although it's possible to directly install the terminal emulation component to a subdirectory on the default Web site (C:\Inetpub\Wwwroot), it probably makes more sense to install it (and the Administrative WebStation) elsewhere. You can then define a Virtual Directory that makes the folder where you installed it look like a subdirectory of the Web site and of the ftp site that must mirror the Web site.

The final important point is that the default ftp site is set up with the same user name and password that IIS was installed with: typically, Administrator. Version 5.0 of Internet Information Services (IIS) is the ftp and Web server that comes with Windows 2000 Server. Documentation is pre-installed on the Web site and even includes information not available through Help on the IIS menu bar. In Win 2000 Server, start up a Web browser and type a URL of http://localhost/iishelp

INSTALLING THE TERMINAL EMULATION COMPONENT

I went back to the folder where Reflection for the Web was downloaded, double-clicked rwebemulation.exe to start InstallShield Wizard, which extracts necessary files and then launches the wizard with the usual Welcome and Licence Agreement pages. As with the previous install, the Destination Folder was blank and I set it to C:\Program Files\Reflection as I did for the Administrative WebStation component, hoping that it wouldn't interfere. After the install, my only concern was the old uninstall.class file, which had been renamed uninstall.class.old. When I looked again at the readme.html file, it did seem to indicate that the two components were intended to be installed in different folders.

Rather than defining a new ftp site, it was a lot easier to change the definition of the default one so that it pointed to the default Web site. Start/Programs/Administrative Tools/Configure Your Server, click to expand Web/Media Server from the left sidebar, click Web Server, and then click on the Manage Web services link near the bottom of the text.

A window labelled Internet Information Services appeared. I clicked the plus sign ('+') beside the Web server name in the left Tree section, then right-mouse-clicked on Default ftp Site and selected Properties from the pop-up menu. In the Default ftp Site Properties dialogue box that appeared, I clicked on the Home Directory tab, changed the Local Path from c:\inetpub\ftproot to c:\inetpub\wwwroot, and hit the OK button.

This is the same place to define a Virtual Directory for the Web and ftp sites, so that the folder where Reflection was installed appeared as if it was a subdirectory of the Web and ftp roots. I right-clicked on Default Web Site and selected New/Virtual Directory from the pop-up menu that appeared. The Virtual Directory Creation Wizard appeared. I hit the Next button, entered the subdirectory name I wanted to use in the URL, hit Next, then hit Browse and found the folder where I installed the terminal emulator component, clicked on the folder, hit OK, hit Next, did not change the default Access Permissions (Read and Run scripts), hit Next, and then hit Finish.

I repeated this by right-clicking on Default ftp Site, but changed the Access Permissions to allow both Read and Write. I clicked the X in the upper-right corner of both windows (IIS and Configure Your Server) to close them.

UPLOADING THE SESSION FILE

To upload the session file, I went to Start/Programs/Reflection for the Web/Administrative WebStation and clicked the Upload sessions to a Web server link, and pushed the Connect button. You need to ensure that the fields read:

- Server name – localhost
- User name – administrator
- Password – the Windows password for the Administrator ID
- Reflection root directory – rweb or whatever Virtual directory you just created for ftp.

You can then push the Connect button. Note that the Connect button

will be greyed out until the Password field is filled in. If the password is blank (null), enter one or more blanks in the Password field.

Next, hit the Yes button, then the Disconnect button when it has completed and click the X in the upper-right corner to close the Administrative WebStation. Start up a Web browser on Windows 2000 Server and type a URL of:

http://localhost/rweb/session/NAME.html

(NAME will be whatever you defined in the New session name field when you first started the Administrative WebStation.)

Once you've OKed the security agreement and limited time evaluation dialogue box, you should see your host's VTAM log-on page.

I then moved to another workstation running off the same hub as the Windows 2000 Server machine, and started up a Web browser (again IE Version 6, but this time on Windows XP Professional. Instead of localhost, I typed the IP address of the Windows 2000 Server machine to form a URL like this:

http://192.168.111.2/rweb/session/NAME.html

On both machines, the results from a Web browser were identical: a full-function 3270 mainframe screen on an otherwise barren Web page. Only the session name is seen above the screen. This Web page, can be customized within the Administrative WebStation, as can almost everything else.

Noticeably absent was Reflection's menu bar. This is the default, but the Applet tab (discussed above) also allows:

- Terminal sessions to be run in their own window, complete with menu bar.
- Equivalent functionality provided via a shortcut menu.

This means that, by default, users cannot configure their own terminal sessions. But administrators can choose to allow some or all users to configure some or all settings.

Armand Minet
(Canada)

Information point – reviews

MAINFRAME WEEK – http://www.mainframeweek.com

As its name implies, there is a new issue of *Mainframe Week* every week, available on-line or delivered free by e-mail in one of several formats. It was launched by Xephon at the beginning of the year with:

- Mainframe news – product announcements
- A broad range of detailed technical articles
- Web site of the week
- Mainframe jobs – available employment and contracts.

It's intended to read like an electronic magazine, with a table of contents and pages you turn to read or glance over the publication. Many of the articles include links to code that you can cut and paste for your own use. Networking will be covered regularly in the technical articles, though not necessarily in every issue.

And it's worth noting that the mainframe news is hand-picked by real human beings. That's obviously not the case at another popular mainframe Web site, which recently highlighted stories about the British SAS (Special Air Service) regiment in Afghanistan, based on automated searches intended to catch SAS Institute and its SAS (Statistical Analysis System) products.

Jon E Pearkins
(Canada)

E-mail alerts

Our e-mail alert service will notify you when new issues of *TCP/SNA Update* have been placed on our Web site. If you'd like to sign up, go to http://www.xephon.com/tcpsna and click the 'Receive an e-mail alert' link.

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*, *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. Many have discovered that writing an article is not the daunting task that it might appear to be at first glance.

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

If you have ever experienced any difficulties, or made an interesting discovery, you could receive a cash payment, a free subscription to any of our *Updates*, or a credit against any of Xephon's wide range of products and services, simply by telling us all about it.

More information about contributing an article to a Xephon Update, and an explanation of the terms and conditions under which we publish articles, can be found at www.xephon.com/index/nfc/css/nfc.html. Alternatively, please write to the editor, Fiona Hewitt, at any of the addresses shown on page 2, or e-mail her at fionah@xephon.com

NewEra Software has announced the availability of IMAGE Focus 4.0 for OS/390 and z/OS image change management. IMAGE Focus tests, monitors, and documents changes made to system images, reducing the risk of system outages due to IPL failure.

Amongst other features, the JES2, JES3, VTAM, and TCP/IP Subsystem Inspectors are designed to allow those with a need to access only, for example, VTAM to define and test VTAM with no concern for the operations of JES2, JES3, or TCP/IP.

URL: http://www.newera.com

* * *

The new release of Software Diversified Services' Vital Signs/VM provides detailed analysis of TCP/IP traffic and performance on z/VM and VM/ESA systems.

Vital Signs/VM 4.3 monitors one or many TCP/IP stacks, and, amongst other things, allows system administrators to monitor important performance measurements, identify peaks, valleys, and trends in system usage, eliminate bottlenecks and I/O congestion, and improve device service time and balance resource utilization.

URL:http://www.sdsusa.com

* * *

IBM has made a number of announcements, including:

- Version 1.3 of Tivoli NetView Performance Manager for TCP/IP, its automated TCP/IP management application for zSeries systems and the connected IP network.
- Version 2.7 of Tivoli NetView Performance Monitor (NPM), which monitors, records, and reports network communication, performance, and utilization through both Java-based GUI and traditional 3270 SNA displays.
- CICS Transaction Server for z/OS Version 2.2 with a range of major enhancements, including improved CICS exploitation of TCP/IP services.
- DB2 Server for VSE & VM, Version 7 Release 2, with auto-restart of TCP/IP support.
- VSE/ESA Version 2 Release 6, offering expanded capabilities to create integrated solutions in a hybrid environment, including TCP/IP support for CICS External Call Interface (ECI).

URL: http://www.tivoli.com/products
URL: http://www.tivoli.com/products
URL: http://www.ibm.com/software/ts/cics
URL: http://www-4.ibm.com/software/
URL: http://www.ibm.com

* * *