



47

TCP/SNA

September 2002

In this issue

- 3 Demystifying today's Web services
 - 15 SNA-TCP network convergence
 - 24 SNA to IP migration without CICS-
APPC program modification
 - 39 Setting up a corporate portal –
getting started
 - 46 Accessing the mainframe with ftp
software products
 - 68 TCP/SNA news
-

© Xephon plc 2002

update

TCP/SNA Update

Published by

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

North American office

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

Subscriptions and back-issues

A year's subscription to *TCP/SNA Update*, comprising four quarterly issues, costs £130.00 in the UK; \$190.00 in the USA and Canada; £136.00 in Europe; £142.00 in Australasia and Japan; and £140.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the March 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/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.

© Xephon plc 2002. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.

Demystifying today's Web services

Web services would be much easier to come to grips with if they had a more descriptive name. The current name is ambiguous, misleading, and misused – so much so that ‘Web services’ may one day surpass SNA’s infamous ‘PU’ as the most misunderstood term in the computer industry. In just the same way that an SNA ‘Physical Unit’ was not a piece of hardware but instead a piece of software that controlled a hardware device, a Web service is not a service in the sense that most people think of computer, network, or Internet services. ‘Services’, in computer circles, means many things to many people, but invariably with a common underlying theme; it refers to any external entity that does work on your behalf. So people are used to print services, directory services, file services, security services, etc.

It has got to the point where in the context of enterprise networking there’s an overall connotation that services are provided by ‘middleware’ such as Java 2 Enterprise Edition (J2EE), IBM’s WebSphere Application Server, BEA WebLogic, IBM MQSeries, and Microsoft BizTalk Server.

But while Web services are indeed services in the formal sense of the term, and they are delivered across the Web, they aren’t ‘middleware’. Instead, they’re meant to represent high-level services – even modular applications. In reality, Web services are a new genre of Web applications. Start off by thinking of Web services in terms of: credit card authorization, international currency rate converter, stock quote provider, package delivery status locator, shipping rate calculator, local weather update, local traffic update, personalized horoscope, and user authentication – for example, with two-factor authentication.

The underlying rationale for Web services is to provide specific, self-contained (ie modular) functionality. It’s all about the availability of easily ‘pluggable’, reusable functionality, to expedite and simplify new application development. To be fair, however, since there are no restrictions or guidelines as to what Web services can or cannot do, anything is possible. Yes, you could most certainly have a print Web service (eg remote printing to a specialized printer), a directory Web

service (eg white pages or yellow pages), or even a file service. But Web services should not be confused with protocols or other system-enabling middleware ‘services’. In particular, Web services should not be thought of as referring to SOAP, WSDL, and UDDI.

SOAP, WSDL, UDDI, and the nascent Web Services Flow Language (WSFL) are the innovative and strategic technology used to make Web services possible. Though they are indeed middleware services, they are not themselves Web services; they’re enabling technologies for Web services. However, there is much confusion about this distinction in the media; out of three articles about Web services in trade journals, you might find two that talk exclusively about the enabling technologies while the third is actually talking about Web services as modular applications. It’s also not unusual to see headlines such as ‘Intel to support Web services’, where the services in question are the enabling technologies as opposed to the high-level, application-oriented services.

REMOTE PROCEDURE CALLS OF SORTS

Most simply stated, Web services are a platform- and programming language-independent way to call and obtain high-level functionality from other programs. They rely on a remote procedure call (RPC) mechanism in the true sense of that phrase (not to be confused with the heavily-used Unix RPC scheme that popularized this term). Web services were meant to be programmatic, as opposed to visually interactive, solutions – ie they were for program-to-program rather than for human-to-program consumption. Web services were going to provide new applications with the rich set of functionality that humans had been enjoying on the Web for so many years.

Take package tracking, often cited as a good prospect for a Web service, as an example. The likes of FedEx, UPS, Airborne, and even the US Postal Service provide on-line tracking systems on the Web. But these services are for human use. E-tailers that currently offer shipment tracking (eg Amazon) typically do so by providing a hot-link, usually with the tracking number embedded, to the Web page of the shipper in question. But it’s a two-stage operation: you do the shipment tracking outside the e-tailer’s actual e-commerce application.

If the shippers (eg FedEx) provided their tracking function as a Web service, the e-tailers would be able to plug it into their overall application as a built-in feature. The shippers shouldn't be averse to this, since the e-tailer's unlikely to hide or distort their identity, so the consumer would still know that the shipment they are awaiting would arrive via FedEx, UPS, or whoever. What the package tracking Web service does in this instance is provide function integration to ensure application completeness. This is the classic programmatic model for Web services.

Web services can play a pivotal role with regard to portals, whether it be corporate, public, or employee-only. More and more portal architectures are exploiting the concept of a 'portlet' – a content channel or an application 'window' within an overall portal view. The use of portlets makes the tie-in between Web services and portals that much more convenient and compelling. Web services can become the modular software 'engines' that power a particular portlet, either on a one-Web-service-to-one-portlet basis (eg stock price portlet), or a multiple-Web-services-to-one-portlet arrangement (eg 'your commute home' portlet that could combine information from three Web services offering information about local weather, traffic conditions, and radio station programming).

Given this 'gravitational' pull between Web services and portlets, it was inevitable that somebody would put forward the notion of Web services that came complete with their own presentation services. That would make the portlet integration that much simpler – though you're now back again to the visual, interactive paradigm. Well, IBM has taken the lead in defining 'Web services for remote portals' (WSRP) – also known as 'remote portlet Web services' – and has presented it as a potential standard to OASIS (www.oasis-open.org – a not-for-profit global consortium with more than 400 corporate and individual members in 100 countries around the world that drives the development, convergence, and adoption of e-business standards). The introduction of WSRP puts a big dent into the initial notion that Web services were meant to be programmatic. But that is the dynamic nature of all things related to the Web.

THE SCOOP ON WEB SERVICES

Web services are an XML-centric way of integrating modular programs over the Web using open, standardized interfaces that are decoupled from proprietary APIs. Their goal is to facilitate simple program-to-program programmatic interactions around the Web – though IBM's new WSRP initiative is attempting to extend the scope of Web services beyond just programmatic scenarios. Thanks to Web services, different companies will be able to easily interconnect software components to create sophisticated e-business applications. Web services are expected to become the predominant means by which companies will interact with each other on a programmatic basis. Hence their pivotal significance when it comes to corporate portals.

The key, defining attributes of a Web service are as follows:

- *Modular.* The notion of an 'object' that can be snapped together with other modular components.
- *Self-contained.* It needs to deliver the entire functionality it promises without other external dependencies.
- *Self-describing.* As per the XML precept of self-identification, Web services, using WSDL, describe their interfaces and what you need to do in order to 'bind' with them.
- *Self-advertising.* They can be easily located on the Web using UDDI.
- *XML-centric.*
- *Platform-independent.*
- *Programming language independent.*
- *Easily combined* with each other in mix-and-match mode, irrespective of platform and programming language differences.

Even more so than the platform- and programming language-independence, it is the self-advertising and self-description aspects that make Web services truly special. In much the same way that you could today use a search engine like Google to find Italian olive oil producers

or no-fee dating services, it will be possible to search for Web services that offer the exact service that you seek. This would be done when developing a new application or a portlet. When faced with the need for some self-contained, modular functionality, the first and automatic reaction should now be to see whether this functionality is available as a Web service. This would be realized using UDDI and a UDDI registry. It could be done manually by a human being or programmatically by an application. Although it's not essential to the Web services initiative, it's highly likely that down the road some search engines such as Google might include UDDI support so that these popular search engines may also be used to locate Web services.

When searching for a Web service to source, you don't have to limit the search to specific platforms, formats, APIs, or programming languages. Web services transcend all of these issues. Thanks to their inherent XML-enforced standardization, Web services are easy to integrate with each other to realize composite functionality. Rather like Lego blocks, they're meant to snap together with the minimum of effort.

This ability to easily obtain software functionality in the form of one or more Web services should positively influence all future discussions and decisions about new application development – particularly as it applies to corporate portals. It's important that you don't lose sight of this. Web services can dramatically compress schedules, further enhance the functionality of applications, increase the competitive element of applications, and even reduce cost.

PRICING, SECURITY, RELIABILITY, AND SCALABILITY

This isn't to say that there will be no cost associated with acquiring Web services. The pricing model for Web services is currently in its embryonic stage, with most people, quite rightly, still focusing their attention on the technical side of things – though IBM, a major shaker-and-mover when it comes to Web services as well as a company that knows a thing or two about complex, usage-based pricing models, has, to its credit, already started publishing position papers about possible accounting and metering schemes for Web services. What's abundantly clear is that there will be a very wide spectrum, ranging from freeware

to expensive, premium offerings coupled with many permutations, in the case of the non-freeware offerings, as to how the pricing will be structured.

The pricing options available will definitely include: one-time charge schemes, periodic licensing (eg monthly or yearly), and umpteenth usage-based options. In addition, it's likely that there could be third-party Web service distributors, though the inherent dynamic discovery capability of a Web service dilutes some of the potential value that can be offered by a distributor. In the case of Web services, the main value that a distributor could provide is that of handling the billing and collection. Doubtless there will also be syndicated Web services.

Whatever the pricing model, it will still not detract from the dynamic, 'pick-the-best-and-most-pertinent' service you need principle that is so fundamental to the whole rationale of Web services. It is this notion of dynamic discovery, self-description, and the XML-based, standardized invocation that truly distinguishes Web services from all the other object-oriented approaches such as Common Object Request Broker Architecture (CORBA) and Microsoft's Distributed Component Object Model (DCOM). While Web services will not displace these other technologies, Web services are poised to become the preferred and strategic means for adding modular functionality to new e-business and portal applications.

Security with regard to Web services is already attracting a lot of attention and scrutiny, given that there is vast potential for exposure and abuse. A 'malicious' Web service could do untold damage; it would be the perfect Trojan horse. So Web services need to be from trusted sources and authenticated, ideally with digital certificate technology. But then there's the flip-side, where, once integrated into an application, a Web service needs to perform security-related functions – eg authenticate users logging on to the application or encrypt data that is being transmitted.

This means that the Web services need to perform security-related functionality. In fact, obtaining such security-related functionality is a valuable and compelling use for Web services, and the issues are really no different from those involved in acquiring any other piece of

important software; the only caveat being that in the case of Web services you can obtain the software on the open market, so to speak, exploiting the dynamic discovery capabilities of UDDI.

But this is where common sense has to step in. It is probably not prudent to source a security-related Web service from an unknown company claiming to be located in Timbuktu. Despite all the self-describing and self-advertising ‘magic’, a Web service, cut to the chase, is just another software service, and you shouldn’t let the dynamic sourcing aspect cloud any judgement when it comes to validating, testing, and proving what you’re getting.

As with any other piece of software, Web services shouldn’t be immune from concerns about reliability, performance, scalability, and manageability – particularly in the context of portal applications. There is nothing in the Web services ‘standards’ that says that these software components have to be any different from other software vendors in the above areas! To assume that Web services will somehow miraculously break the mould when it comes to all the standard RAS issues (reliability, availability, and serviceability) that have hounded software for the last 40 years would be taking optimism to unheard of heights. A new Web service is not going to be that different from, say, a new release of Netscape for the Mac – rigorous and diligent testing is recommended to make sure there are no hidden surprises.

SUMMARY

Web services are not middleware services, as some portray them to be. Instead, they are high-level services akin to modular applications. Web Services are heavily XML-oriented, modular, self-contained, and self-describing. They can be published, readily located, and easily invoked across the Web. They now provide a standard, strategic and consistent way to package business processes and make them available over the Web to interested and authorized parties. XML permits data to be shared; Web Services permit business logic to be shared and reused.

While it will certainly be possible to dynamically source Web services across the Internet, not many are likely to be ‘freeware’. As with any

other piece of valuable software, there's likely to be a charge to cover development, maintenance, support, and future releases. The pricing model will, however, vary widely depending on the supplier as well as the functionality being sought. The dynamic sourcing aspect should also not cloud traditional concerns about security, reliability, performance, etc. Once dynamically sourced, a Web service, like any other piece of mission-critical software, should be researched, tested, and validated.

Anura Gurugé
Strategic Consultant (USA)

© Xephon 2002

TCP/IP fundamentals

The huge number of options that can be configured in TCP/IP can make it daunting to work with. This article describes exactly how TCP/IP works, providing a perfect starting point for TCP/IP novices as well as a useful memory-jogger for the more experienced.

Transmission Control Protocol/Internet Protocol (TCP/IP) has been around for over 20 years. Putting it very simply, TCP handles packet flow between systems, and IP handles the routing of the packets. TCP/IP is very important because it's the protocol the Internet uses.

Modern networks are designed using a layered approach, where each layer presents a predefined interface to the layer above it. This means that a modular design can be developed, to minimize problems in the development of new applications or adding new interfaces. TCP/IP has four layers:

- *Application layer.* The application layer includes SMTP (mail), telnet, FTP, Rlogin, NFS, NIS, and LPD.
- *Transport layer.* The transport layer uses two protocols, UDP and TCP. UDP (User Datagram Protocol) doesn't guarantee packet delivery, and applications using it must provide their own means of verifying delivery. TCP, by contrast, does guarantee delivery of packets to the applications that use it.

- *Network layer.* The network layer is concerned with packet routing, and uses low-level protocols such as ICMP, IP, and IGMP. In addition, routing protocols such as RIP, OSPF, and EGP are available.
- *Link layer.* The link layer is concerned with the actual transmittal of packets as well as with IP-to-Ethernet address translation.

TCP

TCP is a connection-oriented transport protocol that sends data as an unstructured stream of bytes. By using sequence numbers and acknowledgement messages, TCP can provide a sending node with delivery information about packets transmitted to a destination node. Where data has been lost in transit from source to destination, TCP can retransmit the data until a time-out condition is reached, or until successful delivery has been signalled. TCP can also recognize duplicate messages and will discard them appropriately. If the sending computer is transmitting too fast for the receiving computer, TCP can employ flow control mechanisms to slow data transfer. TCP can also communicate delivery information to the upper-layer protocols and applications it supports.

IP ADDRESSES

IP addresses are globally-unique, 32-bit numbers assigned by the Network Information Centre. Globally-unique addresses permit IP networks anywhere in the world to communicate with each other.

An IP address is divided into three parts: the first designates the network address, the second the subnet address, and the third the host address.

IP provides error reporting and fragmentation and reassembly of information units called datagrams for transmission over networks with different maximum data unit sizes.

The network number represents the network that contains a given IP address.

An IP address is made up of a network address (netid) and a host address

(hostid). With IP Version 4, the IP address is 32 bits long, the netid is 8 bits for class A, 16 bits for class B, and 24 bits for class C. The hostid occupies the rest of the 32 address bits. With IP Version 6, the address is 128 bits long, and the header is always 40 bytes long. IP6 overcomes many of the limitations of IP4.

A computer can distinguish between the netid and hostid because of something called the subnet mask. The subnet mask is located in a field directly below the IP address on the TCP/IP properties sheet. A simple subnet mask would be something like 255.255.0.0. The numbers that make up the subnet mask indicate which portion of the IP address is the netid and which portion is the hostid. The four numbers of the subnet mask correspond directly to the four numbers in the IP address. For example, if you had a computer with an IP address of 123.100.100.12 and a subnet mask of 255.255.0.0, the first two numbers in the subnet mask (both are 255) indicate that the first two numbers of the IP address are the netid. The second two numbers (both are 0) indicate that the second two numbers of the IP address are the hostid. Therefore, in the IP address 123.100.100.12, the 123.100 part denotes which network the computer is located on, and the 100.12 portion represents a particular computer on that network. Of course, subnet masks can become more complicated where an individual network is subdivided.

THE DEFAULT GATEWAY

How does a computer attach to a foreign network? The answer is by using routing tables. Routing tables are stored in routers, and tell the router where to connect in certain situations. Note that one of the TCP/IP configuration options is for a default gateway. The default gateway is the address where a TCP/IP packet is sent if TCP/IP can't locate the destination IP address on the local network. Usually, the default gateway points to the router that controls an office's link to the outside world (often through an Internet connection). Since this router has a table of other routers, it knows the address and location of these routers. The destination IP address helps the router determine to which router to pass the packet. Typically, a TCP/IP packet is passed through several routers before it arrives on the destination network and, finally, on the destination PC.

ROUTING

Routers within the Internet are organized hierarchically. Some routers are used to move information through one particular group of networks under the same administrative authority and control. (Such an entity is called an autonomous system.)

Routers used for information exchange within autonomous systems are called interior routers, and they use a variety of interior gateway protocols (IGPs) to accomplish this end.

Routers that move information between autonomous systems are called exterior routers; they use the Exterior Gateway Protocol (EGP) or Border Gateway Protocol (BGP).

The routing protocols used with IP are dynamic in nature. Dynamic routing requires the software in the routing devices to calculate routes. Dynamic routing algorithms adapt to changes in the network, and automatically select the best routes. Static routing, by contrast, calls for routes to be established by the network administrator. Static routes don't change until the network administrator changes them.

IP routing tables consist of destination address/next hop pairs. IP datagrams travel across an internetwork one router hop at a time – the entire route is not known at the beginning of the journey. Each node's involvement in the routing process consists only of forwarding packets based on internal information.

ICMP

The Internet Control Message Protocol (ICMP) reports routing failures back to the source, and also provides a method for testing node reachability across an Internet (the ICMP Echo and Reply messages). This is used to increase routing efficiency (the ICMP Redirect message). It also informs sources that a datagram has exceeded its allocated time to exist within an Internet (the ICMP Time Exceeded message).

RIP

Routing Information Protocol (RIP) is an interior routing protocol that works well in small environments but has serious limitations when used

in larger internetworks (eg limiting the number of router hops between any two hosts to 16).

IGRP

The Interior Gateway Routing Protocol (IGRP) is a routing protocol that determines the best path through an Internet by examining the bandwidth and delay of the networks between routers.

OSPF

OSPF is an intradomain, link state, hierarchical routing protocol. OSPF supports hierarchical routing within an autonomous system. Autonomous systems can be divided into routing areas. A routing area is typically a collection of one or more subnets that are closely related. All areas must connect to the backbone area.

INTEGRATED IS-IS

ISO 10589 (IS-IS) is an intradomain, link state, hierarchical routing protocol. Integrated IS-IS is an implementation of IS-IS for more than just OSI protocols, which supports both OSI and IP protocols.

EGP

Exterior Gateway Protocol (EGP) is an exterior routing protocol that provides dynamic connectivity but assumes that all autonomous systems are connected in a tree topology, which, unfortunately, is no longer true.

BGP

Like EGP, BGP is an interdomain routing protocol created for use in the Internet core routers. Unlike EGP, BGP was designed to prevent routing loops in arbitrary topologies and to allow policy-based route selection.

DHCP

One or more servers on a network can act as a Dynamic Host Configuration Protocol (DHCP) server. When a client computer that's

set to use the DHCP option comes online, the DHCP server automatically configures TCP/IP on that computer. This means that the DHCP server has to be configured, but it is not necessary to configure each client manually. DHCP can be very useful in an environment with a limited number of IP addresses to work with. Only the computers that are active at a given moment will use IP addresses.

WINS

On Windows-based networks, each computer has a name. The WINS option can be configured to contain the IP address of a Windows NT server that's running a WINS service. The WINS service resolves computer names (NetBIOS names) to IP addresses. If one PC user tries to access another computer by its name, the first computer will go directly to the WINS database and search for the name and IP address. It won't need to query every computer on the network to determine the right IP address. This greatly reduces potential network traffic.

DNS

DNS is similar to WINS, except that it manages domain names rather than computer names. A domain name is part of a Web address – xephon.com is a domain name. When a browser has www.xephon.com entered in the address line, the computer will ask a DNS server for the IP address for that domain name. If the DNS server doesn't know the address, it asks another DNS server. This process continues until finally a DNS server knows the address. Then, that server informs the user's computer, and all of the other DNS servers that were involved, of the domain name's IP address.

*Nick Nourse
Independent Consultant (UK)*

© Xephon 2002

SNA-TCP network convergence

As networks become more and more diverse, we find that we're running multiple networking protocols in both large and small networks. Although it was once normal to use only SNA-type networking, this is

no longer the case. LAN-based networking and increased intersite communication mean that TCP/IP has become the accepted protocol.

However, SNA professionals are generally facing the same issues: how to link SNA to these new systems and vice versa, and (as always) how to provide adequate performance.

There are both hardware and software options to facilitate the integration of SNA and TCP/IP networks. These range from cost-reducing options by allowing resource sharing, reducing duplication, all the way to eliminating one protocol altogether. There are many choices, including the following:

- Sharing links between two networks by using frame relay or ATM.
- Encapsulating TCP/IP over SNA using Data Link Switching.
- Concentrating protocols using software options.

Using a single backbone protocol allows you to remove the complexity of multiple protocol stacks. Having more than one protocol stack means that every packet in the network has to have different structures and be managed differently. Merging these protocols over one protocol makes the network more efficient and easier to manage. One solution is the AnyNet architecture, which, in essence, runs an application over a networking protocol that it was not specifically designed to run on. IBM has created many products to allow the co-existence of TCP/IP and SNA networks, including:

- Communications Server Sockets over SNA
- IBM eNetwork Host On-Demand
- Files on Demand
- Lotus Connect for SNA
- CICS Internet Gateway
- CICS Gateway for Java
- IBM Net.DATA.

There are several possible solutions for connecting TCP/IP workstations

to SNA applications on an SNA host. tn3270e functionality can be provided by the following products:

- IBM Communications Server for Windows NT 5.0
- IBM SNA Application and Client Access for AIX.

tn3270e will provide connectivity between the TCP/IP workstations and SNA hosts.

A Web server with Host On Demand and the tn3270e functions that are provided by Communications Server allow Java-enabled browsers connectivity to SNA applications. Host On Demand will allow users to access SNA applications with Web browsers without the need to install additional software. Communication between SNA applications can be transported over the TCP/IP network using the SNA over TCP/IP AnyNet functionality. Installing SNA over TCP/IP access node capabilities at end nodes means that the actual SNA applications will communicate over TCP/IP ports.

To allow communication for sockets-based applications, the Sockets over SNA AnyNet functions need to be implemented. Installing Sockets over SNA access node capabilities on end nodes means that socket applications, such as a Web Browser, can communicate over SNA ports.

IBM's Communications Server is a multifunction gateway allowing people to build heterogeneous networks. Protocols such as NetBIOS, SNA, IPX, TCP/IP, etc can be integrated and thus allow applications to be delivered to users with no concern for the underlying network protocol on which they have to run. Communications Server consists of many products that provide solutions on OS/390, AIX, OS/2, Windows, and NetWare platforms. It's also fully operable with OS/400 networks. You can get details of all IBM networking products from <http://www.networking.ibm.com>.

WHY USE THE SNA NETWORK?

Before we go on to look at the possible solutions for integrating the SNA and TCP/IP networks, it's worth asking: why bother using the SNA network at all? Well, apart from the fact that you've probably invested

a lot of money in it, you'll find that non-SNA applications running over SNA benefit from the SNA networking features. You get the following when the backbone is SNA:

- *Cost-effective bandwidth utilization and predictable response times.* Non-SNA applications benefit from a steady throughput and the predictable response time of an SNA network – achieved by using SNA's flow control prevention algorithms. IBM claims that tests of sockets applications on AIX, OS/2, OS/400, and MVS platforms show that they may outperform sockets applications running on a native TCP/IP network.
- *Traffic prioritization.* The configuration of Sockets over SNA combinations allows the association of class of service and priority for better known TCP/IP applications such as ftp and telnet. Telnet can be configured to have a higher priority than batch-based applications such as ftp.
- *Data compression.* This reduces the amount of data being exchanged between partners, improving the response times of the applications and data transfer rates.
- *Advanced Peer to Peer Networking (APPN) dynamics.* Many of the benefits of Advanced Peer to Peer Networking are extended by AnyNet, by allowing socket applications to communicate over Advanced Peer to Peer Networks. AnyNet increases the number of applications that can communicate on Advanced Peer to Peer Networks.
- *High Performance Routing.* HPR is an enhanced (but simpler) form of APPN. It's a very scalable architecture that supports non-disruptive session switching around failures.

ANYNET SOCKETS OVER SNA

The AnyNet suite of software chiefly consists of multi-protocol gateway nodes and multi-protocol access nodes based around the multi-protocol transport networking architecture (MPTN).

This protocol describes the formats, logical structures, operating principles, and protocols that allow applications using networks other

than the one on which they were chiefly written to function on a different type of network without having to change the existing application. The following are included in the multi-protocol transport networking implementations:

- SNA over IPX
- SNA/APPC over TCP/IP
- SNA over TCP/IP
- Sockets over IPX
- Sockets over NetBIOS
- Sockets over SNA
- IPX over TCP/IP
- IPX over SNA
- NetBIOS over TCP/IP
- NetBIOS over SNA.

If you're an applications developer, you'll know that networking application programming interfaces are tied to one network protocol. If you develop a program that uses a socket application programming interface, then it's tied to the TCP/IP protocol stack. If a program is developed using the CPI-C application programming interface, it would normally be tied to the SNA protocol stack. Multi-protocol transport networking removes this dependence from the application.

AnyNet uses `NODE TYPES` to implement this independence. There are two different types of node:

- The Access Node provides functions that allow the application program to use a network for which it was not originally developed. On any access node you can run a socket program that uses SNA to communicate with another socket program on another access node.
- The Gateway Node will connect two different networks, providing network protocol conversion between the two related network protocols.

A Native Node is defined in the Multi-protocol Network Transport architecture as a node that doesn't implement multi-protocol transport network functions. As an example, take a non-IBM Unix platform that has native sockets implemented on a standard TCP/IP stack. In order to use AnyNet functions, this host would have to route its IP packets through an AnyNet gateway node. AnyNet functionality is provided by a number of IBM products.

Functional overview of Sockets over SNA

The Sockets over SNA AnyNet option allows application programs written to the socket API standard to run over SNA network transport. You need to consider two things when considering the co-existence of TCP/IP and Sockets over SNA:

- Sockets over SNA and TCP/IP are identified by separate Internet protocol addresses. If you're using IP subnets, you must assign these separate IP addresses in different IP subnets.
- Sockets over SNA and TCP/IP operate independently of each other and do not have an awareness of each other.

When you use the AnyNet Sockets over SNA functions, the nodes that are involved have to map IP addresses to SNA logical unit names. If this is considered from the TCP/IP area, the SNA/APPN section of the network constitutes a separate IP network or sub-network. The Sockets over SNA gateway nodes act as normal IP routers routing packets across the two interfaces. One interface will be a standard TCP/IP interface and the other will be the AnyNet interface. For data sent to the AnyNet interface – normally known as `sna0` – the IP address will be mapped to an SNA logical unit name by Sockets over SNA and the data will be sent over LU 6.2 sessions that are established between the Sockets over SNA nodes. This address mapping occurs as:

- A socket application passes the IP address of the destination host to the Sockets over SNA.
- Sockets over SNA queries the IP-LU mapping table to translate the IP address to a fully-qualified logical unit name.
- Sockets over SNA allocates an LU 6.2 session to the fully-qualified

logical unit name which is the remote Sockets over SNA.

Sockets over SNA can either use an algorithm to map an IP address to an SNA fully-qualified logical unit name, or it can use an explicit approach. The difference is in the address mask used to perform the operation. A mask with all bits set, 255.255.255.255, will be treated as an explicit logical unit name. The algorithm approach will have a bit mask that doesn't have all bits set. The logical unit name is then created by combining the logical unit template with an algorithmically generated suffix based on bits of the destination IP address designated by the address mapping.

The table used to map IP-LU contains four columns: the IP address, address mask, SNA network ID, and logical unit name template. The IP-LU mapping table will be searched for the entry that matches the destination IP address and has the longest address mask. Care has to be taken when defining this area as you can end up generating incorrect names.

ENETWORK HOST ON-DEMAND

IBM has a small tn3270 emulator application known as the eNetwork Host On-Demand that operates as a Java applet on a Web Browser. Host On-Demand allows you to use a Web browser to access a 3270-based host application. It provides:

- Customized 3270 windows
- Multiple sessions
- Persistent connections
- Platform flexibility
- Security.

Users will connect to a Web server and will either get the Host On-Demand application by default or will request it by going to a specific link. This is then downloaded to the browser as a Java applet. It's a standard telnet 3270 application. This application then contacts the same server and connects to the tn3270e server of the IBM Communications Server.

Host On-Demand is available for the following products:

- IBM Communications Server for OS/2 Warp
- IBM Communications Server for AIX
- NetWare for SAA
- IBM Communications Server for Windows
- IBM Communications Server for MVS
- IBM TCP/IP for MVS
- OS/390.

You need to check which release of these products you're running, as Host On-Demand is available only on some later releases. Note that Host On-Demand isn't a replacement for IBM Personal Communications; it has limited functionality compared with that product, but it can be very useful in certain scenarios.

Host On-Demand functions as a standard 3270 application with a limited set of functions. Its limitations are:

- Sessions are limited to two sessions. When you open the Host On-Demand connection you have a SAME button to open a second session. You can only open two sessions like this. If, however, you start the Host On-Demand connection in a separate Window you can open more sessions by going back to the Web browser and specifying the Host On-Demand URL. This will get a third session started and you can then use the SAME button again.
- You're restricted to Java fonts.
- It has support for limited terminal types.

Most people use Windows NT workstations. To run Host On-Demand on a Windows NT platform you need a Web server such as IBM's Internet Connection Server for Windows NT, Microsoft's Internet Information Server, or Lotus Domino. The Web server has to be running on the same machine as Host On-Demand, and cannot be split across multiple machines.

Host On-Demand uses the tn3270e functions of Communications Server to establish a logical unit session with the mainframe host. The tn3270e function provides the gateway between the TCP/IP workstations and the SNA host. You have to configure this by defining a local node and a Control Point definition back to the host. You also need Gateway definition and a pool of logical units, as well as definitions for the logical units on the host computer.

All of this definition can be found in the relevant IBM manuals or in some of the excellent Red Books that can be found on the Web at:

<http://www.redbooks.ibm.com>

CONCLUSION

It is possible to use existing investment in SNA networks to Web-enable legacy applications without major redevelopment. It's debatable whether or not this will be a long-term solution, but it can be a cost effective one in today's struggling economies.

In the future, as more and more networks see SNA being removed, there will probably be no need for this type of connection other than in very large businesses that will find it impractical to remove their legacy applications.

John Bradley
Systems Programmer (UK)

© Xephon 2002

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.

SNA to IP migration without CICS-APPC program modification

We have a large company network with the OS/390 mainframe in the centre and thousands of clients that use mainframe databases. Our business logic programs are under CICS, and communication with clients is based on APPC (LU6.2) protocol. In the first stage of SNA/IP integration, we used standard encapsulation technology Data Link Switching (DLSw), which enables routers to put SNA messages into IP packets.

When looking at ways of cutting IT costs, we considered purchasing cheap routers without the additional software that enables DLSw. One possibility was to migrate applications from APPC to CICS TCP/IP Sockets Interface, but we didn't want to write and test hundreds of code lines just to end up with functionality similar to what we already had.

The other possibility was to install CICS Transaction Gateway, which would give us SNA over IP communication with Sync level 2. This solution would cost extra money for the CTG licence, and would require modification of all the CICS and client programs.

In the end, we settled on the solution presented here. It suited us because:

- It enabled us to reduce our costs.
- It needed only minimal modification of programs on the client side.

The only software prerequisite is Personal Communication feature AnyNet, which was already present. We added a standard prologue and epilogue to all client programs, as you can see in the `initsess.c` program. The CICS programs didn't need any changes at all.

In the following example, the client's parameters are passed as the arguments of the program. In our production environment, the parameters that define all network clients are placed in the SQL server database. This approach enables centralized administration of the parameters that describe APPC communication.

CLIENT PROGRAM

```
/*-----*/
/*                                             */
/*          Function - quest  HOST          */
/*                                             */
/*-----*/
#include <windows.h>
HINSTANCE hInst;
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <winappc.h>
#include <wincsv.h>
#include <winnof.h>
#include <process.h>
#include "transld.c"

#define clear_vcb()  memset(vcbptr,(int)'\0',sizeof(*vcbptr))

FILE *fm;
char lu_name|9|="          ";          /* local lu name          */
char lu_alias|9|="          ";          /* local lu alias        */
char plu_name|9|="          ";          /* partner lu name       */
char plu_alias|9|="          ";          /* partner lu alias      */
char mode_name|9|="          ";
char fqp_lu_name|18|="          ";      /* fully qualified partner */
char fq_lu_name|18|="          ";       /* fully qualified        */
char tp_name|5|="          ";
char sna_suffix|21|="          ";

unsigned short appc_rc_p;
unsigned long appc_rc_s;
char tp_id|8|;
unsigned long conv_id;

unsigned char send_buffer|2000|,receive_buffer|2000|;

short length_of_send_buffer;
short max_leng_of_receive_buffer;
int how_much_received;
int init_node;
int i;

typedef union{
    struct tp_started          tpstart;
    struct mc_allocate        alloc;
    struct mc_send_data       snd_data;
    struct mc_flush           flu;
}
```

```

        struct mc_receive_and_wait      rcv_and_wait;
        struct mc_deallocate            dealloc;
        struct tp_ended                  tpend;
        struct initialize_session_limit  initsess;
        struct change_session_limit     changesess;
        struct query_node                qnode;

    } vcbtype;
vcbtype *vcbptr,vcb;

void prog_end(void);
void do_build_acg(void);
void do_tp_started(void);
void do_mc_allocate(void);
void do_mc_send(void);
void do_mc_flush(void);
void do_mc_rcv_wait(void);
void do_mc_deallocate(void);
void do_tp_ended(void);
void do_initialize_session_limit(void);
void do_change_session_limit(void);
void do_query_node(void);

/*****
/*          QUERY HOST          */
*****/
main(int argc, char *argv||)
{

//    checking arguments passed to program

if(argc!=10)
{
    printf("ERROR - Wrong number of arguments!!!\n");
    exit(5);
}

i=strlen(argv|1|);
if (i>8)
{
printf("\n ERROR - Wrong length of lu_name\n");
exit(5);
}
strncpy(lu_name,argv|1|,8);

i=strlen(argv|2|);
if (i>8)
{
printf("\n ERROR - Wrong length of lu_alias\n");

```

```

exit(5);
}
strncpy(plu_alias,argv|2|,i);

i=strlen(argv|3|);
if (i>8)
{
printf("\n ERROR - Wrong length of plu_name\n");
exit(5);
}
strncpy(plu_name,argv|3|,i);

i=strlen(argv|4|);
if (i>8)
{
printf("\n ERROR - Wrong length of plu_alias\n");
exit(5);
}
strncpy(plu_alias,argv|4|,i);

i=strlen(argv|5|);
if (i>8)
{
printf("\n ERROR - Wrong length of mode_name\n");
exit(5);
}
strncpy(mode_name,argv|5|,i);

i=strlen(argv|6|);
if (i>17)
{
printf("\n ERROR - Wrong length of fq_lu_name\n");
exit(5);
}
strncpy(fq_lu_name,argv|6|,i);

i=strlen(argv|7|);
if (i>17)
{
printf("\n ERROR - Wrong length of fqp_lu_name\n");
exit(5);
}
strncpy(fqp_lu_name,argv|7|,i);

i=strlen(argv|8|);
if (i>4)
{
printf("\n ERROR - Wrong length of tp_name\n");
exit(5);
}

```

```

}
strncpy(tp_name,argv|8|,i);

i=strlen(argv|9|);
if (i>20)
{
printf("\n ERROR - Wrong length of sna_suffix\n");
exit(5);
}
strncpy(sna_suffix,argv|9|,i);

printf("\n LU_NAME=%s",lu_name);
printf("\n LU_ALIAS=%s",lu_alias);
printf("\n PLU_NAME=%s",plu_name);
printf("\n PLU_ALIAS=%s",plu_alias);
printf("\n MODE_NAME=%s",mode_name);
printf("\n FQ_LU_NAME=%s",fq_lu_name);
printf("\n FQP_LU_NAME=%s",fqp_lu_name);
printf("\n TP_NAME=%s",tp_name);
printf("\n SNA_SUFFIX=%s\n",sna_suffix);

max_leng_of_receive_buffer=2000;
/* ***** */
/*          PROLOGUE - contains functions:          */
/*          - do_query_node()                        */
/*          - do_build_acg()                          */
/*          - do_initialize_session_limit();          */
/* ***** */

vcbptr=&vcb;
do_query_node();

if(appc_rc_pl=AP_OK)
{

do_build_acg();

init_node=system("csstart pc50");
if(init_node!=0)
{
printf("\n Function system=%lx",init_node);
printf("\n INITIALIZATION OF NODE NOT OK. !!!");
printf("\n CALL FOR HELP");
exit(10);
}
}

vcbptr=&vcb;
do_initialize_session_limit();

```

```

if(appc_rc_pl==AP_OK)
{
    printf("\n Primary rc=%x",appc_rc_p);
    printf("\n Secondary rc=%lx",appc_rc_s);

    appc_rc_s=vcbptr->initssess.sense_data;
    printf("\n Sense data=%lx",appc_rc_s);

    printf("\n SESSION LIMIT NOT OK. !!!");
    printf("\n CALL FOR HELP");
    exit(20);
}
/* ***** */
/*          END OF PROLOGUE          */
/* ***** */
vcbptr=&vcb;
do_tp_started();

if(appc_rc_pl==AP_OK)
{
    printf("\n Primary rc=%x",appc_rc_p);
    printf("\n Secondary rc=%lx",appc_rc_s);
    printf("\n TP_STARTED NOT OK. !!!");

    exit(30);
}

vcbptr=&vcb;
do_mc_allocate();

if(appc_rc_pl==AP_OK)
{
    printf("\n Primary rc=%x",appc_rc_p);
    printf("\n Secondary rc=%lx",appc_rc_s);
    printf("\n MC_ALLOCATE NOT OK.!!!");
    exit(40);
}

do_mc_flush();

strcpy(send_buffer,"          DATA SENDING TO HOST          \0");
length_of_send_buffer=strlen(send_buffer);
do_mc_send();

if(appc_rc_pl==AP_OK)
{
    printf("\n Primary rc=%x",appc_rc_p);
    printf("\n Secondary rc=%lx",appc_rc_s);
}

```

```

        printf("\n SEND DATA NOT OK. !!!");
        prog_end();
        exit(4);
    }

do_mc_rcv_wait();
if(appc_rc_pl=AP_OK)
{
    printf("\n Primary rc=%x",appc_rc_p);
    printf("\n Secondary rc=%lx",appc_rc_s);
    printf("\n RECEIVE AND WAIT NOT OK. !!!");
    prog_end();
    exit(4);
}

    printf("\nhost replay : %s",receive_buffer);

prog_end();
exit(0);
}
/***** FUNCTION PROG_END *****/
void prog_end()
{
do_mc_deallocate();
if(appc_rc_pl=AP_OK)
    printf("\n MC_DEALLOCATE NOT OK. !!!");
do_tp_ended();
if(appc_rc_pl=AP_OK)
    printf("\n TP_ENDED NOT OK. !!!");

/* ***** */
/*                               EPILOGUE                               */
/* ***** */
do_change_session_limit();
if(appc_rc_pl=AP_OK)
{
    printf("\n Primary rc=%x",appc_rc_p);
    printf("\n Secondary rc=%lx",appc_rc_s);
    appc_rc_s=vcbptr->changesess.sense_data;
    printf("\n Sense data=%lx",appc_rc_s);
    printf("\n CHANGE_SESSION_LIMIT NOT OK. !!!");
}
else
    printf("\n SESSION DEACTIVATED !!!");
}
/* ***** */
/*                               END OF EPILOGUE                               */
/* ***** */

/***** QUERY NODE *****/

```

```

void do_query_node()
{
    clear_vcb();
    vcbptr->qnode.opcode=AP_QUERY_NODE;
    vcbptr->qnode.format=0;

    WinNOF((long)(vcbptr),(unsigned short)sizeof(vcbptr->qnode));
    appc_rc_p=vcbptr->qnode.primary_rc;
    appc_rc_s=vcbptr->qnode.secondary_rc;
}

/***** INITIALIZE SESSION LIMIT *****/
void do_initialize_session_limit()
{
    clear_vcb();
    vcbptr->initsess.opcode=AP_INITIALIZE_SESSION_LIMIT;
    vcbptr->initsess.format=0;

    memcpy(vcbptr->initsess.lu_name,lu_name,8);
    TranslD(vcbptr->initsess.lu_name,8,0x20);
    memcpy(vcbptr->initsess.lu_alias,lu_alias,8);
    memcpy(vcbptr->initsess.plu_alias,plu_alias,8);
    memcpy(vcbptr->initsess.fqplu_name,fqp_lu_name,17);
    TranslD(vcbptr->initsess.fqplu_name,17,0x20);
    memcpy(vcbptr->initsess.mode_name,mode_name,8);
    TranslD(vcbptr->initsess.mode_name,8,0x20);
    vcbptr->initsess.set_negotiable=AP_NO;
    vcbptr->initsess.plu_mode_session_limit=1;
    vcbptr->initsess.min_conwinners_source=1;
    vcbptr->initsess.min_conwinners_target=0;
    vcbptr->initsess.auto_act=1;

    WinNOF((long)(vcbptr),(unsigned short)sizeof(vcbptr->initsess));
    appc_rc_p=vcbptr->initsess.primary_rc;
    appc_rc_s=vcbptr->initsess.secondary_rc;
}

/***** CHANGE SESSION LIMIT *****/
void do_change_session_limit()
{
    clear_vcb();
    vcbptr->changesess.opcode=AP_CHANGE_SESSION_LIMIT;
    vcbptr->changesess.format=0;

    memcpy(vcbptr->changesess.lu_name,lu_name,8);
    TranslD(vcbptr->changesess.lu_name,8,0x20);
    memcpy(vcbptr->changesess.lu_alias,lu_alias,8);
    memcpy(vcbptr->changesess.plu_alias,plu_alias,8);
    memcpy(vcbptr->changesess.fqplu_name,fqp_lu_name,8);
}

```

```

Trans1D(vcbptr->changesess.fqplu_name,8,0x20);
memcpy(vcbptr->changesess.mode_name,mode_name,8);
Trans1D(vcbptr->changesess.mode_name,8,0x20);
vcbptr->changesess.set_negotiable=AP_NO;
vcbptr->changesess.plu_mode_session_limit=0;
vcbptr->changesess.min_conwinners_source=0;
vcbptr->changesess.min_conwinners_target=0;
vcbptr->changesess.auto_act=0;
vcbptr->changesess.responsible=AP_SOURCE;
WinNOF((long)(vcbptr),(unsigned short)sizeof(vcbptr->initsess));
appc_rc_p=vcbptr->changesess.primary_rc;
appc_rc_s=vcbptr->changesess.secondary_rc;

/* appc_rc_s=vcbptr->changesess.sense_data;
   printf("\n Sense data=%lx",appc_rc_s); */
}
/***** TP_STARTED *****/
void do_tp_started()
{
    clear_vcb();
    vcbptr->tpstart.opcode=AP_TP_STARTED;
    memcpy(vcbptr->tpstart.lu_alias,lu_alias,8);
    memcpy(vcbptr->tpstart.tp_name,tp_name,4);
    Trans1D(vcbptr->tpstart.tp_name,4,0x20);
    APPC((long)(vcbptr));
    appc_rc_p=vcbptr->tpstart.primary_rc;
    appc_rc_s=vcbptr->tpstart.secondary_rc;
    memcpy(tp_id,vcbptr->tpstart.tp_id,8);
}

/***** MC_ALLOCATE *****/
void do_mc_allocate()
{
    clear_vcb();
    vcbptr->alloc.opcode=AP_M_ALLOCATE;
    vcbptr->alloc.opext=AP_MAPPED_CONVERSATION;
    memcpy(vcbptr->alloc.tp_id,tp_id,8);
    vcbptr->alloc.sync_level=AP_CONFIRM_SYNC_LEVEL;
    vcbptr->alloc.rtn_ctl=AP_WHEN_SESSION_ALLOCATED;
    vcbptr->alloc.security=AP_NONE;
    vcbptr->alloc.pip_dlen=0;
    memcpy(vcbptr->alloc.plu_alias,plu_alias,8);
    memcpy(vcbptr->alloc.tp_name,tp_name,4);
    Trans1D(vcbptr->alloc.tp_name,4,0x20);
    memcpy(vcbptr->alloc.mode_name,mode_name,8);
    Trans1D(vcbptr->alloc.mode_name,8,0x20);
    APPC((long)(vcbptr));
    appc_rc_p=vcbptr->alloc.primary_rc;
    appc_rc_s=vcbptr->alloc.secondary_rc;
}

```



```

    if(appc_rc_p==AP_OK)
        conv_id=vcbptr->alloc.conv_id;
}

/***** MC_SEND *****/
void do_mc_send()
{
    clear_vcb();
    vcbptr->snd_data.opcode=AP_M_SEND_DATA;
    vcbptr->snd_data.opext=AP_MAPPED_CONVERSATION;
    vcbptr->snd_data.conv_id=conv_id;
    memcpy(vcbptr->snd_data.tp_id,tp_id,8);
    vcbptr->snd_data.dlen=length_of_send_buffer;
    TransID(send_buffer,length_of_send_buffer,0x20);
    vcbptr->snd_data.dptr=&send_buffer|0;
    APPC((long)(vcbptr));
    appc_rc_p=vcbptr->snd_data.primary_rc;
    appc_rc_s=vcbptr->snd_data.secondary_rc;
}

/***** MC_FLUSH *****/
void do_mc_flush()
{
    clear_vcb();
    vcbptr->flu.opcode=AP_M_FLUSH;
    vcbptr->flu.opext=AP_MAPPED_CONVERSATION;
    vcbptr->flu.conv_id=conv_id;
    memcpy(vcbptr->flu.tp_id,tp_id,8);
    APPC((long)(vcbptr));
    appc_rc_p=vcbptr->flu.primary_rc;
    appc_rc_s=vcbptr->flu.secondary_rc;
}

/***** MC_RCV_WAIT *****/
void do_mc_rcv_wait()
{
    memset(receive_buffer,' ',max_leng_of_receive_buffer);
    clear_vcb();
    vcbptr->rcv_and_wait.opcode=AP_M_RECEIVE_AND_WAIT;
    vcbptr->rcv_and_wait.opext=AP_MAPPED_CONVERSATION;
    vcbptr->rcv_and_wait.conv_id=conv_id;
    memcpy(vcbptr->rcv_and_wait.tp_id,tp_id,8);
    vcbptr->rcv_and_wait.max_len=max_leng_of_receive_buffer;
    vcbptr->rcv_and_wait.dptr=&receive_buffer|0;
    APPC((long)(vcbptr));
    appc_rc_p=vcbptr->rcv_and_wait.primary_rc;
    appc_rc_s=vcbptr->rcv_and_wait.secondary_rc;
    how_much_received=vcbptr->rcv_and_wait.dlen;
    TransID(receive_buffer,how_much_received,0x40);
}

```

```

    receive_buffer|how_much_received|=0;
}

/***** MC_DEALLOCATE *****/
void do_mc_deallocate()
{
    clear_vcb();
    vcbptr->dealloc.opcode=AP_M_DEALLOCATE;
    vcbptr->dealloc.opext=AP_MAPPED_CONVERSATION;
    vcbptr->dealloc.conv_id=conv_id;
    memcpy(vcbptr->dealloc.tp_id,tp_id,8);
    vcbptr->dealloc.dealloc_type=AP_ABEND;
    APPC((long)(vcbptr));
    appc_rc_p=vcbptr->dealloc.primary_rc;
    appc_rc_s=vcbptr->dealloc.secondary_rc;
}

/***** TP_ENDED *****/
void do_tp_ended()
{
    clear_vcb();
    vcbptr->tpend.opcode=AP_TP_ENDED;
    memcpy(vcbptr->tpend.tp_id,tp_id,8);
    APPC((long)(vcbptr));
    appc_rc_p=vcbptr->tpend.primary_rc;
    appc_rc_s=vcbptr->tpend.secondary_rc;
}

/***** BUILD_ACG *****/
void do_build_acg()
{
    fm=fopen("C:\\program files\\personal
communications\\private\\pc50.acg","w");

    fprintf(fm,"%s\n","NODE=");
    fprintf(fm,"%s\n","    ANYNET_SUPPORT=ACCESS_NODE");
    fprintf(fm,"%s","    CP_ALIAS=");
    fprintf(fm,"%s\n",lu_name);
    fprintf(fm,"%s\n","    DEFAULT_PREFERENCE=NONNATIVE");
    fprintf(fm,"%s\n","    DISCOVERY_SUPPORT=NO");
    fprintf(fm,"%s\n","    DLUR_SUPPORT=MULTI_SUBNET");
    fprintf(fm,"%s","    FQ_CP_NAME=");
    fprintf(fm,"%s\n",fq_lu_name);
    fprintf(fm,"%s\n","    NODE_ID=05D00000");
    fprintf(fm,"%s\n","    NODE_TYPE=END_NODE");
    fprintf(fm,"%s\n","    REGISTER_WITH_CDS=1");
    fprintf(fm,"%s\n","    REGISTER_WITH_NN=1");
    fprintf(fm,"%s\n","");
}

```

```

fprintf(fm,"%s\n","PORT=(");
fprintf(fm,"%s\n","      PORT_NAME=ANYNET");
fprintf(fm,"%s\n","      ACTIVATION_DELAY_TIMER=30");
fprintf(fm,"%s\n","      DLC_NAME=ANYNET");
fprintf(fm,"%s\n","      IMPLICIT_CP_CP_SESS_SUPPORT=0");
fprintf(fm,"%s\n","      IMPLICIT_DEACT_TIMER=0");
fprintf(fm,"%s\n","      IMPLICIT_HPR_SUPPORT=0");
fprintf(fm,"%s\n","      IMPLICIT_LIMITED_RESOURCE=NO");
fprintf(fm,"%s\n","      IMPLICIT_LINK_LVL_ERROR=0");
fprintf(fm,"%s\n","      LINK_STATION_ROLE=NEGOTIABLE");
fprintf(fm,"%s\n","      MAX_ACTIVATION_ATTEMPTS=10");
fprintf(fm,"%s\n","      MAX_IFRM_RCVD=127");
fprintf(fm,"%s\n","      MAX_RCV_BTU_SIZE=9216");
fprintf(fm,"%s\n","      PORT_TYPE=SATF");
fprintf(fm,"%s\n","      RETRY_LINK_ON_DISCONNECT=1");
fprintf(fm,"%s\n","      RETRY_LINK_ON_FAILED_START=1");
fprintf(fm,"%s\n","      RETRY_LINK_ON_FAILURE=1");
fprintf(fm,"%s\n","    ");
fprintf(fm,"%s\n","MODE=(");
fprintf(fm,"%s\n","      MODE_NAME=SNASVCMG");
fprintf(fm,"%s\n","      AUTO_ACT=0");
fprintf(fm,"%s\n","      COMPRESSION=PROHIBITED");
fprintf(fm,"%s\n","      COS_NAME=SNASVCMG");
fprintf(fm,"%s\n","      DEFAULT_RU_SIZE=0");
fprintf(fm,"%s\n","      MAX_NEGOTIABLE_SESSION_LIMIT=2");
fprintf(fm,"%s\n","      MAX_RU_SIZE_UPPER_BOUND=512");
fprintf(fm,"%s\n","      MIN_CONWINNERS_SOURCE=1");
fprintf(fm,"%s\n","      PLU_MODE_SESSION_LIMIT=2");
fprintf(fm,"%s\n","      RECEIVE_PACING_WINDOW=1");
fprintf(fm,"%s\n","    ");
fprintf(fm,"%s\n","PARTNER_LU=(");
fprintf(fm,"%s","      FQ_PLU_NAME=");
fprintf(fm,"%s\n",fqplu_name);
fprintf(fm,"%s\n","      CONV_SECURITY_VERIFICATION=1");
fprintf(fm,"%s\n","      MAX_MC_LL_SEND_SIZE=32767");
fprintf(fm,"%s\n","      PARALLEL_SESSION_SUPPORT=0");
fprintf(fm,"%s","      PARTNER_LU_ALIAS=");
fprintf(fm,"%s\n",plu_alias);
fprintf(fm,"%s\n","      PREFERENCE=NONNATIVE_THEN_NATIVE");
fprintf(fm,"%s\n","    ");
fprintf(fm,"%s\n","ANYNET_COMMON_PARAMETERS=(");
fprintf(fm,"%s\n","      CONNWAIT_SECS=30");
fprintf(fm,"%s\n","      CONN_RETRY_SECS=300");
fprintf(fm,"%s\n","      INACTIVITY_TIMER_SECS=30");
fprintf(fm,"%s","      SNASUFFIX=");
fprintf(fm,"%s\n",sna_suffix);
fprintf(fm,"%s\n","      SNA_IP_NODE_TYPE=1");
fprintf(fm,"%s\n","      UNACKED_DG_RETRY_SECS=10");
fprintf(fm,"%s\n","      UNSENT_DG_RETRY_SECS=3");

```

```

    fprintf(fm,"%s\n",");");
    fprintf(fm,"%s\n","SHARED_FOLDERS=(");
    fprintf(fm,"%s\n","    CACHE_SIZE=256");
    fprintf(fm,"%s\n",");");
    fprintf(fm,"%s\n","VERIFY=(");
    fprintf(fm,"%s\n","    CFG_MODIFICATION_LEVEL=13");
    fprintf(fm,"%s\n","    CFG_VERSION_LEVEL=1");
    fprintf(fm,"%s\n",");");

    fclose(fm);
}

```

SUBROUTINE FOR CONVERSION FROM ASCII TO EBCDIC AND VICE VERSA

```

#include <stdio.h>
#define USI unsigned short int
#define ULI unsigned long int
#define UC unsigned char
/*****
    Translate tables in that module specify translation from EBCDIC to
    ASCII and vice versa according to our national standard.
*****/

UC TrT_Ebcdic_Ascii|256| = {
/*---- TRANSLATION TABLE EBCDIC ==> ASCII -----*/
/*----  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F */
/*0*/"\x00\x01\x02\x03\xDF\x09\xBA\xAA\xAE\xC3\xC4\x0B\x0C\x0D\x0E\x0F"
/*1*/"\x10\x11\x12\x13\xAF\xB9\x08\xBB\x18\x19\xCA\xC5\x1C\x1D\x1E\x1F"
/*2*/"\xB0\xB1\xB2\xB3\xB4\x0A\x17\x1B\xBC\xBF\xC0\xC1\xC2\x05\x06\x07"
/*3*/"\xC8\xC9\x16\xCB\xCC\xCD\xCE\x04\xD9\xDA\xDB\xDC\x14\x15\xFE\x1A"
/*4*/"\x20\xFF\x83\x84\xEE\xA0\xC7\x9F\x87\x86\x5B\x2E\x3C\x28\x2B\x21"
/*5*/"\x26\x82\xA9\x89\x85\xA1\x8C\x96\x92\xE1\x5D\x24\x2A\x29\x3B\x5E"
/*6*/"\x2D\x2F\xB6\x8E\xF1\xB5\xC6\xAC\x80\x8F\x7C\x2C\x25\x5F\x3E\x3F"
/*7*/"\xF3\x90\xA8\xD3\xDE\xD6\xD7\x95\x91\x60\x3A\x23\x40\x27\x3D\x22"
/*8*/"\xF4\x61\x62\x63\x64\x65\x66\x67\x68\x69\x98\xE5\xD0\xEC\xFD\xAD"
/*9*/"\xF8\x6A\x6B\x6C\x6D\x6E\x6F\x70\x71\x72\x88\xE4\xE7\xF7\xF2\xCF"
/*A*/"\xA5\x7E\x73\x74\x75\x76\x77\x78\x79\x7A\x97\xD5\xD1\xED\xFC\xB8"
/*B*/"\xFA\xA4\xBE\xDD\xBD\xF5\xA7\xAB\xA6\x8D\x9D\xE3\xE6\xF9\xEF\x9E"
/*C*/"\x7B\x41\x42\x43\x44\x45\x46\x47\x48\x49\xF0\x93\x94\xEA\xA2\x8B"
/*D*/"\x7D\x4A\x4B\x4C\x4D\x4E\x4F\x50\x51\x52\xB7\xFB\x81\x9C\xA3\xD8"
/*E*/"\x5C\xF6\x53\x54\x55\x56\x57\x58\x59\x5A\xD4\xE2\x99\xE8\xE0\x8A"
/*F*/"\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\xD2\xEB\x9A\x9B\xE9\x7F"
};

UC TrT_Ascii_Ebcdic|256| = {
/*---- TRANSLATION TABLE ASCII ==> EBCDIC -----*/
/*----  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F */

```

```

/*0*/"\x00\x01\x02\x03\x37\x2D\x2E\x2F\x16\x05\x25\x0B\x0C\x0D\x0E\x0F"
/*1*/"\x10\x11\x12\x13\x3C\x3D\x32\x26\x18\x19\x3F\x27\x1C\x1D\x1E\x1F"
/*2*/"\x40\x4F\x7F\x7B\x5B\x6C\x50\x7D\x4D\x5D\x5C\x4E\x6B\x60\x4B\x61"
/*3*/"\xF0\xF1\xF2\xF3\xF4\xF5\xF6\xF7\xF8\xF9\x7A\x5E\x4C\x7E\x6E\x6F"
/*4*/"\x7C\xC1\xC2\xC3\xC4\xC5\xC6\xC7\xC8\xC9\xD1\xD2\xD3\xD4\xD5\xD6"
/*5*/"\xD7\xD8\xD9\xE2\xE3\xE4\xE5\xE6\xE7\xE8\xE9\x4A\xE0\x5A\x5F\x6D"
/*6*/"\x79\x81\x82\x83\x84\x85\x86\x87\x88\x89\x91\x92\x93\x94\x95\x96"
/*7*/"\x97\x98\x99\xA2\xA3\xA4\xA5\xA6\xA7\xA8\xA9\xC0\x6A\xD0\xA1\xFF"
/*8*/"\x68\xDC\x51\x42\x43\x54\x49\x48\x9A\x53\xEF\xCF\x56\xB9\x63\x69"
/*9*/"\x71\x78\x58\xCB\xCC\x77\x57\xAA\x8A\xEC\xFC\xFD\xDD\xBA\xBF\x47"
/*A*/"\x45\x55\xCE\xDE\xB1\xA0\xB8\xB6\x72\x52\x07\xB7\x67\x8F\x08\x14"
/*B*/"\x20\x21\x22\x23\x24\x65\x62\xDA\xAF\x15\x06\x17\x28\xB4\xB2\x29"
/*C*/"\x2A\x2B\x2C\x09\x0A\x1B\x66\x46\x30\x31\x1A\x33\x34\x35\x36\x9F"
/*D*/"\x8C\xAC\xFA\x73\xEA\xAB\x75\x76\xDF\x38\x39\x3A\x3B\xB3\x74\x04"
/*E*/"\xEE\x59\xEB\xBB\x9B\x8B\xBC\x9C\xED\xFE\xCD\xFB\x8D\xAD\x44\xBE"
/*F*/"\xCA\x64\x9E\x70\x80\xB5\xE1\x9D\x90\xBD\xB0\xDB\xAE\x8E\x3E\x41"
};

```

```

void Translate(UC *TargetStr, UC *SourceStr,
              UC *TrT, ULI NoCh)
{
    ULI i;
    for(i=0; i<NoCh; i++,SourceStr++)
        *TargetStr++ = TrT[*SourceStr];
}

void TranslD(UC *Str, ULI LenStr, char TypeCode)
{
    void Translate(UC *,UC *,UC *, ULI);

    if (TypeCode == 0x20) /* Source is ASCII code */
        Translate(Str,Str,TrT_Ascii_Ebcdic,LenStr);
    else
        if (TypeCode == 0x40) /* Source is EBCDIC code */
            Translate(Str,Str,TrT_Ebcdic_Ascii,LenStr);
}

```

INSTRUCTIONS FOR PROGRAM COMPILATION

We used Microsoft Visual Studio 6.0 C++ to create tcp62prm.exe. The additional library path for the C/C++ preprocessor is as follows:

C:\Program files\Personal communications\include

The additional library path for link input is:

C:\Program files\Personal communications\lib

The following libraries must be added to Link Object/library modules:

- winnof32.lib
- wappc32.lib

EXECUTION OF CLIENT PROGRAM – SAMPLE

```

tcp62prm LUTEST11 LUTEST11 CICSTEST CICSTEST IBMRDB NETID1.LUTEST11
NETID1.CICSTEST SAMP IBM.PS.CO.YU
SAMPLE CICS PROGRAM
SAMPLE: PROC OPTIONS(MAIN);
  /*** SAMPLE CICS PROGRAM FOR ILLUSTRATING CLIENT'S FUNCTIONALITY ***/
  /*** PROGRAM IS CONNECTED WITH TRANSACTION SAMP ***/
  DCL INPUT CHAR(36);
  DCL OUTPUT CHAR(50);
  DCL LEN36 BIN FIXED(15) INIT(36);
  DCL LEN50 BIN FIXED(15) INIT(50);
  EXEC CICS RECEIVE INTO(INPUT)
          LENGTH(LEN36);
  OUTPUT='TRANSACTION ANSWER IS RECEIVED! ';
  EXEC CICS SEND
          FROM (OUTPUT)
          LENGTH(LEN50);
  EXEC CICS RETURN;
END SAMPLE;

```

Gordana Arsenijevic
Systems Programmer
Postal Savings Bank (Yugoslavia)

© Xephon 2002

Why not share your expertise and earn money at the same time? *TCP/SNA Update* is looking for REXX EXECs, macros, CLISTs, program code, etc, that experienced networkers have written to make their life, or the lives of their users, easier. We will publish it (after vetting by our expert panel) and send you a cheque when the article is published. Articles can be of any length and can be sent or e-mailed to Fiona Hewitt at any of the addresses shown on page 2.

More information about how to contribute is contained in our free booklet entitled *Notes for Contributors*. Ask us and we will send you a copy, or download it from our Web site – <http://www.xephon.com/nfc>.

Setting up a corporate portal – getting started

Although the cynics among you may yet again scoff (despite recent evidence confirming the gravitational pull of Internet technologies), the next stage in the evolution of ‘TCP/SNA’ applications will revolve around corporate portals.

Corporate portals are the next logical step in the evolution of today’s obligatory ‘home page’ Web sites. A corporate portal differs from a static ‘home page’ Web site in two fundamental ways: transactional capability and personalization. Corporate portals enable authorized users to conduct bi-directional, self-service transactions – eg query account balances, make payments using credit cards or electronic cheques, order goods or services, check the status of an order delivery, and so on. Since many of these transactions will be realized using today’s mission-critical applications, there is an inextricable link between TCP/SNA applications and this emerging trend.

The other key feature of a corporate portal is personalization. This feature allows the content, applications, and services available via a corporate portal to be tailored and personalized to meet the exact demands of specific users or user groups – who will be identified through an authentication process or via a ‘self-describing’ scheme involving some kind of ‘cookie’ (or applet) technology.

A corporate portal is a secure, Web-based, easy-to-use focal point of access to a diverse range of corporate information, services, applications, and expertise – for both internal and external users. It is the foundation for building a sound and extensive Web-based corporate operation. It is, in effect, a Web-based, fully-automated emulation of a Utopian, 24x7 call centre operation, where customers, partners, prospects, suppliers, and investors will always be able to get what they want without ever being put on hold or made to jump through the hoops of a voice response system. It’s the optimum way to conduct most of the business transactions previously accomplished using telephones, faxes, or express mail. A corporate portal, at a minimum, will reduce operational costs, increase corporate reach, enhance competitiveness, bolster customer loyalty, improve productivity, expedite business processes,

and accelerate decision making.

The IBM world is curiously split when it comes to corporate portals. The large Fortune 500 mainframe and AS/400 customers (eg Charles Schwab, FedEx, American Airlines, CitiBank, and Fidelity Investments) have been major proponents of corporate portals for some years – with some even implementing their first portals in the mid-1990s. However, this enthusiasm hasn't been shared by mid-tier corporations with mainframes and AS/400s, even though the advantages of portals are now widely known and accepted across much of the business world.

With cost containment still dominating the corporate world, it's certain that portals will be adopted over the next four years, because they offer a relatively easy way to reduce corporate costs while at the same time increasing corporate reach, image, and effectiveness. This doesn't mean that corporate portals are easy and inexpensive to implement and maintain. What it does mean is that the ROI from a corporate portal is realized very quickly (ie in less than two years), and that the ongoing payback after this is significant and virtually guaranteed.

However, this article is not about the business rationale for portals. Instead, it deals with the key technical pre-requisites that you as a senior IT professional will be expected to know about when senior management decides that your company should upgrade the existing Web site to be a corporate portal. (If you're interested in the business justification, you should review portal-related documentation available on the Web sites of companies like IBM, SAP, Microsoft, Oracle, BEA, iPlanet CA, and PeopleSoft.)

ELIMINATING THE STUMBLING BLOCKS

More than with any other contemporary IT initiative, getting started on a corporate portal tends to depend on high-level 'political' mandates rather than on technical concerns and viability. There are many reasons for this, and the most important ones are obvious: for a start, a corporate portal will start to change the way that customers, prospects, suppliers, and partners conduct business with the company. The portal will also soon become a company's primary, and most cost-effective, means for public relations and marketing. Furthermore, implementing and running

a corporate portal requires a sizeable budget, cross-department cooperation (eg for getting up-to-date content), and a committed team of portal-specific personnel. Clearly, then, IT departments cannot typically embark on implementing a corporate portal without a clear and unequivocal mandate from senior management, backed up by an appropriate budget.

Most companies create a detailed business plan to justify and confirm the advantages of having a portal, weighed against the anticipated costs and risks. There are four primary technical issues that invariably get raised during the preparation of such a portal-related business plan. These are:

- Security measures needed to prevent attacks, safeguard corporate assets, and avoid unauthorized access to sensitive information.
- The optimum platform for implementing the portal in terms of reliability, performance, resilience, scalability, and add-on availability.
- The best software options for implementing a comprehensive and sophisticated corporate portal in terms of features, flexibility, availability, extensibility, support, and value for money.
- The options for easily leveraging existing TCP/SNA applications within the portal structure.

This two-part article aims to give you the information you need to address these questions.

Given that security is by far the most often cited reason why companies shy away from public-access portals, it's always best to start off by getting the security issues off the table. Portal-related security concerns invariably revolve around two distinct scenarios: preventing unauthorized access to portal and IT resources, and safeguarding the portal and IT resources from viruses. Note, however, that implementing a portal is unlikely to affect a corporation's vulnerability to viruses one way or another.

E-mail-borne viruses, now the most prevalent and damaging form of virus propagation, will occur irrespective of whether a corporation has a portal or not, as long as it has an e-mail interface with the outside

world. While e-mail is usually one of the main collaborative tools offered by a portal, e-mail-borne virus distribution is still really an e-mail issue rather than a portal issue. In most cases, portals will typically leverage a company's existing e-mail infrastructure rather than implement a portal-specific e-mail scheme. Measures for the detection and deflection of e-mail-borne viruses therefore have to be continually performed regardless of whether a portal is present. It might be argued (rightly) that the presence of a portal will make it easier for hackers to determine valid e-mail addresses relating to the company. But shying away from portals for this reason would be rather like stopping using the company's name in public because that too could provide hackers with hints as to the company's e-mail address structure (given that most companies opt to use the company name as the basis for their e-mail addresses).

FIREWALLS ALONE ARE NOT ENOUGH

The other concern when it comes to viruses is that of a hacker using the corporate portal, with its obligatory Internet interface, as a means through which to infiltrate the company's IT structure in order to plant a virus on a server. This is a valid concern, though it still falls into the category of preventing unauthorized access to the portal and the resources accessible through it. Firewalls alone are not the total answer to this challenge. The reason is that a primary goal of a corporate portal is to provide an open, but secure, Web-based interface for b2c transactions with the general public and b2b transactions with prospective partners (eg new suppliers from new countries). This 'open-to-the-public' aspect means that firewalls cannot be used to filter out unknown addresses, since most of the addresses will indeed be unknown. That's not to say that firewalls aren't necessary; it's customary to have firewalls configured in so-called 'proxy mode' both in front of and behind (ie between it and the other IT systems) a corporate portal, so that the company's real IP addresses aren't exposed to the outside world.

Given that portal server vendors are well aware that security is of paramount concern, they go to great lengths to offer credible and comprehensive security functionality, in the form of authentication, rules-based policy enforcing engines, and personalization, as well as advocating the placement of portal servers within 'demilitarized zones'

(DMZs) created by the use of firewalls in front of and behind the portal server. If a user isn't explicitly authenticated (ie identified and validated), personalization ensures that that user has only basic, non-intrusive rights to data and function access. Casual users from the general public will start off in this unauthenticated category. Personalization, backed up by rules-based policy enforcement, can ensure that only selected company personnel have the necessary rights to view sensitive data or modify portal or IT resources. Decisive and bullet-proof authentication is a prerequisite for personalization.

PUTTING YOUR TRUST IN AUTHENTICATION

All of today's off-the-shelf, portal server solutions (eg IBM's WebSphere Portal and BEA's WebLogic Portal) offer extensive and extensible user authentication functionality. In addition, trusted and respected names in network security, such as RSA Security, Tivoli, and CheckPoint, offer powerful, policy-based authentication systems that can be used to augment the functionality provided by portal servers. Most of the authentication systems, in particular those from the security specialists, offer multiple, time-tested options for identifying and validating users. These include simple user-ID and password schemes, Internet-oriented client-side digital certificates, and RSA's award-winning 'two-factor' SecurID token-based authentication.

Digital certificates, the fast-emerging basis for many Internet-centric security initiatives, are an electronic credential issued by a trustworthy organization such as a large company (eg IBM or Microsoft) or a security-specific entity such as VeriSign or Tivoli SecureWay Trust Authority. The digital certificate vouches for an individual's or business's identity and authority to conduct secure transactions over the Web. In essence, they replace traditional user-IDs and passwords. Products such as IBM's WebSphere Host On-Demand and WebSphere Host Publisher have supported digital certificates for the last few years. The Public Key Infrastructure (PKI) that is rapidly becoming the accepted standard for enforcing Internet encryption is the underlying basis for digital certificates. Digital certificates are defined by the X.509 standard.

'Two-factor' authentication is even more powerful, and can offer significant peace of mind when it comes to corporate portal security. By

the start of 2002, RSA's two-factor SecurID user authentication was already being used by over 10 million Web users around the world, with this user base including the employees of most of the top, high-tech networking companies such as Cisco and NT. It is referred to as two-factor authentication because it requires users to identify themselves using two unique factors – one on top of the other. One of the factors would be something they know (eg password or PIN) while the other factor would be something they have. Automated teller machine (ATM) cards, though not based on RSA's SecurID, are an easy-to-understand example of two-factor authentication, where your personal PIN is the factor that you know and the ATM card with its encoded magnetic stripe is the factor that you have.

In reality, RSA does offer an ATM card type system, based on a chip-embedded smart card and a smart card reader that's connected to the user's PC. This kind of smart card-based system, though offering exceptional security, is obviously rather cumbersome, complicated, costly, and – above all – difficult to deploy in the case of remote or travelling users. But it is, nonetheless, a viable option. It could be used selectively to safeguard your most sensitive information, with the smart cards and the readers restricted to specific, privileged users.

RSA's SecurID, however, can be used very effectively, without the use of this smart card system. In general, it works with a user-specific, secret password and a 'token'. The password is the factor that a user knows, while the token becomes the factor that the user has. The token is a time-synchronized code that is periodically generated, typically every minute, starting with a unique seed code supplied by RSA. RSA can determine the validity of the token based on the time-sensitive code entered by the user. A valid token proves that the user has access to the 'factor' that they're supposed to hold. In order to be successfully authenticated, a user has to enter this continually updated code (ie the token) and user-specific secret password. RSA tokens can be generated using RSA-supplied hardware gizmos or using RSA-supplied software that can run on standard PCs, Palm handheld PDAs, or even smart telephones.

BOTTOM LINE WHEN IT COMES TO SECURITY

So the first step in getting up to speed with regard to implementing a

corporate portal is to master the contemporary security options that are now available. Start by visiting www.rsasecurity.com. Look at the products it offers. Then visit www.verisign.com and go to its section for 'enterprises'. Check out the references to 'PKI solutions', which cover both digital certificates and PKI. Also check out IBM's Web site for information on digital certificates and PKI. If you already use firewalls for your Web site, as you most likely do, there's no need for additional research on firewalls since they'll be used in much the same way with a corporate portal as they are with a Web site.

Coming up to speed on authentication, digital certificates, PKI, and 'two-factor' authentication will allow you to persuasively address management concerns about the security aspect of corporate portals. Always remember that, though it may sometimes feel like it, you are not in the vanguard of the corporate portal implementation push. There are thousands of highly successful and heavily-used corporate portals in production use today – many of which were implemented pre-Year 2000.

A follow-up article, in the next issue, will address the platform, software choices (ie portal servers), and TCP/SNA application integration options involved when you make the move to a corporate portal.

Anura Gurugé
Strategic Consultant (USA)

© Xephon 2002

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*.

Send your suggestions to the editor at any of the addresses shown on page 2, or e-mail fionah@xephon.com.

Accessing the mainframe with ftp software products

The Internet standard File Transfer Protocol (ftp) is arguably the best way to transfer mainframe files and datasets to and from workstations, servers, non-IBM hosts, and other mainframes. This is especially true for workstations emulating 3270s, where the slow and error-prone IND\$FILE has been the standard for nearly 20 years.

This article looks at popular ftp software products and components for Windows workstations, and tests how they work when accessing the datasets under your TSO/E user ID on a z/OS mainframe.

Before we start, however, it's worth noting quickly that we're not talking here about IBM's File Transfer Program, which was withdrawn in 1992. Despite the similarity in name and the same acronym, this was SNA- not TCP/IP-based. A more sophisticated version eventually became known as the Tivoli NetView File Transfer Program.

WHAT FTP IS

Originally developed in 1971, ftp provides a standard method of exchanging files across the Internet and other TCP/IP networks, even when the platforms differ at either end of the transfer. The implementation is a client/server architecture, where the requestor is the client. These days, the ftp server in a z/OS environment runs on Unix System Services (USS). And z/OS batch jobs can access an ftp client that will allow access to an ftp server on other mainframes and hosts.

There are two types of file in ftp: binary and text. Conversions between ASCII and EBCDIC are performed only on text files, so it's usually important to get it right. A more subtle problem is that ftp doesn't expect text files to have anything but text in them; this can cause corruption, usually lost characters, when certain characters are transferred as text instead of binary.

ftp provides a fairly standard view of file systems, opting for the Unix hierarchical model and file attributes. While not a stretch for

Windows/DOS, which was inspired by Unix, it's less than a perfect fit for z/OS where even Partitioned DataSet (PDS) members are viewed as yet another level of file hierarchy, as is each qualifier of the DataSet Name (DSN).

From the user perspective, the base processing for a workstation ftp client and a host ftp server is as follows:

- Start the workstation ftp client software.
- Connect to the host ftp server.
- Enter user ID and password.
- Position yourself into the correct directory within the host file hierarchy.
- Position yourself into the correct workstation directory.
- Select the file(s) to be transferred.
- Indicate the direction of transfer: workstation to host, or vice versa.

The sequence of these steps may vary, and may even be automated for you, depending on the ftp software being used, and it's possible to transfer multiple files at a time.

FTP ON THE MAINFRAME

ftp on the mainframe is provided by TCP/IP for MVS, which is included in TCP/IP Services – part of Communications Server.

z/OS ftp includes both a server and a client. Although I'm not sure why you would want to do so, you can even use the ftp client to access the ftp server on the same system, as follows:

```
//FTPTEST EXEC PGM=FTP,PARM='209.217.251.162'  
//SYSPRINT DD SYSOUT=*  
//OUTPUT DD SYSOUT=*  
//INPUT DD *  
JONPE  
{password goes here}  
SYST  
DIR  
QUIT
```

The SYSOUT data from the OUTPUT DD statement is shown below:

```
//
EZA1736I FTP 209.217.251.162
EZA1450I IBM FTP CS V2R10 2000 093 23:39 UTC
EZA1554I Connecting to: 209.217.251.162 port: 21.
220-FTPD1 IBM FTP CS V2R10 at S390, 03:59:23 on 2002-06-23.
220 Connection will close if idle for more than 50 minutes.
EZA1459I NAME (209.217.251.162:JONPE):
EZA1701I >>> USER JONPE
331 Send password please.
EZA1789I PASSWORD:
EZA1701I >>> PASS
230 JONPE is logged on. Working directory is "JONPE.".
EZA1460I Command:
EZA1736I SYST
EZA1701I >>> SYST
215 MVS is the operating system of this server. FTP Server is running on
OS/390 UNIX System Services.
EZA1460I Command:
EZA1736I DIR
EZA1701I >>> PORT 209,217,251,162,13,180
200 Port request OK.
EZA1701I >>> LIST
125 List started OK
EZA2284I Volume Unit Referred Ext Used Recfm Lrecl BlkSz Dsorg Dsname
EZA2284I ISPW89 3390 2002/06/04 1 15 FB 80 27920 PO ACTIVE.ASM
EZA2284I ISPW89 3390 2002/06/04 1 1 VB 255 27998 PO ACTIVE.CLIST
EZA2284I ISPW89 3390 2002/06/22 4 13 FB 80 27920 PO ACTIVE.CNTL
EZA2284I ISPW81 3390 2002/06/03 2 6 FB 80 27920 PO ACTIVE.COBOLE
EZA2284I ISPW89 3390 2002/05/31 8 36 FB 80 27920 PO ACTIVE.DATA
EZA2284I ISPW89 3390 2002/06/03 2 2 VB 255 27998 PO ACTIVE.EXEC
EZA2284I ISPW81 3390 2002/06/03 2 11 U 27998 27998 PO ACTIVE.LOAD
EZA2284I ISPW89 3390 2002/05/30 1 15 FB 80 27920 PO ACTIVE.MACLIB
EZA2284I ISPW89 3390 2002/03/22 1 1 FB 80 27920 PO ACTIVE.PLI
EZA2284I ISPW89 3390 2001/10/21 1 1 FB 80 27920 PO ARCHIVE.CNTL
EZA2284I ISPW89 3390 2002/02/02 1 7 VB 255 27998 PO ARCHIVE.EXEC
EZA2284I ISPW89 3390 2002/06/22 2 8 FB 80 3120 PO ISPF.ISPPROF
250 List completed successfully.
EZA1460I Command:
EZA1736I QUIT
EZA1701I >>> QUIT
221 Quit command received. Goodbye.
```

Although beyond the scope of this article, it should be mentioned that, as shown in the JCL, in-stream data for the INPUT DD works only without sequence numbers. This is the most common mainframe ftp error.

THE PRODUCTS AND COMPONENTS TESTED

The ftp products and components tested in this article were selected because they're the ones to which you're most likely to have access. Those included with the most popular 3270 emulators were the first choice, because they were designed with mainframe ftp in mind, are likely to be feature-rich and powerful, may already be installed on your workstation, and are effectively free. These include the ftp clients with WRQ Reflection for IBM, WRQ Reflection for the Web, IBM Host On-Demand, and RUMBA. (Unfortunately, however, the latest version of RUMBA was not received in time for evaluation.) Attachmate lists ftp as an (extra charge) option, so it was not included.

Shareware is another commonly used source of software. WS_FTP and CuteFTP are two of the highest rated and most popular ftp products, even though distribution of the latest versions differs little from commercial products: downloadable 30-day evaluation copies.

THE TESTS

The tests performed are shown in Figure 1 in the order in which they were conducted. Note that only the exceptional results are reported in the individual reviews below; if most products had the same noteworthy results, it is shown in Figure 1 instead.

THE TEST ENVIRONMENT

The tests were performed on Version 2.10 of OS/390, with the ftp server running on OS/390 Unix System Services. The workstation environment was a 300MHz processor running Windows XP Professional with all updates applied, as offered through Windows Update, with display resolution set at 1024x768. For the two Web-based ftp components, the workstation was connected through a 500MHz Windows 2000 Server with all updates applied.

Both workstation and server were running McAfee VirusScan Online, and both were connected to a WatchGuard SOHO firewall which connects to the Internet via ADSL. In turn, 200 miles away, the mainframe was connected to the Internet.

Does default install/connection display mainframe datasets?

Initial view: how are DSNs displayed? All products display DSNs without their High Level Qualifier (HLQ)

Are dates displayed when mainframe uses non-standard system date format?

How to view ftp commands and response.

Up one level from default view. All products display a blank file list without further explanation and no obvious way to escape

Size displayed for sequential dataset. Only the Reflection products display a non-zero value – space allocated in bytes.

(How) are VSAM components displayed?

Is size displayed for VSAM data components? No product does this.

(How) are GDG base components displayed? No product even displays the DSN.

Is PDS displayed as folder? All but CuteFTP do this.

Is size displayed for PDS? Only Reflection for the Web does, showing Bytes allocated.

Can you double-click to see a PDS directory? All but CuteFTP allow you to double-click a PDS to see its member directory.

Is size displayed for PDS member with STATS? Reflection for the Web displays the number of lines in each PDS member, WS_FTP displays 0KB or 1KB, and the rest show Size as blank.

Is size displayed for PDS member without STATS? All products display blank or zero values when no ISPF statistics exist for a PDS member.

View after Up One Level from PDS. All products display LLQ of PDS and rest of qualifiers for any DSNs with the same higher level qualifiers.

Can you double-click the PDS again? Works for all products.

Download 7 members of a PDS. 80-byte fixed length records of text.

Open one member in NotePad on workstation. To determine whether it was correctly translated from EBCDIC to ASCII.

Upload 7 members of a PDS. From workstation to mainframe.

Are members identical? ISPF option 3.12 SuperC was used to compare.

Delete 7 members of a PDS. Keyboard Del key is not supported by all products; when all members are selected and a single Delete request made, some products require confirmation of each member's deletion.

Download a PDS directly. Select the PDS for transfer to the workstation, rather than opening it up and selecting PDS members.

Figure 1 (part one): Tests performed

Open one member in NotePad on workstation.

View after downloading a PDS. As they should, all products (that support selecting a PDS directly for download) maintain the same View as seen before the download.

Does Refresh change view? Only a beta bug failed this test.

How to Refresh? Button and/or menu bar choice.

Does Refresh refresh both mainframe and workstation view? All do except HOD.

Download a VSAM cluster. No product can do this.

Drag and drop to Download and Upload? Surprisingly wide variation among products.

How is binary/text decision made? What determines whether a mainframe PDS member, sequential dataset, or entire PDS (all members) is transferred as-is, or treated as text and converted from EBCDIC to ASCII on its way to the workstation? This is a much more complex issue than you might expect.

Use Delete key to delete non-empty PDS. Selecting a PDS and hitting the keyboard's Del key.

Use Delete key to delete non-empty workstation folder.

Figure 1 (part two): Tests performed (continued)

WRQ REFLECTION FOR IBM

WRQ Reflection for IBM is a Windows-based 3270 terminal emulator that includes WRQ Reflection FTP Client. Version 8 was reviewed in '3270 terminal emulation using Reflection' (*TCP/SNA Update* 41, March 2001, pp 14-20). Version 10 is due out in September, so a May 2002 beta version was tested. A default ('typical') install includes ftp, but a custom install was performed with only the following selected: FTP client, user help, and English language support.

Running ftp

From the Start menu, All Programs/WRQ Reflection/FTP Client runs the stand-alone ftp component. A Reflection FTP Client window is immediately overlaid by a Connect to FTP Site dialogue box with two WRQ ftp site entries (which can be eliminated during custom install). Hit the New button and an Add FTP Site wizard appears, prompting you

for the minimum information required for any ftp site you wish to connect to:

- Host name or numeric IP address.
- User name, unless you choose anonymous connection.
- Password – you're given the choice of having the password encrypted into the registry or entering it each time you connect.
- Name of site.
- Whether or not you want to immediately connect to the site.

You will see the Connect to FTP Site dialogue box each time you start Reflection FTP. It displays four fields for each site:

- Name
- Address – host name or IP address
- User Name
- Home Directory.

Home Directory, and many other values not prompted for by the wizard, can be set by selecting the relevant Name entry (ftp site) and pushing the Properties button. There, you'll find five tabs:

- General
- Connection
- Directories
- Transfer
- Translation.

Once you're connected

Selecting an entry and pushing the Connect button, the Connect to FTP Site dialogue box soon disappears, leaving the Reflection FTP Client window divided into two. On the left you'll find the contents of your workstation's My Documents folder. The toolbar has an Up One Level icon and you can go down one level by double-clicking on any folder.

The menu bar's Tools/Go To allows you to type a complete path into a blank field.

A highlighted title bar indicates that any actions will apply to the right, where you'll find your mainframe datasets displayed without their high level (your RACF and/or TSO/E ID). Partitioned DataSets (PDSs) are shown as folder icons. Sequential datasets have the icons used by Windows to indicate file types not associated with known programs. VSAM files are not displayed. Nor is a GDG base. Although not tested, View/Filters allows you to control what's seen through Server View Filters.

The file list

Both sides are displayed alphabetically, folders first, in what's known in Windows as Large Icons view. Either side's view can be changed from the menu bar: View/Local Pane and View/Server Pane(s), respectively. Details view is the most useful, with the following fields:

- *Name*. Rest of DSN name beginning at current level, or PDS member name.
- *Size*. Bytes allocated for a sequential dataset, lines for a PDS member, zero if no ISPF statistics, and not shown for a PDS itself.
- *Type*. The LLQ for a sequential dataset, 'File Folder' for a PDS, and 'File' for a PDS member.
- *Modified*. Date and time of last change, ****NONE**** for datasets unchanged since creation, and blank for PDS members without ISPF statistics.
- *Attributes*. VOLSER where the dataset resides and blank for PDS members.

The Modified date was incorrectly displayed for a mainframe with non-standard system date format. This problem doesn't exist in Version 9.0.1 and will be fixed either in the final Version 10.0 or in 10.1.

At the top of the window, just below the title bar, are a menu bar and tool bar. View/Command Window displays the ftp commands generated and responses received from the mainframe. There's even space for you

to enter your own ftp commands.

Test results

A Smart Transfer File Type dialogue box pops up the first time you attempt to download or upload each File Type, asking you to select from three types of file transfer: ASCII, Binary, and Tenex. According to Reflection Help, Tenex is also known as Local 8, and is used for transferring non-8-bit bytes found on computers such as the DEC System 20.

File Type is the Low Level Qualifier (LLQ) of mainframe datasets and the Extension of workstation files, with two exceptions:

- When in a View that displays only the LLQ of a DSN
- PDS members.

In both cases, File Type becomes No Extension. Smart Transfer uses the No Extension type of each PDS member when a PDS, rather than members in a PDS, is selected for transfer.

Deletion of multiple PDS members was the easiest of any of the products tested. Only one confirmation is required after hitting the keyboard's Del(ete) key. The Del key can also be used to delete a PDS, although it can take a while for a PDS with a large number of members, because Reflection FTP needlessly deletes each member before deleting the PDS itself.

From the menu bar, View/Refresh updates both the workstation and mainframe file lists. There is no browser-style Refresh button.

No attempt was made to evaluate or even look at programmability, a major focus of Reflection FTP. It provides script, typically used for automating tasks, as well as OLE, allowing Visual Basic (VB) applications to access FTP functionality without having the GUI present.

WRQ REFLECTION FOR THE WEB

WRQ Reflection for the Web (rweb) is a browser-based (Java) 3270 terminal emulator with ftp capabilities. Version 4.5 was reviewed in

‘Web-based terminal emulators – Reflection for the Web’ (*TCP/SNA Update* 45, March 2002, pp 56-69) and ‘HOD versus Reflection’ (*TCP/SNA Update* 46, June 2002, pp 3-6). But installation has been improved for Version 5.0, which was tested for this review.

Running ftp

The ftp client is the only one reviewed that must be initiated from the 3270 emulator. From the menu bar, File/Transfer/FTP gets you there. During installation, if you configured the terminal session with Basic User Menu Level, users will not have access to the ftp client and the menu item will not be shown.

An ftp window appears, immediately overlaid by a Connect to Server dialogue box. FTP Address contains, greyed out, the IP address used for the terminal emulation session. User name and Password are entered and the Connect button pushed.

Once you’re connected

The Connect to Server dialogue box disappears, leaving an ftp window dominated by a Local files section on the left and Server files section on the right. The Local files section is titled with the workstation’s System Name defined when the operating system was installed. Both the path and list of files are blank. To the right of each Path field is a button with an Up One Level icon on it. Moving a mouse over the button displays ‘Change to parent directory’ in the Status bar. Pushing the button generates a valid folder, from where you can navigate to wherever you want:

```
C:\WINDOWS\Java\reflectionweb\
```

If you want ftp to remember your folder – and anything else, such as user ID and password between sessions – you must change the Deployment Director’s default when creating the terminal session. You would do this in the Deployment Director’s Configuration step (Step 4), where you see a 3270 terminal session, select Administrator/Set User Preference from the menu bar and check the box to save each user’s ftp settings.

Then you can use Setup/General tab/Default local folder. The Browse button initiates the wrong type of Open dialogue box (titled Select

Default Directory), forcing you to select a file, not a folder, before it will insert the path, not including the file name, into the field. This is a problem if your desired folder is empty, but it should be resolved in the next release.

The Server files section is titled with the FTP Address and shows your High Level Qualifier (HLQ)/ID in single quotes: 'JONPE.'

The file list

The list of files begins with each PDS in alphabetical order and displayed as a folder, and then shows the rest as text files. Sequential datasets and all VSAM components are shown, but not GDG base entries. Both sides are shown in Details view, with more fields shown on the mainframe:

- *Name.* Lower level qualifiers of DataSet Name (DSN), initially missing HLQ; member name inside a PDS.
- *Size.* Bytes allocated, blank for VSAM and PDS members.
- *Modified.* ****NONE**** or date of last modification, 'VSAM' for VSAM entries, and blank for PDS members without ISPF statistics.
- *Attributes.* Blank.
- *Owner.* Blank.
- *Group.* Blank.

The top left corner of the ftp window has Disconnect and Setup buttons; there is no menu bar or tool bar. The bottom has a Show FTP Console check box, Help, Close, Copy console data to clipboard, and Erase console data buttons, and a status bar, typically with Connected or Disconnected greyed out.

Between the two sides are six buttons, each with an icon on it: Refresh, Upload – Send, Download – Receive, New folder, Rename file/directory, and Delete file/directory. These buttons are vital, given that drag and drop uploads and downloads are not supported. Nor is selecting a file and hitting the Delete key on your keyboard.

Clicking the Show FTP Console box places a check mark in it, and

expands the size of the window to include a scrollable eight-line area showing ftp commands and responses.

Test results

Differentiating between text (EBCDIC to ASCII conversion) or binary transfers is based on Extension: Low Level Qualifier (LLQ) for datasets, even when only the LLQ is displayed in a file list. The one exception is a PDS. Whether transferring a PDS member or an entire PDS, a Null Extension is assumed based on the fact that member names cannot contain a dot (‘.’).

By default, the transfer type is Binary, with the following extensions specifically defined as ASCII (text): bat, htm, html, ini, java, ofslogfl, sys, txt, xyz. And the following as Binary: 123, 3fl, com, doc, exe, gif, jpg, lib, MAR, p, wav. You can see and change these definitions and the default transfer type, by hitting the Setup button and clicking the Transfer tab.

The Server files view can change without your knowledge when downloading a complete PDS. For example, beginning in default view, if you select TESTFTP.CNTL and request a download, the view (mistakenly) changes to CNTL after the download, but you’ll still see TESTFTP.CNTL. If you double click on TESTFTP.CNTL in the Server files list, you’ll suddenly see an empty file list as the path changes to JONPE.TESTFTP.TESTFTP.CNTL. This small bug is scheduled for resolution in the next release.

IBM HOD

Host On-Demand (HOD) is a Web-based 3270 terminal emulator, part of IBM Host Access Client Package (HACP), reviewed last issue in two articles: ‘Host Access Client Package – HOD Component’ (pp 6-11) and ‘HOD versus Reflection’ (pp 3-6). As in those articles, Version 2.0 of HACP, which includes HOD V6.0, was tested.

Installation

With HOD, there are two ways to use ftp, but both require creating special, non-default sessions with the Deployment Wizard. With

Windows 2000 Server, Start/Programs/IBM Host On-Demand/Administration/Deployment Wizard.

When creating a Host Session, select the usual Host Type of 3270 Display, hit Add, hit Properties, and change the File Transfer Type on the General tab to FTP from the default Host File Transfer. You can even push the File Transfer Default button to view the Connection tab where you can pre-set, and optionally lock, the user ID, password, and local and remote home directories. Given the fact that you're on a server, not the user's client workstation, there's no Browse button for the Directories. Session Name, Destination Address, and Port have reasonable defaults which can also be changed. You can also change the default that disables Anonymous Login. There are also three more tabs of defaults: Advanced, More Advanced, and Screen.

Alternatively, when creating a Host Session, you can select a Host Type of FTP. Hit Add, hit Properties, and see the same Connection tab (with access to the other three) that you saw when you hit the File Transfer Default button above.

Running ftp

In fact, both methods give you the same ftp client. The only difference is in the way that you initiate it from the Reflection session Web page. In the first case, you double-click the 3270 terminal emulator icon and then select Actions/Transfer Files from the menu bar of the Reflection for Web client window. In the second case, you directly double-click an ftp icon on the Reflection session Web page.

Initially, the ftp window is overlaid with an FTP Login dialogue box containing four fields: user ID, password, destination address, and destination port.

Unless you specified them when creating this Host Session, you'll have to enter the user ID and password now and hit the OK button.

Once you're connected

The Login dialogue box disappears, leaving the ftp window. In the default Side-by-side View, the window is divided in half. The left is

labelled Local and the right with the Destination Address. Both sides are populated with directory listings, with the mainframe (right) side greyed out, to indicate the current focus: any actions are performed on the workstation. Above each listing is a Directory field and buttons with Up One Level and New Folder icons on them.

The installation default for the workstation directory path is C:\. On my test client workstation, C: is assigned to a partition used only when dual booted as a production workstation, so I hit the Up One Level button, expecting to be able to choose another drive, but nothing happened. Instead, in the Directory field, I typed E: and hit the Enter key. E:\ appeared and I double-clicked on the Documents and Settings folder, a folder with my name on it (my Windows XP user name), My Documents, and then an appropriate subfolder.

The file list

On the mainframe side, folder entries were listed first, alphabetically, then files, also alphabetically. Just above, the Directory field showed my High Level Qualifier (HLQ)/ID followed by a dot ('.'). And the DSNs were displayed without the HLQ.

Each PDS was a folder, as was each VSAM component. Each GDG base was displayed with a Name field of 'attributes' instead of its DSN, TAPE in the Size field, and all other fields blank. Sequential datasets were listed as files.

The fields displayed for each workstation and mainframe entry, including PDS member, were:

- *Name*.
- *Size*. Blank, except TAPE for GDG base entries.
- *Date*. Modification date, ****NONE**** or blank for all PDS members.
- *Attributes*. Blank.

The initial default view didn't display all of the date. But you could widen it by dragging and dropping using the mouse.

There were a menu bar and tool bar at the top of the window and a status

bar at the bottom: double-clicking on it produces a small scrollable Status Bar History window, including the ftp commands generated and responses received.

Test results

Drag and drop uploading and downloading is not supported. To download, you can either use the Download (left arrow) button between the Local and Remote halves of the window, or a right mouse button drag and drop which displays a four item pop-up menu: Receive Files from Host, Delete, Rename, and Create Directory.

Default file transfer (Edit/Preferences/Transfer Mode) is Auto, which means binary for all but those with their Low Level Qualifier (LLQ)/Extension defined. From the menu bar, select Edit/Preferences/ASCII File Types to see the current list, which defaults to: TXT, HTML, HTM, README, SCRIPT, BAT, INI, LOG, TEXT, LIST, and MBR.

On the mainframe, Extension (LLQ) is not just the name after the last dot; if there is no dot in the displayed DSN or PDS member name, the entire name becomes the Extension. This eliminates the Extension inconsistency between Views where a DSN with only LLQ displayed would otherwise be considered to have no Extension. Unfortunately, it also introduces a new, admittedly obscure, problem: any PDS member named as any of the defined Extensions will be transferred based on that Extension.

Just as it does for a VSAM component, HOD attempts a download, and receives and generates an error when a PDS is selected for download. You have to double-click on the PDS to open its member directory, select all members, and then request a download.

Unique among the products tested, HOD's Refresh feature updates only the file list in focus; the greyed out file list may be out of date. As with all the products, Refresh is necessary only to reflect changes made outside of the ftp component. It should automatically refresh after any of its actions that could change the file list.

One final note. When testing changes made with the Deployment Wizard, your Web browser's Refresh button will not reload the Cached Client; you must close down your browser and restart it. Changes made

with the Deployment Wizard aren't reflected until the Cached Client is reloaded.

CUTEFTP

CuteFTP is currently the best-known shareware ftp product and is highly rated by shareware sites. CuteFTP 5.0 XP was tested, as it's the latest version and costs only \$39.95. The more advanced CuteFTP Pro 2.0 costs \$59.95 and didn't appear to offer any mainframe-specific features. Both are available for a software-enforced 30-day evaluation by downloading 3MB from <http://www.cuteftp.com>.

Running ftp

CuteFTP 5.0 XP installs very rapidly, with minimal interaction. It can be run from the Start button menu:

All Programs/GlobalSCAPE/CuteFTP/CuteFTP.

The first time, you must push the Start Trial button, which starts the CuteFTP Connection Wizard. It prompts you for:

- Label to identify your ftp site.
- Host address.
- User name and password, with anonymous log-on and mask password options.
- Default local folder.
- Options to Connect to this site automatically and Add to right-click shell integration.

It connects to the site even when the Connect option is not selected. The next time you start CuteFTP you are prompted to Continue Trial, then Site Settings for CuteFTP Download Site are displayed. You'll also see the site you defined on the left side of the window. Simply move the selection to your own site, and push the connect button. On subsequent starts, your site will be pre-selected for you.

Once you're connected

There was instantly a big problem: no files were shown on the mainframe. Hitting the Edit button on the Site Settings window displayed a Site Properties dialogue box with General and Advanced tabs. On the General tab, the Host Type was Auto-Detect. Three more promising choices were in the drop down list:

- MVS (no auto-detect)
- MVS V3R1 (no auto-detect)
- MVS X (no auto-detect).

Unfortunately, it seemed impossible to find out the difference between the three.

The file list

Selecting MVS V3R1 and Connecting did display a list of files on the mainframe, but there were still plenty of obvious problems:

- Each PDS showed as a file, not a folder.
- From the menu bar, Commands/Directory/Change Directory to JONPE.TESTFTP.CNTL didn't generate an error, but listed none of the 100 members of the PDS.
- A drag and drop download of a sequential dataset ending .LIST didn't prompt for type of transfer and didn't do an EBCDIC to ASCII translation.

A final bad sign was that an uninstall didn't erase the ftp sites defined, but a reinstall still began by starting the CuteFTP Connection Wizard, prompting you to redefine the ftp site.

Since this was clearly not the product for the mainframe, I stopped my evaluation and moved on.

WS_FTP

WS_FTP is a well-known and highly rated shareware ftp product. Version 7.5 of WS_FTP Pro is the most recent release and was

tested for this evaluation. It's priced at \$39.95 and is available for a software-enforced 30-day evaluation by downloading 4MB from <http://www.ipswitch.com>.

Installation

During installation, a Select Site and Firescript Configuration dialogue box asks you to choose between:

- Share sites and firescripts with all users.
- Allow each user to configure their own sites and firescripts.

In Windows XP, if you choose Per User, you're then warned that each user must have more permissions than a Limited Account type.

You're then prompted for your e-mail address as the default password for anonymous log-ons. The actual installation takes less than 20 seconds even on a slow (300MHz) workstation. The final prompt allows you to uncheck the box Launch WS_FTP Pro Now and hit Finish.

Running ftp

If you didn't choose to install any of the shortcuts, you can run the program from the Start button menu: All Programs/WS_FTP Pro/WS_FTP Pro. You'll need to push Start Evaluation on a Welcome dialogue box; on subsequent runs, Continue Evaluation in an 'Evaluation is running out' dialogue box. Two dialogue boxes are initially displayed – WS_FTP Pro and Transfer Manager – before Connect to Remote Host overlays WS_FTP Pro. Hit Create Site to begin.

The WS_FTP Pro Site Profile Wizard appears, requiring some arbitrary decisions: name the ftp site and where (push Browse), in a hierarchy of folders, you would like to create the site profile:

Sites
MySites
Predefined Sites

Click on either of the lower level sites – Sites itself does not work – and hit OK. Or click on the plus sign ('+') to the left of Predefined Sites to

view the existing categories. Computing and Special Interest have subcategories. You can add your own folders anywhere in the hierarchy by hitting the New button.

Next, you're prompted for Host Name or IP Address. As you begin typing, the Next and Finish buttons are no longer greyed out. The dialogue box that follows has user ID, password, and account fields, and check boxes labelled Anonymous, Save Password, and Save Account. For mainframe ftp sites, Account should be left blank.

Conceivably, my choice in the next dialogue box caused the major failure that would later cause me to abandon further evaluation. The drop-down list for Host Type includes IBM VM, but no other mainframe choices. Instead, I chose the default: Automatic detect. I also chose the other default values:

- Port – 21
- Passive mode – checked
- Use SSL – not checked.

You're asked to select a Firewall Entry and can select <no firewall> from the drop-down list, but the Next button remains greyed out. Hit the Finish button, and you'll see your entry appear in the hierarchy. Select it and hit Connect for immediate access.

Once you're connected

The main window was divided in half, labelled Local system on the left and Remote system on the right. Each half had ChgDir, Mkdir, View, Execute, Rename, Delete, Refresh, and Info buttons to the right of the file list. At the bottom was a count of objects in the file list, and their total size. Above was the current directory which, on the Local system, was set to the installation directory: C:\Program Files\WS_FTP Pro, by default. This annoyance was somewhat offset by the inclusion of all lettered drives at the bottom of the file list on the Local system.

On the mainframe side, each PDS was shown as a folder, a sequential dataset was a file, and VSAM entries were not displayed. GDG Base entries were listed at the bottom of the list as spurious binary files named Error; any attempts to view or transfer them generated Dataset not

found errors.

By default, files on both sides are shown in Details view: Name, Size, Type Modified, and Attributes.

The top of the window had a menu bar, tool bar, and browser-like address field that read:

```
ftp://jonpe@209.217.251.162/JONPE.
```

Near the bottom of the window was a scrollable window of ftp commands and responses; from the menu bar, View/Log Window will eliminate it. Disconnect, Cancel, Help, Options, About and Exit buttons were at the very bottom of the window.

Test results

A simple drag and drop download of sequential file ACTIVE.LIST appeared to complete without incident, but opening it in NotePad on the workstation displayed all special characters, indicating it was transferred as a binary file without EBCDIC to ASCII translation.

From the menu bar, Tools/Options and clicking the Extensions tab indicated the problem: “In Auto-Mode, all files are transferred as binary unless the filename or extension is listed here. Any file with a filename or extension listed here will be transferred as ASCII. This applies to transfers, View, Execute and Edit.”

ASCII Filenames and Extensions displayed an empty field and a list box below it with txt in it. Clicking on the empty field, typing LIST, hitting the Add and OK buttons, repeating the transfer, and opening it on the workstation with NotePad accurately displayed the contents of the ACTIVE.LIST sequential dataset.

Download of a complete PDS was supported by simply dragging and dropping its folder to the Local system subwindow. For JONPE.TESTFTP.CNTL, displayed as TESTFTP.CNTL on the Remote system subwindow, the folder CNTL was created on the workstation. Even though CNTL was added to the ASCII Filenames and Extensions list, binary transfers of each member still occurred because the members were viewed individually, and were considered to have no extension. To force the EBCDIC to ASCII translation to occur, a dot (‘.’) entry can be

added to the ASCII Filenames and Extensions list. It's not possible to add a null entry, but you can add a blank entry, which should also work, though it wasn't tested.

The showstopper was repeated attempts to select PDS members from within the PDS directory, and drag and drop them to the workstation. Although the action was accepted, the operation failed. Lots of errors were generated, many claiming that WS_FTP was not logged on to the mainframe ftp server, and only a few PDS members were transferred.

Given this behaviour, the evaluation was abandoned at this point. As mentioned above, setting Host Type to IBM VM might help. But this wasn't tested given the fact that the PDS is not part of the native z/VM file structure.

Armand Minet
(Canada)

© Xephon 2002

Need help with a TCP/SNA problem or project?

Maybe we can help:

- If it's on a topic of interest to other subscribers, we'll commission an article on the subject, which we'll publish in *TCP/SNA Update*, and which we'll pay for – it won't cost you anything.
- If it's a more specialized, or more complex, problem, you can advertise your requirements (including one-off projects, freelance contracts, permanent jobs, etc) to the hundreds of TCP/SNA professionals who visit *TCP/SNA Update*'s home page every month. This service is also free of charge.

Visit the *TCP/SNA Update* Web site

<http://www.xephon.com/tcpsna>

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

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*, and *MQ 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 <http://www.xephon.com/nfc>. 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

TCP/SNA news

IBM has announced z/VM V4.3, including level 430 of TCP/IP for z/VM. This exploits the multicast capability of a HiperSockets adapter and improves the security of the TCP/IP stack by detecting, reporting, and preventing more types of DoS attack than V4.2.

URL: <http://ibm.com/eserver/zseries/zvm>

* * *

Version 4.3 of NewEra's IMAGE Focus systematically identifies, locates, and inspects thousands of system parameters and resources that make up an OS/390 or z/OS sysplex, including the operating system, JES2/3, VTAM, and TCP/IP.

It includes support for TCP/IP.DATA and RESOLVER configuration syntax introduced in z/OS V1R2 and V1R3, augmenting the existing support for TCP/IP.PROFILE configuration syntax.

URL: <http://www.newera.com>

* * *

MacKinney Systems has announced VTAM Virtual Printer (VVP) Release 1.3, which is used to route print from CICS TS to the JES queue for OS/390 and z/OS or to the POWER Queue for VSE.

VVP is said to require no programming changes and it runs as a VTAM application. All printer maintenance may be done through on-line panels.

URL: <http://www.mackinney.com>

* * *

IBM has announced Version 6.0 of its VisualAge Smalltalk Server for OS/390 and z/OS, the run-time deployment environment for Smalltalk Server applications on OS/390 and z/OS.

With VisualAge Smalltalk Enterprise and VisualAge Server Workbench, sites can write transactions for IMS and CICS, access DB2, VSAM files, and QSAM files, interface with applications using MQSeries TCP/IP, APPC, and CPI-C, and interface with COBOL, C, and PL/I applications.

It also enables sites to run XML applications, interface with Web servers using servlets, run applications that host or invoke Web services, run applications with VisualAge Smalltalk Web Connection, and develop on Windows and OS/2.

Web address: <http://www.ibm.com/software/ad/smalltalk>

* * *

IBM has announced decreases in monthly licence charges for selected zSeries Entry Licence Charge Software for the zSeries 800 family, including TCP/IP for VSE/ESA.

URL: <http://www.ibm.com>

* * *

HP has released OpenVMS Alpha V7.3-1 for AlphaServers, with, amongst other things, an enhanced TCP/IP stack for extended interoperability and performance.

URL: <http://www.hp.com>

* * *



xephon