46 TCP/SNA update

June 2002

In this issue

TCP/SNA Update

Published by

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

North American office

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

Subscriptions and back-issues

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

Editorial panel

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.

Printed in England.

HOD versus Reflection

*In this issue (pp 44-48) and the last one (*TCP/SNA Update *45, March 2002, pp 56-69), our look at 3270 mainframe terminal emulators has taken us to two of the most popular Web-based products: Version 4.5 of WRQ's Reflection for the Web and Version 6.0 of WebSphere Host On-Demand (HOD), currently a component of Version 2.0 of IBM Host Access Client Package for Multiplatforms. This article discusses how they compare when tested side-by-side.*

CO-EXISTENCE AND FTP

The first issue to address is whether Reflection for the Web 4.5 and WebSphere HOD 6.0 can co-exist on the same Windows 2000 Server. They can, and the clients can even run simultaneously on the same Windows XP client workstation. There is, however, one notable difference in installation: to upload its session files, Reflection requires an ftp server that can share a subfolder with the Web server. The default ftp and Web servers in Windows 2000 Server did not point to the same root folder on disk. Worse yet, the default (typical) Windows 2000 Server install does not install an ftp server.

To install the ftp server:

- Start/Programs/Administrative Tools/Configure Your Server.
- Click Advanced on the left sidebar, then Optional Components below it.
- Click Start the Windows Components wizard link.
- Single click to select Internet Information Services (IIS) and push the Details button.
- Click to check the File Transfer Protocol (FTP) Server (if it's already checked, you can cancel as it is already installed).
- Push the OK button, the Next button, insert the Windows 2000 Server CD, and hit OK when prompted.
- Hit Finish to close the wizard.

	HOD	Reflection
Version	6.0	4.50
Separate session window	Default	Option
Embedded session window	Option	Default
Separate: menu bar	Default	Default
Separate: no menu bar	No	No
Embedded: menu bar	No	No
Embedded: no menu bar	Default	Default
Separate: tool bar	Default	Option
Separate: no tool bar	Option	Default
Embedded: tool bar	Default	Option
Embedded: no tool bar	Option	Default
tn3270	Option	Default
tn3270e	Default	Option
Ctrl is Enter	Default	Option
3270 Enter will Reset	Option	No
🖱 is Enter	Default	Default
Left and Right Ctrl can differ	No	No
Users can remap keyboard	Default	Option
Two definitions for one 3270 key	Possible	Possible
Typeahead	Default	Default
No typeahead	No	Option
Insert toggles Insert mode?	Default	Default
Embedded: shortcut menu	No	Option
FTP file transfers	Yes	Yes
IND$FILE	Yes	Yes

Figure 1: Features

JAVA AND CTRL

Because both products are implemented in Java, they both face a major limitation of Java: the left and right Control (Ctrl) keys are not differentiated. This is a key issue because 3270 mainframe terminals have their Reset and Enter keys positioned where the left and right Ctrl keys are located on a PC keyboard.

HOD has a unique, though not perfect, solution: a combination Enter and Reset function that is, by default, initiated by (either) Ctrl key. Whenever the keyboard is locked, it acts as a Reset key; otherwise, it's an Enter key. The only major drawback is during Typeahead. For example, if you're in the habit of typing something and hitting the right Ctrl (3270 Enter) twice to get past the first panel, it won't work. The second Ctrl is interpreted as a Reset, immediately unlocking the keyboard, which was locked by the immediately preceding Enter.

	HOD	Reflection
To view current keyboard mapping	Edit/Preferences/Keyboard, Category=Host Functions	Setup/Keyboard/View All
How to map a key	Edit/Preferences/Keyboard	Setup/Keyboard/Add
Select screen text	Click, hold and move mouse	selects rectangle of text
File Transfer	Actions/Transfer Files	File/Transfer
Shortcut menu	No	Right mouse click
Allow users to remap keyboard	Default	Intermediate or above menu

Figure 2: How to?

	HOD	Reflection
3270 Enter	🖱	🖱
3270 Reset	Not assigned	Esc
Combo Reset/Enter	Ctrl	Not available
Clear	Esc or Pause	ScrollLock or Ctrl+F2
System Request	Not assigned	Ctrl+S
Attention	Not assigned	Ctrl+F1
PA1	Not assigned	PageUp or Ctrl+One
PA2	Not assigned	PageDown or Ctrl+2
Erase EOF	Not assigned	End
3270 End of Field	End	Not assigned
Home	Home	Home
Insert	Insert	Insert
New Line	Shift+Enter	Shift+Enter
PF12	F12	F12 or Alt+2
PF24	Shift+F12	Shift+F12 or Alt+4
BackTab	Shift+Tab	Shift+Tab
Cursor Select	Not available	Ctrl+F3
Logical Not (“¬”)	Not assigned	Not available
Stick (“|”)	–	–

Figure 3: Default keyboard mapping

Speaking of Reset, whenever you lock the keyboard by typing characters where you shouldn’t, HOD tells you to move the cursor. But, of course, the cursor keys are locked. You can move the cursor with the mouse, by moving and clicking.

THE COMPARISONS

Figure 1 compares some of the features that differentiate the two products, particularly the defaults. Figure 2 looks at how you perform key functions. And Figure 3 focuses on the default keyboard mappings. Admittedly, you can customize the keyboard mapping that your users see when they first use the product. But, no matter how it's done, keyboard mapping can make or break initial user acceptance of a terminal emulator.

NOTES AND SUCH

One final comment needs to be made. Menus (shortcut or menu bar) are optional in Reflection, and there are four menu levels: Basic (default), Intermediate, Advanced, and Administrator. They are defined for each session on the Applet tab of the Deployment Director within the Administrative WebStation. Some of the menu options described in Figure 2 are not available on the Basic level of menu.

Armand Minet
(Canada)

Host Access Client Package – HOD component

Continuing our series on 3270 mainframe terminal emulators, we look at IBM's Web-based Host On-Demand (HOD), now a component of Host Access Client Package. This article looks at HOD installation and customization on a Windows 2000 Server with a Windows XP Professional client, using tn3270e to access an OS/390 mainframe across the Internet.

INSTALLATION CDs

Version 2.0 of Host Access Client Package (HACP) for Multiplatforms comes in two packs of CD-ROMs, one entirely Personal Communications for OS/2 in various languages, the other Windows and other platforms. The CD-ROM labelled WebSphere Host On-Demand for Multiplatforms is clearly the one of interest, although the Host Access Toolkit also looks interesting. There are also HOD CD-ROMs for OS/400 and AIX/Unix.

A couple of things to note. First, each component, including HOD, has its own version number. HOD Version 6.0 comes with HACP Version 2.0. Second, as explained in the multilingual booklet, the *Getting Started* manual is on the CD-ROM:

```
d:\doc\en\doc\install\install.pdf and .html
```

Inserting the CD-ROM also brings up an HOD logo that stays for 50 seconds then displays a Welcome dialogue box with a menu in the left sidebar with the following options: installation prerequisites, view documentation, install, visit our Web site, and deployment Wizard.

THE INSTALL MANUAL

As you'd expect, View Documentation loads install.html into your Web browser. There are a few points to note, as follows:

- The key piece of information in the Firewall chapter is that HOD's Service Manager uses port 8999 by default. Testing was done with the firewall built into Belkin's new four-port cable/ DSL gateway/router, which didn't interfere with HOD.
- *Support client operating systems* in the 'Planning' chapter doesn't list Windows XP, though you're referred to the readme file for updates.
- *Before installing Host On-Demand* earlier in the same chapter begins with an invalid link to the readme file – invalid because it assumes that readme is in the same directory, which it isn't, but at:

  d:\readme\en\readme.html

  There you'll also find no mention of Windows XP, but it does begin with a US link to the latest information at:

  http://www.ibm.com/software/webservers/hostondemand

On that Web page, under Operating systems, Windows XP isn't listed.

Back to the *Getting Started* manual. I clicked the *Installing on Windows NT and Windows 2000* link in the Installing chapter, and chose a GUI install. You can also create a response file and use Windows InstallShield in silent mode. There you'll find that the obvious is stated:

- A Web server is required
- Installation must occur from an Administrator ID.

Eight simple install steps follow.

THE ACTUAL INSTALL

The first notable occurrence once installation begins is the appearance of two Destination Folder prompts. The defaults are c:\hostondemand for server files, and c:\hostondemand\hod for Web-accessible files, including code downloaded to the client. Next, the install wizard estimates disk space requirements of 210MB, even though the manual lists 174MB. Installation then continues without human intervention until a Web Server configuration dialogue box appears without any further explanation:

```
The Web servers listed below were detected on your system.
o Microsoft Internet Information Server
```

Leaving the check mark beside IIS and hitting the Next button seems the logical thing to do. The next dialogue box (Specify Publish-Directory Alias) creates the Virtual Directory for c:\hostondemand\hod, where HOD puts the Web pages that users will need to access HOD:

```
Specify the alias for your 'Publish' directory.
The alias will be part of the URL used by download clients to access
Host On-Demand. For example:
http://[yourwebserver]/hod/[client_name].html
```

You're prompted merely to accept the default Virtual Directory name of hod or specify one of your own choosing. Once you do, an Information dialogue box immediately appears:

```
Web server configuration is complete!
```

```
You must stop and restart the web server for your changes to take
effect.
```

CUSTOMIZATION

When you hit OK, the use of Port 8999 by Service Manager is confirmed, and the install itself is complete. Unless you remove the check marks, you then go through product registration, after which your Web browser is pointed at basic configuration steps in c:\hostondemand\hod\en\help\basicsteps.html. It begins by describing the HOD Deployment Wizard, which allows you to choose from three configuration models: HTML-based, configuration server-based, and combined.

Each is also a link that you can click on for more information, and the ‘Planning’ chapter of the *Getting Started* manual is also helpful. The configuration server-based model offers centralized management, but there are some major issues associated with deploying it in large organizations where centralized management is most needed:

- It is time-consuming to annually create and maintain user IDs.
- Automatically creating HOD IDs from Windows user names requires them to be in a Windows domain.
- Unless you have a Single Sign-On (SSO) product, you’ll undoubtedly want to have users automatically logged on to HOD using their Windows user name.
- Performance can be an issue with large numbers of users accessing personalized configurations on the configuration server.

That second requirement, Windows NT/200*x* domains, is a major one if you don’t already have them in place, especially in a test network where workgroups are the norm. Suffice it to say that a strong understanding of domains and your current network are the only way to avoid ending in failure. For testing purposes, you may want to stick with the default HTML-based model, as I did for this evaluation.

DEPLOYMENT WIZARD

The Deployment Wizard is started from Start/Programs/IBM Host

On-Demand/Administration/Deployment Wizard. Once you've created a session with the Add button on the wizard's Host Sessions dialogue box, the message below the rest of the buttons reads:

```
To configure run-time options, such as window size, position, colours,
toolbar, etc., click Start
```

As its name implies, Start actually begins the host session you just defined. Even if you have no immediate need to change the defaults, it provides an excellent way to test the validity of the session you set up.

With the HTML-based model, the Deployment Wizard must be run to create each HTML page. When you're done, you can Close or Restart Wizard to create another page.

TESTING HOD

And that is all there is to it. To test HOD, go to another workstation, start up a Web browser and type the URL of the Windows 2000 Server's Web server, the HOD virtual directory set-up during install (default=HOD), and the HTML page file name you just created with the Deployment Wizard. In my case:

```
http://192.168.2.2/HOD/BenchMark.html
```

Unlike Unix, case is irrelevant in Windows Web servers, so that URL could have been coded with all lower-case. If you chose the Deployment Wizard defaults, expect the usual Security Warning dialogue box for IBM Host On-Demand 6.0 Cached Client, and hit the Yes button. You'll also get a browser dialogue box:

```
The cached client will now be installed.
Do not browse to a different HTML page until install completes.
```

Hit the OK button, and less than 5 seconds later you'll see the Security Warning twice more, again for the Cached Client and also for IBM Screen Customizer 2.0.60, before the actual install begins. It's done in less than 15 seconds, and is followed by another browser dialogue box:

```
Cached client installation complete.
Restart your browser before using the cached client.
```

CACHED CLIENT AND TRANSPARENT CHANGES

This time, after restarting your browser and entering the same URL, the Security Warning dialogue box appears just once for the Cached Client. Hit Yes and a few seconds later you'll see an icon for the host session you created. Double-click on it and your session begins, in a browser window of its own.

The Cached Client survives a reboot, works on all HOD sessions you create, and doesn't need to be re-cached even if you modify a session definition. When you do change a session with the Deployment Wizard, using the Properties button on the Host Sessions dialogue box, it's transparent to the user in the sense that nothing special happens on next use by a client workstation.

*Editor's note: The article immediately preceding (pp 3-6) presents a side-by-side comparison of HOD and WRQ's Reflection for the Web; Reflection installation details were covered in the last issue (*TCP/SNA Update *45, March 2002 pp 56-69).*

Armand Minet
(Canada)

Policy and the z/OS Communications Server Policy Agent

Companies have always had a business management policy – even if they didn’t formalize it. When I joined a large corporation several decades ago, it was an unwritten policy that the executive in the large corner office received his mail first, had coffee delivered by the secretary, had his memos typed first, and so on.

Today’s use of networking and the Internet has produced a real need for a more formalized – and arguably more useful – policy definition. Ideally, a system or network administrator could provide priority (fast) access to the executive, allow only key financial officers access to unannounced business performance numbers, prohibit access to sensitive data from home and (other) unsecured sites, and temporarily discontinue access for an employee off on a sabbatical.

Policy is generally made up of multiple actions (authorizations and/or restrictions) that can include:

- Control of the overall (TCP) traffic flow through the internal and external networks.
- Prioritizing of application traffic (eg tn3270) critical to the company operation.
- Determining which users have access to company resources.
- Delivering differentiated services to users according to their needs.
- Balancing voice and video traffic with data.

Policy manifests itself in the setting and control of bandwidth management, routing, Virtual Private Networks, firewalls, and caching.

For companies with a moderate number of employees, establishing a policy quickly becomes untenable without a succinct and organized way to define the policy and then have that policy result in automatic and specific control of corporate resources. Many networking providers have recognized that a network administrator’s attempt to manage policy by individual control of individual elements is destined to fail and have attempted (with some collaboration) to provide a solution.

Most people realize that good reporting has five Ws: who, what, where, when, and why. This article provides a generic overview of the components required to successfully define, maintain, and implement (enforce) a policy. With this background, it then discusses the capabilities of the Policy Agent within z/OS Communications Server.

POLICY MANAGER

A policy manager translates company policies regarding internal and external privileges to create concise network control definitions. These definitions help control the connections (bandwidth) among the devices that make up the network that connects the user to the application. A policy manager implementation can be distributed and hierarchical in structure. For example, a general corporate policy can be combined with more specific policies for individual departments to provide, as an example, the policy for the financial department mentioned earlier. Several components make up the policy manager: a Policy Repository (along with an editor), a Policy Decision Point, and a Policy Enforcement Point.

Policy information can reside on a flat-file, an administrative server, or a directory server database. Typically, this information resides on an LDAP (organized) database server provided by companies such as IBM or Microsoft. The policy repository contains all relevant information for the domain and is generally grouped in a distributed or hierarchical manner. The policy is represented in the repository as a set of rules. The syntax and semantics of the policy representation has to be standardized to allow operation among different vendors. When the policy is in an LDAP directory, a schema defines the types of entry, the attributes for each type of entry, and the relationships between different entry types.

The policy schema proposed by the IETF collaboration consists of three main types of entry: policies, conditions, and actions. A policy entry rule is generally the semantic ‘IF Condition A, THEN Action A’. Condition is a clause used to identify the packet sub-stream to which the policy rule applies. Action specifies what should happen to the packet sub-stream. Policy conditions are generally grouped into categories for items such as hosts, users, and applications. Simple policy conditions can be identified IP header fields (ie source and destination address), protocol, or UDP header (ie source and destination port numbers). Actions are service specific – an RSVP action would qualify a reservation request, and a differential services action would specify an outgoing packet priority field value. Other repository information can identify when the policy is valid and specify priority hierarchies to resolve conflicting policies.

The policy decision point determines what actions should be applied to which packets. The decision point interprets policy rules for one or more enforcement points, based on condition information such as packet sub-stream content, network conditions, and dynamically allocated addresses. For example, the policy decision point could decide whether to honour a bandwidth reservation request based on originator identity.

The policy enforcement point is the component that monitors the packets and is responsible for enforcing the policy. Typically, the enforcement point is located within the packet-forwarding component of a network server or access router. Typical enforcements include filtering, marking, traffic shaping, and rate control.

Some network components are powerful and flexible enough to enable policy decision and policy enforcement to be combined. Most network servers and high-end routers with a dedicated control processor make their own decisions as well as enforcing them.

OS/390 Version 2 Release 7 quietly introduced a policy agent that combines decision and enforcement functions. Some of the basic policy agent functions were available with Version 2 Release 5, but this release added necessary structure. Subsequent releases have added incremental enhancements.

IBM POLICY AGENT

Communications Server allows network administrators to define policy in two ways. One is through a policy configuration file, and the other is through an LDAP directory-based repository. The Policy Agent can access the service policy configuration files or go to an LDAP server, or both, to retrieve service policy entries/objects. Since policy may contain sensitive information regarding network resource allocation and SLA information, Secured Socket Layer (SSL) security is provided to protect LDAP server access. The policy agent configuration file can itself be protected by RACF.

IBM z/OS policy core schema closely follows the policy schema effort that is currently on-going in the IETF (recall the collaborating vendors). This schema will contain all the functions necessary to allow the construction of complex policy (eg nesting of policy groups). The Internet Draft (a working document within the IETF) contains a snapshot of this core schema. This snapshot is what the IBM Communications Server implementation was based on. The core schema defines general structural classes for defining policies, but specific applications, such as QoS policy, require specialized subclasses, and IBM defined (its own) auxiliary classes that go beyond the core Policy Condition and Policy Action. (IBM has committed to amend its definitions to agree with the final IETF definitions.)

As mentioned above, policy information can reside either on a configuration file or on an LDAP server. Defining policy in a configuration file uses the format of policy-condition-policy-action statements described above. In most cases, there’s a one-to-one mapping between attributes that can be specified by LDAP schema or the configuration statement.

- An IBM auxiliary subclass contains attributes that describe quality of service actions in terms of observable or measurable behaviours for both RSVP and Differentiated Services:
  - Applications can explicitly reserve bandwidth through RSVP using RSVP API. The (IBM) RSVP action attributes enable network administrators to limit how much bandwidth an application can reserve for an RSVP flow. In addition, the number of flows can be specified.
  - Attributes in the (IBM) subclass give network administrators

differentiated service control of service levels. Flexibility ranges from setting how traffic is treated end-to-end, to being able to control the throughput of individual TCP connections.

- A Web content-based QoS feature in Communications Server, along with IBM WebSphere and z/OS Work Load Manager (WLM), offers enterprise network administrators and service providers a function that they can leverage for a competitive advantage in Web-enabling their business or services. That is, there are a variety of contents that are delivered over the Web, and each of them can use a different QoS level. For instance, an HTTP request for downloading a file (eg browsing) should have a lower QoS level than a purchasing transaction that's being processed.

WebSphere can use the WLM function to route incoming requests, based on the requested URL, to the appropriate Communications Server WLM enclave. Different enclaves can have different CPU resource allocations and server delay time goals. Web content-based QoS classifies and assigns different network service levels based on the requested URL. Remember that the URL identifies the kind of HTTP transaction and the content to be delivered (eg file retrieval or conduct of business). The two components, server processing and network QoS, ensure that a Web transaction is assigned a QoS level that can satisfy end-to-end SLA.

The Communications Server component responsible for URL classification is called the Fast Response Cache Accelerator (FRCA). When it receives an inbound HTTP request, FRCA parses the request, retrieves the URL, and invokes the policy component that, along with other criteria (eg IP addresses, port number), classifies the request. The URL policy can be specified in the (IBM-defined) extension of the policy schema. If a classified policy rule exists, the corresponding QoS service level is assigned to the returned data. If no matching rule for the URL is found, the QoS level for the associated HTTP connection (if one exists) will be assigned to the returned data. Note that other policy attributes can be used to differentiate the QoS service level, for example the client or server's IP address. Different QoS service levels can be assigned to the same data requested by different clients (eg premium versus regular clients/users).

TRAFFIC REGULATION

Traffic regulation enables network administrators and service providers to avoid two common problems that can bring network services to a halt. The first is caused by greedy clients that send multiple requests to a server in an attempt to get better service. The second is a flooding attack where a server is bombarded with so many connection requests that little useful service can be rendered – a denial of service attack. Traffic regulation policies need to be defined within the policy configuration file to enable this function. (Note that traffic regulation policies are not fully supported until z/OS V1.2.) Traffic regulation is applied on a per application basis; however, a single traffic regulation policy should be applied to all applications.

Traffic regulation has the following options:

- Define the total number of concurrent connections allowed for an application (eg HTTP port 80).
- Gather statistical information about the normal number of connections to an application server for capacity planning purposes.
- Limit a client to a defined percentage of remaining available connections and deny a client’s request when it exceeds this threshold percentage.
- Log when a client’s request exceeds the threshold but the request (for some unknown reason) is honoured.

QOS MONITORING

Once QoS service levels are defined and (hopefully) enforced, proactive monitoring of performance can protect client Service Level Agreements. An SLA Management Information Base can define thresholds for different QoS performance attributes (eg maximum-rate, minimum-rate, maximum-delay) so that deviations can be sent to the right SNMP manager for appropriate action. The policy information reported in the SLA MIB can also be effectively used for accounting/billing data (eg bytes/packets sent/received, in-profile versus out-of-profile counts).

CONCLUSION

Policy will have an ever-increasing importance to differentiation among company networks. With the Policy Agent within Communications Server, IBM has provided a valuable tool that can be used today to implement network policy. However, all that's provided is the configuration file; IBM hasn't integrated the policy agent with the necessary policy manager to build the LDAP database. We must wait and hope that Tivoli or some other management provider teams up to create the repository that simplifies the definition and management of the policy. However, despite its shortcomings, Policy Agent should be used to establish network policy. Why? Because it provides the required functions.

Richard Tobacco
(USA)

A program to extract information from a Front End Processor

The program presented here, PPGRDFEP, extracts information from a Front End Processor (FEP) that describes an NCP's environment, such as its type of controller and its version number. PPGRDFEP acts as a Program Operator. It requires, as input, the seven-character name assigned to the NCP load module that was fabricated by the NCP generation process. There will be one name for each FEP. It is passed as input to PPGRDFEP via a SYSIN DD statement. Each name must begin in column one.

PPGRDFEP issues "D NET,NCPSTOR,..." commands for the name of each NCP load module encountered in its processing, and then processes all VTAM responses generated by those commands. The name provided as input is used as an operand on the ID keyword in the display command. Its processing is predicated upon receipt of IST245I and IST241I messages in the proper order. It is a moderately temperamental program and may enter a wait state unless it receives precise responses from VTAM. The storage path that it follows to extract specific information is documented within the code that does it, so I won't duplicate that information here.

In order for PPGRDFEP to extract information from an FEP, an APPL must first be properly defined for its use, and activated. The name specified on the APPL control statement *must* match the name of the step that invokes PPGRDFEP, otherwise an OPEN failure will occur. A switch, named PPGSW, controls the issuance of error and normal messages returned from VTAM's services. It's initially set to a value of X'02', which prevents the transcription of those messages. Set PPGSW to X'00' if messages of this kind are needed for analysis. This is dynamically doable by specifying PARM=0 on the EXEC statement invoking PPGRDFEP.

The following VTAM control statements can be used to define an APPL for PPGRDFEP. Remember that the name on the APPL statement *must* match the name of the step that invokes PPGRDFEP. In the examples provided here, it is CHART. PRTCT is a password and must match the one defined in PPGRDFEP at the statement named PPGPSWD. It may also be specified as a parameter on the EXEC statement, after the specification for PPGSW, if present. CHARTAPL must be active before the invocation of PPGRDFEP, otherwise an OPEN error will occur and FEP information will not be produced. After the following statements have been added to member CHARTAPL in SYS1.VTAMLST, CHARTAPL is activated by issuing the following operator command: V NET,ACT,ID=CHARTAPL

```
**********************************************************************
**                                                                  **
**       PROVIDE PROGRAM OPERATOR INTERFACE ( POI ) FOR PPGRDFEP    **
**                                                                  **
**********************************************************************
CHARTAPL VBUILD TYPE=APPL
**********************************************************************
CHART    APPL  AUTH=(SPO),EAS=1,PRTCT=CLAM
```

Note that the value defined in AL1(4) on that statement must be the exact length of the password provided.

My endeavours to extract information from an FEP were successful primarily because of the superlative help that I received from two fellows who monitor IBM’s Supportline, a question and answer type service, for VTAM – Gilles Lalonde and Marcus Cook. They were patient and understanding, and gave detailed responses about the location of information that’s available within an FEP and the exact pathway through NCP control blocks that was required to arrive there, something sorely missing from NCP and VTAM’s control block manuals. Although they didn’t help with questions regarding programming requirements, stating that it was beyond the scope of Supportline’s service, they offered to refer me to someone with VTAM experience who would work with me through a billable service. I refused, since Supportline already costs us several thousand dollars a month. Be that as it may, they provided me with enough information for me to accomplish my task and were nudging their threshold of service a little beyond what they should have been providing.

PPGRDFEP has been successfully executed on OS/390 release 2.9 with NCP 7.6 and VTAM 4.3. VTAM requires that it be link-edited into an authorized library with an option of AC=1 specified. PPGRDFEP requires no special programming privileges – that is, it doesn’t enter supervisor state, nor does it modify or access protected storage.

PPGRDFEP can be invoked with the following JCL:

```
//CHART    EXEC PGM=PPGRDFEP
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
seven-character name of NCP load module
```

An example of an EXEC statement requesting both a display of VTAM’s responses and a change to CHARTAPL’s password is as follows:

```
//CHART EXEC PGM=PPGRDFEP,PARM='Ønewpswd'
```

Presented below is sample output created from an NCP load whose name is NCPBBOX:

```
NCPBBOX
3745/61A 3746/9ØØ  TWIN-DUAL   CCU B  16MB  ACF/NCP VØ7RØ8MØ1  LINK-
EDIT: 13:27 11/23/2ØØ1
```

PPGRDFEP SOURCE

```
TITLE 'PPGRDFEP - MAP A NETWORK'S SUBAREAS'
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*  THE PURPOSE OF THIS ROUTINE IS TO EXTRACT DATA FROM A FRONT END    *
*  PROCESSOR(FEP).                                                    *
*  IT USES A PROGRAM OPERATOR INTERFACE(POI) TO ISSUE VTAM COMMANDS   *
*  AND THEN PROCESS THE RESPONSES TO THOSE COMMANDS IN ORDER TO       *
*  OBTAIN INTERNAL INFORMATION REGARDING FRONT END PROCESSORS (FEPs). *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE 2
PPGRDFEP CSECT
         SPACE
PPGRDFEP AMODE 31
PPGRDFEP RMODE 24
         SPACE
         PRINT NOGEN
         SPACE
         USING PPGRDFEP,R12        ESTABLISH PPGRDFEP ADDRESSABILITY
         SPACE
         STM   14,12,12(R13)       PRESERVE REGISTERS AT ENTRY TIME
         LR    R12,R15             PRIME BASE REGISTER
         SPACE
         LA    R15,PPGSAVE         POINT TO A NEW SAVE AREA
         ST    R13,PPGSAVE+4       CHAIN SAVE AREAS - BACKWARD
         ST    R15,8(R13)                           - FORWARD
         LR    R13,R15             POINT TO CURRENT SAVE AREA
         ST    RØ,PPGHOLDØ         STOW CONTENTS OF GPR 1
         SPACE
         L     R11,PPGPARM4        POINT TO OUTPUT AREA
         USING PPGDSECT,R11        ESTABLISH PPGDSECT ADDRESSABILITY
         SPACE
         CLC   PPGLOVE,PPGHOLDØ    TEST IF PPGRDFEP WAS CALLED
         BNE   PPGEXECP            BRANCH IF NOT
         SPACE
         MVC   PPGPARM1(2Ø),Ø(R1)  COPY INPUT PARAMETERS
         LM    R2,R6,Ø(R1)         PRIME REGISTERS WITH THEIR ADDRESSES
         MVC   PPGSW,Ø(R6)         COPY VALUE OF PPGSW FROM PPGCHART
         MVC   PPGPSWD(9),Ø(R3)    COPY VALUE OF PPGPSWD FROM PPGCHART
         MVC   PPGNCPID,Ø(R2)      COPY VALUE OF PPGNCPID FROM PPGCHART
         L     R11,PPGPARM4        POINT TO OUTPUT AREA OF CALLING PGM
         B     PPCEBASE            CONTINUE...
         EJECT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*        PROCESS PARAMETERS SPECIFIED ON THE EXEC STATEMENT.          *
*        IF FIRST DIGIT IS NOT A ZERO, THEN IT'S A PASSWORD.          *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         SPACE 1
PPGEXECP L     R1,Ø(R1)            POINTER TO PARM LENGTH FIELD.
         LH    R2,Ø(R1)            LENGTH INTO 2.
         LTR   R2,R2               TEST IF ANY PARMS ARE PRESENT
         BZ    PPCOPEN             BRANCH IF THERE ARE NONE
         SPACE
         CLI   2(R1),C'Ø'          TEST IF ERROR MESSAGES DESIRED
         BL    PPGCALYN            BRANCH IF NOT
         BE    PPGSETSW            BRANCH IF SO
PPGPRMER WTO   'OIR2Ø2E PARM ERROR - MUST BE Ø, PSWD(<=8), OR ØPSWD'
         LA    R15,8               SET AN UNSUCCESSFUL RETURN CODE
         PR    R14                 BACK TO DUST
         SPACE
PPGSETSW MVI   PPGSW,Ø             INDICATE VTAM ERROR CODES DESIRED
         BCTR  R2,Ø                REDUCE LENGTH BY ONE
         LA    R1,1(R1)            PSEUDO START OF PASSWORD
         SPACE
         LTR   R2,R2               TEST IF ANY ADDITIONAL PARAMETERS
         BZ    PPCOPEN             BRANCH IF NONE
         SPACE
PPGCALYN C     R2,=F'8'            TEST IF PASSWORD EXCEEDS EIGHT BYTES
         BH    PPGPRMER            BRANCH IF SO
         MVC   PPGPSWD+1(8),PPGRECV BLANK PASSWORD
         SPACE
         STC   R2,PPGPSWD          STOW LENGTH OF PASSWORD IN PARM
         BCTR  R2,RØ               REDUCE LENGTH OF PASSWORD FOR MOVE
         EX    R2,PPGETAID         COPY PASSWORD TO PARAMETER AREA
         SPACE
PPCOPEN  OPEN  (PPHODCB,OUTPUT,PPHIDCB) PREPARE DATA SET FOR OUTPUT
         SPACE
PPCREAD  MVI   PPHCC,C' '          SET INITIAL BLANK
         MVC   PPHCC+1(PPHDLEN-1),PPHCC CLEAR IN-OUT AREA
         GET   PPHIDCB,PPHCC+1     OBTAIN AN NCPLOAD'S NAME
         MVC   PPGNCPID,PPHCC+1    COPY IT INTO COMMAND AREA
         PUT   PPHODCB,PPHCC       TRANSCRIBE IT
         EJECT
************************************************************************
*                                                                      *
*        THIS SUBROUTINE EXTRACTS INFORMATION FROM AN FEP              *
*                                                                      *
************************************************************************
         SPACE
PPCEBASE MVC   PPGSIZE,PPGCØØ4     SET AMOUNT OF DATA THAT IS REQUIRED
         MVC   PPGSTORG,PPGCØØ38   SET ADDRESS OF INITIAL DATA IN FEP
         SPACE
         LA    R4,PPGACB           POINT TO ACB
         USING IFGACB,R4           ESTABLISH ACB ADDRESSABILITY
         LA    R5,PPGRPL           POINT TO ACB
         USING IFGRPL,R5           ESTABLISH RPL ADDRESSABILITY
         SPACE
         TM    ACBOFLGS,ACBOPEN    TEST IF PATH PREVIOUSLY TRAVELLED
         BO    PPGLNDS             BRANCH IF SO
         SPACE
         OPEN  PPGACB              PREPARE TO INTERFACE WITH VTAM
         LR    R3,R15              PRESERVE RETURN CODE FROM OPEN
         TM    PPGSW,2             TEST IF ERROR MESSAGES ARE DESIRED
         BO    PPGIGNOR            BRANCH IF NOT
         SPACE
         MVC   PPHVRBLK(11),=CL11'ACBERFLG=GL'
         UNPK  PPHVRBLK+9(3),ACBERFLG(2) ALTER RADIX OF RETURN CODE
         MVI   PPHVRBLK+9+2,C' '  REMOVE DE DETRITUS
         TR    PPHVRBLK+9(2),PPGTRANS-24Ø MAKE RETURN CODE PRETTY
         BAS   R2,PPGSCRIB         TRANSCRIBE DATA
         SPACE
PPGIGNOR LTR   R15,R3              TEST RETURN CODE
         BE    PPGLNDS             BRANCH IF OPEN WAS SUCCESSFUL
         OI    PPGSW,1             SET INHIBIT-PROCESSING SWITCH
         B     PPGLEAVE            EXIT
         EJECT
************************************************************************
*                                                                      *
*      LOCATE THE PRODUCT SET IDENTIFIER(PSI) CONTROL BLOCK VIA        *
*      LOW CORE LOCATION X'38' IN FEP'S STORAGE.                       *
*                                                                      *
*      LOW CORE X'38' ===> WORD DIRECT ADDRESSABLE STORAGE(XDA)        *
*      XDA + X'5C'    ===> EXTENDED HALFWORD DIRECT ADDRESSABLE(HWX)   *
*      HWX + X'5C'    ===> PRODUCT SET IDENTIFIER(PSI)                 *
*                                                                      *
************************************************************************
         SPACE
PPGLNDS  LA    R7,PPGRECV          POINT TO RESPONSE'S RETURN AREA
         USING PPGRECVD,R7         ESTABLISH PPGRECVD ADDRESSABILITY
         SPACE
         LA    RØ,PPGLAST          OBTAIN ADDRESS OF LAST WORD OF DATA
         ST    RØ,PPGLASTA         STOW IT
         SPACE
         BAS   R8,PPGSKMAN         DISPLAY LOW CORE ADDRESS X'38'
         SPACE
PPGLNDSØ BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG1            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
PPGNMSG1 CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGLNDSØ            BRANCH IF NOT
         SPACE
         CLC   PPGCORE,PPGSTORG    TEST IF DESIRED ADDRESS
         BNE   PPGLNDSØ            BRANCH IF NOT
         MVC   PPGVALUE,PPGXDA     STOW ADDRESS OF NEXT DISPLAY CMD
         MVC   PPGLCXDA,PPGXDA     RETAIN ADDRESS OF XDA FOR FUTURE USE
         SPACE
         LH    RØ,PPGX5C           FETCH OFFSET TO HWX
         BAS   R8,PPGCONV          COMPUTE ADDRESS FOR NEXT DISPLAY CMD
         SPACE
         BAS   R1,PPGSKIP          DISCARD REMAINING MESSAGES
         EJECT
************************************************************************
*                                                                      *
*        FETCH ADDRESS OF HWX FROM WITHIN THE XDA                      *
*                                                                      *
************************************************************************
         SPACE
         BAS   R8,PPGSKMAN         DISPLAY WORD DIRECT ADDRESSABLE CORE
         SPACE
PPGSKIP1 BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG2            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
PPGNMSG2 CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGSKIP1            BRANCH IF NOT
         SPACE
         CLC   PPGCORE,PPGSTORG    TEST IF DESIRED ADDRESS
         BNE   PPGSKIP1            BRANCH IF NOT
         SPACE
         L     R1,PPGVHOLD         FETCH ADDRESS OF HWX
         N     R1,PCPØØØF          REMEMBER ONLY THE LOW-ORDER VALUE
         SRL   R1,2                DIVIDE IT BY FOUR
         MH    R1,PPGH9            COMPUTE OFFSET IN HWX TO PSI'S ADDR
         LA    R1,PPGRESP(R1)      POINT TO ADDRESS OF PSI
         MVC   PPGVALUE,2(R1)      STOW CORE ADDR FOR NEXT DISPLAY CMD
         SPACE
         LH    RØ,PPGX5C           FETCH OFFSET TO PSI
         BAS   R8,PPGCONV          COMPUTE ADDRESS FOR NEXT DISPLAY CMD
         SPACE
         BAS   R1,PPGSKIP          DISCARD REMAINING MESSAGES
         EJECT
************************************************************************
*                                                                      *
*        FETCH ADDRESS OF PSI FROM WITHIN THE HWX                      *
*                                                                      *
************************************************************************
         SPACE
         BAS   R8,PPGSKMAN         DISPLAY PRODUCT SET IDENTIFIER INFO
         SPACE
PPGSKIP3 BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG3            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
PPGNMSG3 CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGSKIP3            BRANCH IF NOT
         SPACE
         CLC   PPGCORE,PPGSTORG    TEST IF DESIRED ADDRESS
         BNE   PPGSKIP3            BRANCH IF NOT
         SPACE
         L     R1,PPGVHOLD         FETCH ADDRESS OF HWX
         N     R1,PCPØØØF          REMEMBER ONLY THE LOW-ORDER VALUE
         SRL   R1,2                DIVIDE IT BY FOUR
         MH    R1,PPGH9            COMPUTE OFFSET IN HWX TO PSI'S ADDR
         LA    R1,PPGRESP(R1)      POINT TO ADDRESS OF PSI
         MVC   PPGVALUE,2(R1)      STOW CORE ADDR FOR NEXT DISPLAY CMD
         SPACE
         MVC   PPGSIZE,PPGPSIZE    REQUEST THAT VTAM RETURN MORE DATA
         SR    RØ,RØ               NO OFFSET REQUIRED
         BAS   R8,PPGCONV          COMPUTE ADDRESS FOR NEXT DISPLAY CMD
         SPACE
         BAS   R1,PPGSKIP          DISCARD REMAINING MESSAGES
         EJECT
***********************************************************************
*                                                                     *
*        OBTAIN THEN COMPRESS PSI DATA                                *
*                                                                     *
***********************************************************************
         SPACE
         BAS   R8,PPGSKMAN         DISPLAY PRODUCT SET IDENTIFIER INFO
         SPACE
PPGSKIP4 BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG4            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
PPGNMSG4 CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGSKIP4            BRANCH IF NOT
         SPACE
         CLC   PPGCORE,PPGSTORG    TEST IF DESIRED ADDRESS
         BNE   PPGSKIP4            BRANCH IF NOT
         SPACE
         LA    R9,PPGLASTW         POINT TO ADDRESS OF HOLD AREA
PPGCARP  L     R1,PPGVHOLD         FETCH ADDRESS OF HWX
         N     R1,PCPØØØF          REMEMBER ONLY THE LOW-ORDER VALUE
         SRL   R1,2                DIVIDE IT BY FOUR
         MH    R1,PPGH9            COMPUTE OFFSET TO FIRST FULL WORD
         LA    R1,PPGRESP(R1)      POINT TO ADDRESS OF FIRST OF DATA
         MVC   PPGVALUE,2(R1)      STOW CORE ADDR FOR NEXT DISPLAY CMD
         SPACE
PPGBLOCK TR    Ø(8,R1),PPGHEX2B    REMOVE ZONES FROM OFFSET
         PACK  Ø(5,R9),Ø(9,R1)     COMPRESS OFFSET
         LA    R1,9(R1)            POINT TO NEXT WORD OF DATA
         LA    R9,4(R9)            POINT TO NEXT OUTPUT AREA
         C     R1,PPGLASTA         TEST IF FINISHED WITH THIS BLOCK
         BNH   PPGBLOCK            FORMAT NEXT WORD
         EJECT
         CLC   PPGISTMG,PPG241I    TEST IF COMPLETED COMMAND MESSAGE
         BE    PPGDOUTS            BRANCH IF SO
         SPACE
         BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         MVC   PPGVALUE,PPGCORE    POINT TO CURRENT CORE ADDRESS
         SR    RØ,RØ               ADDRESS REQUIRED IS ABSOLUTE
         BAS   R8,PPGCONV          CONVERT ADDRESS TO BINARY
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG5            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
PPGNMSG5 CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGREMAN            BRANCH IF NOT
         SPACE
         B     PPGCARP             PROCESS RESPONSE
         SPACE
PPGREMAN CLC   PPGISTMG,PPG241I    TEST IF COMPLETED COMMAND MESSAGE
         BE    PPGDOUTS            BRANCH IF SO
         BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         B     PPGREMAN            BRANCH IF NOT
         EJECT
***********************************************************************
*                                                                     *
*        OBTAIN SIZE OF FEP STORAGE FROM XDA CONTROL BLOCK            *
*                                                                     *
***********************************************************************
         SPACE
PPGDOUTS LH    RØ,PPGX6Ø           FETCH OFFSET WITHIN XDA TO HI-ADDR
         MVC   PPGVALUE,PPGLCXDA   SET ADDRESS OF NEXT DISPLAY COMMAND
         MVC   PPGSIZE,PPGCØØ4     SET AMOUNT OF NCP'S DATA REQUIRED
         BAS   R8,PPGCONV          COMPUTE ADDRESS OF NEXT DISPLAY CMD
         SPACE
         BAS   R8,PPGSKMAN         DISPLAY FAX ADDRESS
         SPACE
PPGCMRCP BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNOMSG            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
```

```
PPGNOMSG CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGCMRCP            BRANCH IF NOT
         SPACE
         CLC   PPGCORE,PPGSTORG    TEST IF DESIRED ADDRESS
         BNE   PPGCMRCP            BRANCH IF NOT
         SPACE
         L     R1,PPGVHOLD         FETCH ADDRESS OF FAX
         N     R1,PCPØØØF          REMEMBER ONLY THE LOW-ORDER VALUE
         SRL   R1,2                DIVIDE IT BY FOUR
         MH    R1,PPGH9            COMPUTE OFFSET IN HWX TO PSI'S ADDR
         LA    R1,PPGRESP(R1)      POINT TO SIZE OF STORAGE
         SPACE
         TR    Ø(8,R1),PPGHEX2B    REMOVE ZONES FROM OFFSET
         PACK  PPGVHOLD,Ø(9,R1)    COMPRESS OFFSET
         SR    RØ,RØ               CLEAR REMAINDER REGISTER
         L     R1,PPGVHOLD         FETCH SIZE OF STORAGE IN BYTES
         D     RØ,PAT1MEG          COMPUTE SIZE IN MEGABYTES
         LTR   RØ,RØ               TEST IF THERE IS A REMAINDER
         BE    PPGNOREM            BRANCH IF NOT
         AH    R1,PPGH1            INCREMENT MEGABYTE VALUE BY ONE
PPGNOREM CVD   R1,PPGDOUBL         ALTER RADIX TO PACKED DECIMAL
         MVC   PPGHMB,PPGPATCU     COPY EDIT PATTERN INTO A HOLD AREA
         ED    PPGHMB(4),PPGDOUBL+6 BEAUTIFY ITS CONTENTS
         SPACE
         BAS   R1,PPGSKIP          DISCARD REMAINING MESSAGES
         EJECT
************************************************************************
*                                                                      *
*        OBTAIN DATA IN FULL-WORD DIRECT ADDRESSABLE EXTENSION (FAX)  *
*        LOW-CORE X'38' ==> XDA                                        *
*        XDA + X'78'    ==> FAX                                        *
*        FAX + X'2C'    ==> UTS                                        *
*        UTS CONTAINS A DESCRIPTION OF A CCU'S OPERATING MODE         *
*                                                                      *
************************************************************************
         SPACE
         LH    RØ,PPGX78           FETCH OFFSET TO FAX CONTROL BLOCK
         MVC   PPGVALUE,PPGLCXDA   SET ADDRESS OF NEXT DISPLAY COMMAND
         BAS   R8,PPGCONV          COMPUTE ADDRESS OF NEXT DISPLAY CMD
         SPACE
         BAS   R8,PPGSKMAN         DISPLAY FAX ADDRESS
         SPACE
PPGLNDS7 BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG7            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
PPGNMSG7 CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGLNDS7            BRANCH IF NOT
         SPACE
         CLC   PPGCORE,PPGSTORG    TEST IF DESIRED ADDRESS
         BNE   PPGLNDS7            BRANCH IF NOT
         SPACE
         L     R1,PPGVHOLD         FETCH ADDRESS OF FAX
         N     R1,PCPØØØF          REMEMBER ONLY THE LOW-ORDER VALUE
         SRL   R1,2                DIVIDE IT BY FOUR
         MH    R1,PPGH9            COMPUTE OFFSET IN HWX TO PSI'S ADDR
         LA    R1,PPGRESP(R1)      POINT TO ADDRESS OF PSI
         MVC   PPGVALUE,2(R1)      STOW CORE ADDR FOR NEXT DISPLAY CMD
         SPACE
         LH    RØ,PPGX2C           FETCH OFFSET TO UTS
         BAS   R8,PPGCONV          COMPUTE ADDRESS FOR NEXT DISPLAY CMD
         SPACE
         BAS   R1,PPGSKIP          DISCARD REMAINING MESSAGES
         EJECT
************************************************************************
*                                                                      *
*        OBTAIN DATA IN USAGE TIER STATUS BLOCK                        *
*                                                                      *
************************************************************************
         SPACE
         BAS   R8,PPGSKMAN         DISPLAY FAX ADDRESS
         SPACE
PPGLNDS8 BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG8            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
PPGNMSG8 CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGLNDS8            BRANCH IF NOT
         SPACE
         CLC   PPGCORE,PPGSTORG    TEST IF DESIRED ADDRESS
         BNE   PPGLNDS8            BRANCH IF NOT
         SPACE
         L     R1,PPGVHOLD         FETCH ADDRESS OF FAX
         N     R1,PCPØØØF          REMEMBER ONLY THE LOW-ORDER VALUE
         SRL   R1,2                DIVIDE IT BY FOUR
         MH    R1,PPGH9            COMPUTE OFFSET IN HWX TO PSI'S ADDR
         LA    R1,PPGRESP(R1)      POINT TO ADDRESS OF UTS
         MVC   PPGVALUE,2(R1)      STOW CORE ADDR FOR NEXT DISPLAY CMD
         SPACE
         SR    RØ,RØ               CLEAR OFFSET TO DATA
         BAS   R8,PPGCONV          COMPUTE ADDRESS FOR NEXT DISPLAY CMD
         SPACE
         BAS   R1,PPGSKIP          DISCARD REMAINING MESSAGES
         SPACE
         BAS   R8,PPGSKMAN         DISPLAY UTS DATA
         SPACE
```

```
PPGLNDS9 BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG9            BRANCH IF NOT
         BAS   R2,PPGETLOG         COPY DATA TO OUTPUT AREA
         SPACE
PPGNMSG9 CLC   PPGISTMG,PPG245I    TEST IF DESIRED DATA
         BNE   PPGLNDS9            BRANCH IF NOT
         SPACE
         L     R1,PPGVHOLD         FETCH ADDRESS OF FAX
         N     R1,PCPØØØF          REMEMBER ONLY THE LOW-ORDER VALUE
         SRL   R1,2                DIVIDE IT BY FOUR
         MH    R1,PPGH9            COMPUTE OFFSET IN HWX TO PSI'S ADDR
         LA    R1,PPGRESP(R1)      POINT TO ADDRESS OF UTS
         PACK  PPGLCUTS,2(3,R1)    STOW UTS DATA
         NC    PPGLCUTS(1),PPGREMRS REMOVE RESERVED BITS
         SPACE
         BAS   R1,PPGSKIP          DISCARD REMAINING MESSAGES
         EJECT
***********************************************************************
*                                                                     *
*        FORMAT PSI DATA                                              *
*                                                                     *
***********************************************************************
         SPACE
         LA    R15,PPGLASTW        POINT TO BINARY DATA
         USING PPGPSI,R15          ESTABLISH PSI ADDRESSABILITY
         LA    R14,PPHVRBLK         POINT TO OUTPUT AREA
         USING PCPGEN,R14          ESTABLISH PCPGEN ADDRESSABILITY
         MVC   PCPMCH45,PPGHWMT    COPY MACHINE TYPE TO OUTPUT AREA
         MVI   PCPSLAS4,C'/'       USE SLASH AS DELIMITER
         MVC   PCPMDL45,PPGHMMN    COPY MODEL NUMBER TO OUTPUT
         SPACE
         CLI   PPGSVL,76           TEST IF CONN SUBSYSTEM IS ACTIVE
         BE    PPGWLAST            BRANCH IF NOT
         SPACE
         MVC   PCPMCH46,PPGCMTY    COPY CONN SUBSYS MACHINE TYPE
         MVI   PCPSLAS6,C'/'       STOW A SLASH
         MVC   PCPMDL46,PPGCMNO    COPY CONNECTIVITY SUBSYSTEM MODEL #
PPGWLAST MVC   PCPCOMMN,PCPCMNME   COPY COMMON NAME TO OUTPUT AREA
         MVI   PCPV,C'V'           USE V TO INDICATE VERSION
         MVC   PCPVERFP,PPGCMVRN   COPY COMMON VERSION
         MVI   PCPR,C'R'           USE R TO INDICATE RELEASE
         MVC   PCPRELFP,PPGCMREL   COPY COMMON RELEASE
         MVI   PCPM,C'M'           USE M TO INDICATE MODIFICATION LEVEL
         MVC   PCPMODFP,PPGCMMOD   COPY COMMON MODIFICATION LEVEL
         SPACE
         MVC   PCPCMRMG,PPGHMB+2   COPY STORAGE CAPACITY TO OUTPUT AREA
         SPACE
         MVC   PCPCUBOX,PPHCCUA    ASSUME IT'S THE 'A' BOX
         TM    PPGLCUTS,1          TEST IF THAT'S A VALID ASSUMPTION
         BO    PPGKLEAR            BRANCH IF 'TIS
         MVI   PCPCUBOX+4,C'B'      ELSE MAKE IT THE 'B' BOX
         SPACE
PPGKLEAR NC    PPGLCUTS(1),PPGREMUT REMOVE PREVIOUS INDICATORS
         LA    R8,PPGBOXES         POINT TO CONFIGURATION MODES
         LA    R1,PPGBOXCT         SET NUMBER OF THEM
PPGGMODE CLC   PPGLCUTS(1),Ø(R8)   TEST IF THIS IS THE PROPER MODE
         BE    PPGOTMOD            BRANCH IF IT'S THE CORRECT ONE
         LA    R8,PPGBOXLN(R8)     POINT TO THE NEXT ENTRY
         BCT   R1,PPGGMODE         TRY TRY AGAIN
         B     PPGNOMOD            ENOUGH IS ENOUGH
         SPACE
PPGOTMOD L     R1,4(R8)            POINT TO VERBAGE FOR MODE
         MVC   PCPCONFG,Ø(R1)       THEN COPY IT TO THE OUTPUT AREA
         EJECT
************************************************************************
*                                                                      *
*        CONVERT NCP'S VERSION OF A JULIAN DATE IN PACKED              *
*        DECIMAL TO ITS CORRESPONDING GREGORIAN DATE                   *
*                                                                      *
************************************************************************
         SPACE
PPGNOMOD ICM   R8,15,PPGDTSFK      FETCH LINK-EDIT DATE
         ICM   R8,8,PCPØ1          MANIPULATE HI-ORDER BYTE FOR CONVERT
         SPACE 1
         MVC   PCPLDATE+2(8),PATWORK EDIT PATTERN TO OUTPUT AREA
         ST    R8,PPGCLAMW+4       DATE TO PPGCLAMW
         LA    R8,PPGMONTB-3       PREPARE TO SCAN CONVERSION TABLE
         TM    PPGCLAMW+5,1        TEST FOR POSSIBILITY OF LEAP YEAR
         BO    PPGEDIT             BRANCH IF IMPOSSIBLE
         TM    PPGCLAMW+5,X'12'    TEST AGAIN FOR CERTAINTY
         BM    PPGEDIT             BRANCH IF NOT
         LA    R8,PPGLEAP-3        PREPARE TO SCAN LEAP YEAR CONV TABLE
PPGEDIT  ED    PCPLDATE+7(3),PPGCLAMW+5 SET UP YEAR
         MVC   PCPLDATE+6(2),PPG2Ø ASSUME THAT THE MILLENNIUM IS HERE
         CLI   PPGCLAMW+4,Ø        TEST ACCURACY OF THAT ASSUMPTION
         BNE   PPG21TH             BRANCH IF NAIL WAS HIT ON ITS HEAD
         MVC   PCPLDATE+6(2),PPG19   ELSE IT'S STILL THE 2ØTH CENTURY
PPG21TH  MVI   PCPLDATE+5,C'/'     DATE DELIMITER
         XC    PPGCLAMW(6),PPGCLAMW CLEAR ALL BUT JULIAN DATE
         SR    RØ,RØ               ZERO REGISTER Ø
         L     R1,PPGCLAMW+4       FETCH JULIAN DATA
         SLL   R1,4                SHIFT FOR FOXTROT
         O     R1,=XL4'ØØØØØØØF'   MAKE DATA TRULY PACKED DECIMAL
         ST    R1,PPGCLAMW+4       STOW IT
         CVB   R1,PPGCLAMW         CONVERT JULIAN DATE TO BINARY
PPGDATE  SR    R1,RØ               CONVERT FROM JULIAN DATE TO
         LA    R8,3(R8)            MONTH AND YEAR
         IC    RØ,Ø(R8)            FETCH DAYS IN A MONTH
         CR    RØ,R1               TEST IF INCOMPLETE MONTH
         BL    PPGDATE             BRANCH IF NOT
         EJECT
         CVD   R1,PPGCLAMW         CONVERT TO DECIMAL.
         MVO   PPGCLAMW(2),PPGCLAMW+6(2) SHIFT FOR EDIT
         ED    PCPLDATE+2(3),PPGCLAMW SETUP DAY
         MVI   PCPLDATE+2,C'/'     DATE DELIMITER
         MVC   PCPLDATE(2),1(R8)   MONTH
         SPACE
         MVC   PCPLYNK,PPGLINKC    INDICATE WHAT FOLLOWS THIS CONSTANT
         UNPK  PCPLHOUR(3),PCPDTLE4(2) HOUR OF DAY TO OUTPUT AREA
         MVI   PCPLC,C':'          INSERT SEPARATOR CHARACTER
         UNPK  PCPLMIN(3),PCPDTLE5(2) MINUTE OF THAT HOUR TO OUTPT AREA
         MVI   PCPLMIN+2,C' '      OVERLAY DE TRASH
         SPACE
         BAS   R2,PPGSCRIB         TRANSCRIBE DATA
         SPACE
         CLC   PPGLOVE,PPGHOLDØ    TEST IF PPGRDFEP WAS CALLED
         BE    PPGCLEAV            BRANCH IF SO
         B     PPCREAD              ELSE OBTAIN NAME OF NEXT NCPLOAD
         SPACE
PPGEOF   CLOSE (PPHODCB,,PPHIDCB)  CLEAN-UP
         SPACE
PPGCLEAV CLOSE (PPGACB)            CLEAN-UP
PPGLEAVE L     R1,PPGPARM5         POINT TO PPGSW
         MVC   Ø(1,R1),PPGSW       MAKE A COPY OF PPGSW
         SPACE
         SR    R15,R15             SHOW SUCCESS
         L     R13,PPGSAVE+4       POINT TO PREVIOUS SAVE AREA
         LM    RØ,R12,2Ø(R13)      RESTORE PREVIOUS REGISTERS
         L     R14,12(R13)         POINT TO RETURN ADDRESS
         BSM   RØ,R14              RETURN TO CALLER IN HIS AMODE
         EJECT
************************************************************************
*                                                                      *
*        ISSUE A SEND COMMAND                                          *
*                                                                      *
************************************************************************
         SPACE
PPGSKMAN SENDCMD RPL=PPGRPL,AREA=PPGSEND,RECLEN=PPGCMDLN,OPTCD=(SYN)
         LR    R3,R15              PRESERVE RETURN CODE FROM OPEN
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG6            BRANCH IF NOT
         SPACE
         MVC   PPHVRBLK(23),=CL23'RPLRTNCD=GL  RPLFDB2=LM '
         UNPK  PPHVRBLK+9(3),RPLRTNCD(2) ALTER RADIX OF FDB2
         MVI   PPHVRBLK+9+2,C' '    REMOVE DE DETRITUS
         TR    PPHVRBLK+9(2),PPGTRANS-24Ø MAKE RETURN CODE PRETTY
         UNPK  PPHVRBLK+21(3),RPLFDB2(2) ALTER RADIX OF RTNCD
         MVI   PPHVRBLK+21+2,C' '   REMOVE DE DETRITUS
         TR    PPHVRBLK+21(2),PPGTRANS-24Ø MAKE RETURN CODE PRETTY
         BAS   R2,PPGSCRIB          TRANSCRIBE DATA
PPGNMSG6 LTR   R15,R3               TEST RETURN CODE
         BE    Ø(R8)                RETURN TO CALLER
         OI    PPGSW,1              SET INHIBIT-PROCESSING SWITCH
         B     PPGCLEAV             EXIT
         EJECT
************************************************************************
*                                                                      *
*        ISSUE A RECEIVE COMMAND                                       *
*                                                                      *
************************************************************************
         SPACE
PPGRKMAN RCVCMD  RPL=PPGRPL,AREA=PPGRECV,AREALEN=13Ø,                  N
               OPTCD=(TRUNC,Q,SYN)
         LR    R3,R15               PRESERVE RETURN CODE FROM RECV
         SPACE
         TM    PPGSW,2              TEST IF MESSAGES ARE DESIRED
         BO    PCPNMSG7             BRANCH IF NOT
         SPACE
         MVC   PPHVRBLK(23),=CL23'RPLRTNCD=GL  RPLFDB2=LM '
         UNPK  PPHVRBLK+9(3),RPLRTNCD(2) ALTER RADIX OF FDB2
         MVI   PPHVRBLK+9+2,C' '    REMOVE DE DETRITUS
         TR    PPHVRBLK+9(2),PPGTRANS-24Ø MAKE RETURN CODE PRETTY
         UNPK  PPHVRBLK+21(3),RPLFDB2(2) ALTER RADIX OF RTNCD
         MVI   PPHVRBLK+21+2,C' '   REMOVE DE DETRITUS
         TR    PPHVRBLK+21(2),PPGTRANS-24Ø MAKE RETURN CODE PRETTY
         BAS   R2,PPGSCRIB          TRANSCRIBE DATA
PCPNMSG7 LTR   R15,R3               TEST RETURN CODE
         BE    Ø(R8)                RETURN IF RECEIVE WAS SUCCESSFUL
         OI    PPGSW,1              SET INHIBIT-PROCESSING SWITCH
         B     PPGCLEAV             EXIT
         EJECT
************************************************************************
*                                                                      *
*        CONVERT A HEXADECIMAL VALUE IN EBCDIC TO ONE IN BINARY        *
*                                                                      *
************************************************************************
         SPACE
PPGCONV  TR    PPGVALUE,PPGHEX2B    REMOVE ZONES FROM OFFSET
         PACK  PPGVHOLD,PPGVALUE(7) COMPRESS OFFSET
         L     R1,PPGVHOLD          FETCH COMPRESSED VALUE
         SPACE
         AR    R1,RØ                ADD OFFSET TO ADR OF BLK REQUESTED
         ST    R1,PPGVHOLD          STOW IT
         UNPK  PPGDOUBL(9),PPGVHOLD(5) SEPARATE VALUE WITH FOXTROT
         TR    PPGDOUBL(8),PPGBIN2X CONVERT VALUE TO EBCDIC
         MVC   PPGSTORG,PPGDOUBL+2 SET ADDRESS FOR NEXT DISPLAY COMMAND
         BR    R8                   RETURN TO CALLER
         EJECT
************************************************************************
*                                                                      *
*        TRANSCRIBE VTAM'S RESPONSE                                    *
*                                                                      *
************************************************************************
         SPACE
PPGETLOG L     R1,RPLRLEN          FETCH LENGTH OF RECORD
         C     R1,PPGF6            TEST FOR MINIMUM LENGTH RECORD
         BL    Ø(R2)               RETURN IF NOT PRESENT
         S     R1,PPGF5            REDUCE COUNT FOR MOVE
         EX    R1,PPGCOPY          COPY DATA TO OUTPUT AREA
         UNPK  PPHVRBLK(9),PPGRECV(5) CONVERT HEADER TO EXTERNAL DECIML
         TR    PPHVRBLK(8),PPGTRANS-24Ø MAKE HEXADECIMAL PRETTY
         MVI   PPHVRBLK+8,C' '      DELETE DE TRASH
         SPACE
PPGSCRIB L     R1,PPGPARM3         POINT TO OUTPUT DCB
         L     RØ,PPGPARM4         POINT TO OUTPUT AREA
         PUT   (1),(Ø)             TRANSCRIBE DATA
         MVI   PPHCC,C' '          SINGLE SPACE DATA LINES
         MVC   PPHVRBLK(PPHDLEN-1),PPHCC REFRESH OUTPUT AREA
         BR    R2                  RETURN TO CALLER
         SPACE 3
************************************************************************
*                                                                      *
*        RECEIVE THEN DISCARD VTAM'S REMAINING RESPONSES               *
*                                                                      *
************************************************************************
         SPACE
PPGSKIP  ST    R1,PPGSKIPR         STOW RETURN ADDRESS
         SPACE
PPGSKIPØ BAS   R8,PPGRKMAN         OBTAIN RESPONSE TO DISPLAY COMMAND
         SPACE
         TM    PPGSW,2             TEST IF MESSAGES ARE DESIRED
         BO    PPGNMSG             BRANCH IF NOT
         SPACE
         MVC   PPHVRBLK+1Ø(13Ø),PPGISTMG COPY DATA TO OUTPUT BUFFER
         UNPK  PPHVRBLK(9),PPGRECV(5)
         TR    PPHVRBLK(8),PPGTRANS-24Ø
         MVI   PPHVRBLK+8,C' '
         BAS   R2,PPGSCRIB
         SPACE
PPGNMSG  CLC   PPGISTMG,PPG241I    TEST IF COMPLETED COMMAND MESSAGE
         BNE   PPGSKIPØ            BRANCH IF NOT
         SPACE
         L     R1,PPGSKIPR         RETRIEVE RETURN ADDRESS
         BR    R1                  RETURN TO CALLER
         EJECT
************************************************************************
*                                                                      *
*        CONSTANTS, EQUATES, AND OTHER SUCH NONSENSE                   *
*                                                                      *
***********************************************************************
         SPACE
PPGDOUBL DC    D'Ø'
PAT1MEG  DC    A(1Ø24*1Ø24)
PPGSAVE  DS    18F
PPGSKIPR DS    F
PPGHOLDØ DS    F
PPGLOVE  DC    CL4'LOVE'
PPGSW    DC    X'Ø2'
PPGCØØ4  DC    CL3'ØØ4'
PPGCØØ38 DC    CL6'ØØØØ38'
PPGLCXDA DS    CL6
PPGLASTA DS    F
PPGF5    DC    F'5'
PPGF6    DC    F'6'
PCPØØØF  DC    XL4'ØØØØØØØF'
PPGH9    DC    H'9'
PPGX5C   DC    H'92'               X'5C' ==> OFFSET TO HWX AND PSI
PPGX78   DC    H'12Ø'              X'78' ==> OFFSET TO FAX
PPGX2C   DC    H'44'               X'2C' ==> OFFSET TO UTS
PPGX6Ø   DC    H'96'               X'6Ø' ==> OFFSET TO ADR OF LAST BYTE
PPGHMB   DC    CL6' '
PPGLCUTS DS    CL2
PPGREMRS DC    X'73'
PPGREMUT DC    X'FC'
         SPACE
PPGRPL   RPL   AM=VTAM,ACB=PPGACB,AREA=PPGRECV,AREALEN=13Ø,OPTCD=(SYN)
         SPACE
PPGACB   ACB   AM=VTAM,MACRF=NLOGON,PASSWD=PPGPSWD
         SPACE
PPHODCB  DCB LRECL=137,BLKSIZE=137,DSORG=PS,MACRF=PM,RECFM=FA,         C
               DDNAME=SYSPRINT,DCBE=PPHDCBPM
         SPACE 1
PPHDCBPM DCBE
         SPACE
PPHIDCB  DCB LRECL=8Ø,BLKSIZE=8Ø,DSORG=PS,MACRF=GM,RECFM=F,            C
               DDNAME=SYSIN,DCBE=PPHDCBGL,EODAD=PPGEOF
         SPACE 1
PPHDCBGL DCBE
         SPACE
PPGCOPY  MVC   PPHVRBLK+1Ø(*-*),PPGISTMG ===>  EXEC ONLY  <===
PPGETAID MVC   PPGPSWD+1(*-*),2(R1)      ===>  EXEC ONLY  <===
         EJECT
         YREGS
         SPACE
PPGH1    DC    H'1'
PPGPATCU DC    XL4'4Ø2Ø212Ø',C'MB'
         SPACE
         DS    ØF
PPGBOXES DC    X'1Ø',A(PPHBOX1)
PPGBOXLN EQU   *-PPGBOXES
         DC    X'2Ø',A(PPHBOX2)
         DC    X'4Ø',A(PPHBOX3)
         DC    X'5Ø',A(PPHBOX4)
PPGBOXCT EQU   ((*-PPGBOXES)/PPGBOXLN)
         SPACE
PPHBOX1  DC    CL12'SINGLE CCU'
PPHBOX2  DC    CL12'TWIN-STANDBY'
PPHBOX3  DC    CL12'TWIN-BACKUP'
PPHBOX4  DC    CL12'TWIN-DUAL'
PPHCCUA  DC    CL5'CCU A'
         EJECT
***********************************************************************
*                                                                     *
*        FOLLOWING IS THE COMMAND THAT IS ISSUED FOR VTAM TO PROCESS  *
*                                                                     *
***********************************************************************
         SPACE
         DS    ØF
PPGSEND  DC    X'ØØ'                HEADER
         DC    X'Ø3'                STATUS FIELD ( RETURN REPLY )
         DC    XL2'ØØØ1'            IDENTIFICATION NUMBER
         DC    C'D NET,NCPSTOR,LENGTH='
PPGSIZE  DC    CL3'ØØ4'
         DC    C',ADDR='
PPGSTORG DC    CL6'ØØØØ38'          WORD DIRECT ADDRESSABLE STORAGE(XDA)
         DC    C',ID='
PPGNCPID DC    CL8'NCPE4'
PPGCMDLN EQU   *-PPGSEND
         SPACE
PPGLINKC DC    CL11'LINK-EDIT:'
PPGPSWD  DC    AL1(4),CL8'CLAM'
PPGTRANS DC    C'Ø123456789ABCDEF'
PPG241I  DC    CL8'IST241I'
PPG245I  DC    CL8'IST245I'
PPGVALUE DC    CL6' '
         DS    ØF
PPGVHOLD DC    CL5' '
PPGPSIZE DC    CL3'1ØØ'
         SPACE
         DS    ØF
PPGHEX2B EQU   *-X'C1'              EBCDIC HEX TO BINARY TRANSLATION TBL
         DC    X'ØAØBØCØDØEØF'
         DC    XL41'ØØ'
         DC    X'ØØØ1Ø2Ø3Ø4Ø5Ø6Ø7Ø8Ø9'
         SPACE
PPGBIN2X EQU   *-C'Ø'
         DC    C'Ø123456789ABCDEF'
         SPACE
         DS    ØF
PPGRECV  DC    CL13Ø' '
PPGLASTW DC    CL13Ø' '
         SPACE
PATWORK  DC    X'FØ212Ø4Ø4ØFØ212Ø'
         EJECT
PPGMONTB DC    AL1(31),C'Ø1'
         DC    AL1(28),C'Ø2'
         DC    AL1(31),C'Ø3'
         DC    AL1(3Ø),C'Ø4'
         DC    AL1(31),C'Ø5'
         DC    AL1(3Ø),C'Ø6'
         DC    AL1(31),C'Ø7'
         DC    AL1(31),C'Ø8'
         DC    AL1(3Ø),C'Ø9'
         DC    AL1(31),C'1Ø'
         DC    AL1(3Ø),C'11'
         DC    AL1(31),C'12'
         SPACE 1
PPGLEAP  DC    AL1(31),C'Ø1'
         DC    AL1(29),C'Ø2'
         DC    AL1(31),C'Ø3'
         DC    AL1(3Ø),C'Ø4'
         DC    AL1(31),C'Ø5'
         DC    AL1(3Ø),C'Ø6'
         DC    AL1(31),C'Ø7'
         DC    AL1(31),C'Ø8'
         DC    AL1(3Ø),C'Ø9'
         DC    AL1(31),C'1Ø'
         DC    AL1(3Ø),C'11'
         DC    AL1(31),C'12'
PPGCLAMW DC    2D'Ø'
PPG19    DC    CL2'19'
PPG2Ø    DC    CL2'2Ø'
PCPØ1    DC    X'Ø1'
PCPODATA DC    CL137' '
PPGPARM1 DC    A(PPGNCPID)
PPGPARM2 DC    A(PPGPSWD)
PPGPARM3 DC    A(PPHODCB)
PPGPARM4 DC    A(PCPODATA)
PPGPARM5 DC    A(PPGSW+X'8ØØØØØØØ')
         EJECT
*        TITLE USED FOR MAPPING VIRTUAL ROUTES
         SPACE
PPGDSECT DSECT
PPHCC    DC    C'1'
PPHVRBLK EQU   *
PPHSUBA# DC    CL4' '
         DC    C' '
```

```
PPHDEST# DC    CL4' '
         DC    C' '
PPHVR#   DC    CL4'VR #'
         DC    CL3' '
PPHAJSA# DC    CL4' '
         DC    C' '
PPHER#   DC    CL4' '
         DC    C' '
PPHTP1   DC    CL4'TP #'
         DC    C' '
PPHRT1   DC    CL3'PRI'
         DC    C' '
PPHSTAT1 DC    CL8'VR STATE'
         DC    C' '
PPHFCFS1 DC    CL4'STAT'
         DC    C' '
PPH#LUS1 DC    CL8'SESS-CNT'
         DC    CL2' '
PPHTP2   DC    CL4'TP #'
         DC    C' '
PPHRT2   DC    CL3'PRI'
         DC    C' '
PPHSTAT2 DC    CL8'VR STATE'
         DC    C' '
PPHFCFS2 DC    CL4'STAT'
         DC    C' '
PPH#LUS2 DC    CL8'SESS-CNT'
         DC    CL2' '
PPHTP3   DC    CL4'TP #'
         DC    C' '
PPHRT3   DC    CL3'PRI'
         DC    C' '
PPHSTAT3 DC    CL8'VR STATE'
         DC    C' '
PPHFCFS3 DC    CL4'STAT'
         DC    C' '
PPH#LUS3 DC    CL8'SESS-CNT'
         DC    CL12' '
PPHDLEN  EQU   *-PPHCC
         SPACE 2
PPGPRSTB DC    256X'ØØ'
         ORG   PPGPRSTB+C'.'
         DC    C'.'
         ORG
         EJECT
***********************************************************************
*                                                                     *
*        GRID FOR FIELDS SELECTED FROM PRODUCT SET IDENTIFIER, ETC.   *
*                                                                     *
***********************************************************************
```

```
         SPACE
PCPGEN   DSECT
PCPMCH45 DC    CL4' '
PCPSLAS4 DC    C'/'
PCPMDL45 DC    CL3' '
         DC    C' '
PCPMCH46 DC    CL4' '
PCPSLAS6 DC    C'/'
PCPMDL46 DC    CL3' '
         DC    CL2' '
PCPCONFG DC    CL12' '
PCPCUBOX DC    CL5' '
         DC    CL2' '
PCPCMRMG DC    CL4' '
         DC    CL2' '
PCPCOMMN DC    CL7' '
         DC    C' '
PCPV     DC    C'V'
PCPVERFP DC    CL2' '
PCPR     DC    C'R'
PCPRELFP DC    CL2' '
PCPM     DC    C'M'
PCPMODFP DC    CL2' '
         DC    CL2' '
PCPLYNK  DC    CL11'LINK-EDIT:'
PCPLHOUR DC    CL2' '
PCPLC    DC    C':'
PCPLMIN  DC    CL2' '
         DC    C' '
PCPLDATE DC    CL1Ø' '
         EJECT
************************************************************************
*                                                                      *
*        GRID FOR DATA RECEIVED FROM VTAM                              *
*                                                                      *
************************************************************************
         SPACE
PPGRECVD DSECT
         DC    F'Ø'
PPGISTMG DC    CL8'IST829I'
         DC    C' '
PPGCORE  DC    CL6'1ØØ147'
         DC    CL3' '
PPGRESP  DC    CL8' '
         DC    C' '
         DC    CL8' '
         DC    C' '
         DC    CL2' '
PPGXDA   DC    CL6' '              INVARIANT LOCATION OF XDA IN LOW KOR
         DC    C' '
```

```
PPGLAST  DC    CL8' '
         EJECT
************************************************************************
*                                                                      *
*        GRID FOR SELECTED FIELDS WITHIN THE PRODUCT SET IDENTIFIER   *
*                                                                      *
************************************************************************
         SPACE
PPGPSI   DSECT
PPGSVL   DC    X'4C'               76 BEFORE 9ØØ ACTIVE OR 98 AFTERWARD
PPGSVK   DC    X'1Ø'               SUBVECTOR KEY
         DC    H'Ø'                RESERVED, LENGTH OF PRODUCT ID
         DC    F'Ø'                PRODUCT ID KEY(11), CLASS(1), LEN, Ø
PPGHWSFF DC    X'11'               HARDWARE SUBFIELD
PPGHWMT  DC    CL4'3745'           MACHINE TYPE IN EBCDIC
PPGHMMN  DC    CL3'61A'            MACHINE MODEL NUMBER IN EBCDIC
         DC    CL2' '              PLANT THAT MAUFACTORED THIS BOX
         DC    CL7' '              SERIAL NUMBER OF THIS BOX
         DC    XL3'ØØ'             LENGTH OF PROD ID, ID(11), CLASS(4)
         DC    XL2'ØØ'             LENGTH OF COMPONENT ID, KEY(Ø2)
         DC    CL9' '              COMPONENT ID IN EBCDIC
         DC    CL3' '              COMPONENT RELEASE IN EBCDIC
         DC    XL2'ØØ'             LEN OF COMMON LEVEL SUBFIELD, KEY(4)
PPGCMVRN DC    CL2' '              COMMON VERSION IN EBCDIC
PPGCMREL DC    CL2' '              COMMON RELEASE IN EBCDIC
PPGCMMOD DC    CL2' '              COMMON MOD LVL IN EBCDIC
         DC    X'ØØ'               LEN OF LINK-EDIT DATE & TIME
PPGDTSFK DC    X'Ø9'               KEY(9)
         DC    XL3'ØØ'             LINK-EDIT DATE
PCPDTLE4 DC    X'ØØ'               TIME(HOUR) IN PACKED DEC
PCPDTLE5 DC    X'ØØ'               TIME(MIN) IN PACKED DEC
         DC    XL2'ØØ'             LEN OF COMMON NAME, KEY(6)
PCPCMNME DC    CL7' '              COMMON NAME
         DC    XL2'ØØ'             LEN OF CUSTOMIZATION, KEY(7)
         DC    CL8' '              CUSTOMIZATION ID
         EJECT
*        THE FOLLOWING FIELDS ARE AVAILABLE ONLY AFTER THE 3746
*        MODEL 9ØØ FRAME CONNECTIVITY SUBSYSTEM HAS BEEN ACTIVATED
         SPACE
         DC    XL4'ØØ'             LEN OF PRID ID, KEY(11), CLAS(1), LN
         DC    XL2'ØØ'             HARDWARE SUBFIELD KEY(Ø), FORMAT(11)
PPGCMTY  DC    CL4' '              3746/9ØØ MACHINE TYPE
PPGCMNO  DC    CL3' '              3746/9ØØ MODEL NUMBER
         DC    CL2' '              PLANT ID
         DC    CL7' '              MACHINE SERIAL NUMBER
PCPSNI   DSECT
PCPCC    DC    C'1'
PCPNET   DC    CL8' '
         DC    C' '
PCPRRN   DC    CL8' '
```

```
         DC    C' '
PCPREL   DC    CL3' '
PCPMOD   DC    CL3' '
         DC    CL3' '
PCPUNIT  DC    CL4' '
PCPDLEN  EQU   *-PCPRRN
         ORG   PCPCC+5
PCPSNIE  DC    CL8' '
         DC    CL2' '
PCPSNISA DC    CL4' '
         ORG   PCPNET
PCPADNET DC    CL8' '
         DC    CL3' '
PCPACDRM DC    CL8' '
         DC    C' '
PCPENTRY DC    CL8' '
         DC    C' '
         ORG
         TITLE 'GENERATE OS/39Ø CONTROL BLOCKS'
         IFGRPL
         SPACE 1
         IFGACB
         SPACE 1
         END
```

Systems Programmer (USA) © Reserved 2002

Transferring files from OS/390 to the PC

This article present a small program for transferring files from OS/390 to the PC. We use IBM as the ftp server, the PC as the ftp client, and command mget; if you want to transfer files to OS/390, you should use command mput. We also use ftp with the -i option to turn off interactive prompting during multiple file transfer.

The file FTCOMM.TXT (used in the ftp -s option) contains the following commands:

1 open. <ipadress of your ftp server> (with username and password).

2 cd. Set the working directory to a dataset on the OS/390.

3 mget. Multiple file transfer from the working directory to the ftp client.

4 bye. Close ftp command.

Once IBMLOAD.EXE is started, you'll get the following messages:

```
Load files from IBM

Username>pippg31
Password>****
IBM data set>pippg31.user.pli
File(s) |*|>d*

Loading into c:\IBMLOAD\Files ...
(38 files loaded)

press any key
```

The OUT.TXT is the log file of our ftp command.

```
ftp> Connected to 1.2.3.4.
open 1.2.3.4
22Ø-FTPD1 IBM FTP CS/39Ø V2R5 at PSJES229.posted.co.yu, 13:ØØ:56 on
2ØØ2-Ø2-25.
22Ø Connection will close if idle for more than 5 minutes.
User (1.2.3.4:(none)):
331 Send password please.

23Ø PIPPG31 is logged on.  Working directory is "PIPPG31.".
ftp> ftp>
cd ..
```

```
250 "" is the working directory name prefix.
ftp> cd pippg31.user.pli
250 "PIPPG31.USER.PLI" partitioned data set is working directory
ftp> mget d*
200 Representation type is Ascii NonPrint
200 Port request OK.
125 Sending data set PIPPG31.USER.PLI(DB00C) FIXrecfm 80
250 Transfer completed successfully.
ftp: 1023 bytes received in 0.00Seconds 1023000.00Kbytes/sec.
...

200 Port request OK.
125 Sending data set PIPPG31.USER.PLI(D9) FIXrecfm 80
250 Transfer completed successfully.
ftp: 1066 bytes received in 0.01Seconds 106.60Kbytes/sec.
ftp> bye
221 Quit command received. Goodbye.
```

IBMLOAD.CPP

```
//
// Transfer files from IBM/OS390.
//
// You have to write IP Address of your FTP server.
#define IPADDRESS "1.2.3.4"
#include <conio.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <direct.h>
int Load();
int Input(char *,int,int,char *);
int nFiles(int *);

int main(int argc, char* argv||)
{
      Load();
      printf("\npress any key\n");
      char c=getch();
      return 0;
}

int Load()
{
      FILE *f1;
      char temp1|256|;
      printf("Load files from IBM\n");
      char user|9|,pass|9|,files|9|;
      char cat|256|;
      int num;
      if(Input("Username",0,8,user)) return 1;
      if(Input("Password",1,8,pass)) return 1;
      if(Input("IBM data set",0,255,cat)) return 1;
      if(Input("File(s) |*|",0,8,files)) return 1;
      if(strlen(files)==0) strcpy(files,"*");
      if((f1=fopen("FTPCOMM.TXT","w"))==NULL)
      {
            printf("\nFile FTPCOMM.TXT open ERROR.");
            return 1;
      }
      fprintf(f1,"open %s\n",IPADDRESS);
      fprintf(f1,"%s\n",user);
      fprintf(f1,"%s\n",pass);
      fprintf(f1,"cd ..\n");
      fprintf(f1,"cd %s\n",cat);
      fprintf(f1,"mget %s\n",files);
      fprintf(f1,"bye\n");
      fclose(f1);
      if(chdir("Files"))
      {
            if(mkdir("Files"))
            {
                  getcwd(temp1,256);
                  printf("\nCreate directory %s\\Files ERROR.",temp1);
                  return 1;
             }
            if(chdir("Files"))
            {
                  getcwd(temp1,256);
                  printf("\nChange directory %s\\Files ERROR.",temp1);
                  return 1;
             }
      }
      getcwd(temp1,256);
      printf("\n\nLoading into %s ...\n",temp1);
      system("ftp -i -s:..\\FTPCOMM.TXT >..\\OUT.TXT");
      chdir("..");
      remove("FTPCOMM.TXT");
      if(nFiles(&num)) return 1;
      printf("(%d files loaded)\n",num);
      remove("OUT.TXT");
      return 0;
}

int Input(char * mess,int p, int len,char * res)
{
      char c;
      char s|256|;
      c=0;
      int i=0;
      int dow=1;
      printf("\n%s>",mess);
      while(dow && i<len)
      {
            c=getch();
            if(c==8) // backspace key
            {
                  if(i>Ø)
                  {
                        i--;
                        printf("%c %c",c,c);
                  }
            }
            else
            {
                  if(c|=13 && c|=27)
                  {
                              if(p) // display password char
                              printf("*");
                              else
                              printf("%c",c);
                        s|i|=c;
                        i++;
                  }
                  else dow=Ø;
            }
      }
      s|i|='\Ø';
      strcpy(res,s);
      if(c==27) return 1;
      return Ø;
}

int nFiles(int * num)
{
      char temp1|256|;
      FILE * f1;
      *num=Ø;
      if((f1=fopen("OUT.TXT","r"))==NULL)      return 1;
      while(|feof(f1))
      {
            fscanf(f1,"%s",temp1);
            // Search for "Transfer completed successfully."
            // The word "Transfer" is enough.
            if(strcmp(temp1,"Transfer")==Ø)      (*num)++;
      } // while
      fclose(f1);
      return Ø;
}
```

Dejan Jelic
(Yugoslavia)

Now that IBM has withdrawn the 3745 and 3746

On 26 February 2002, IBM announced that the 3745 Communications Controller and the 3746 Model 900 and 950 Nways Multiprotocol Controllers would be withdrawn from marketing as of the end of September 2002. The final orders for these FEPs – once the standard-bearers for SNA networking – will be accepted in November 2002, although IBM has already warned that even now it will only ship new 374*x*s on an 'as-available' basis.

Although the withdrawal of these machines – particularly the 3745, which is now nearly 13 years old – shouldn't come as a surprise, it's guaranteed to catch many a diehard SNA stalwart off-guard. There are still tens of thousands of 374*x*s in mission-critical use, especially in financial and banking environments, but demand for these network controllers has been in steady decline since the late 1990s. This is for two key reasons:

- The first is the on-going transition from SNA-centric enterprise networking towards IP-based networking.
- The second is the incessant promotion by IBM and its post-1999 networking partner Cisco of LAN-oriented, 'direct-connect', network-to-mainframe options such as OSA-2, OSA-Express, Cisco's Channel Interface Processor (CIP), and Cisco Channel Port Adapter (CPA).

Although this is an unequivocal 'heads up' from IBM as to the changing fortunes of SNA networking, this withdrawal is not too disturbing in itself – at least for the time being. For a start, standard IBM maintenance will be available for these legendarily reliable boxes for at least another five years. After that, it will be on a 'time-and-parts' basis, although you will then start running into the danger of parts unavailability, coupled with the potential of increased failures as the hardware technology starts to exceed 20 years of age. Besides, there's a glut of bargain-basement-priced, pre-owned 374*x*s on the market today – check the Web or even talk to IBM Global Services.

So the bottom line here is that over the next five years there will not be a shortage of 374*x*s on the market, and you'll still be able to get reasonable service contracts for them. There's no doubt that 374*x*s will be this decade's equivalent of IBM 029 card punch units in the 1990s – although the need for any punched cards to sustain mainframe operations were long gone, you would still find 029s parked in dark corners of machine rooms. The same will be true for 374*x*s. However, by the time we head towards 2010, it will probably not be a good idea to have too much mission-critical traffic going in and out of mainframes via 374*x*s.

In *The Evolving Mainframe Networking Landscape*, released by IBM to supplement the 374*x* withdrawal announcement, IBM's key recommendations for migrating from SNA networking to IP revolve around:

- OSA-Express.
- IBM's Communications Server for the mainframe.
- IBM's Enterprise Extender, which is High Performance Routing (HPR, ie next generation APPN) over IP. (Cisco's equivalent is included within what it refers to as 'SNA Switching Services' (SNASw).)
- tn3270(E) Server, which is a part of Communications Server.
- Parallel sysplex.
- IBM Desktop Host Integration – for example, WebSphere Host On-Demand Java-based 3270 emulator and Host Publisher Version 3.5, which can represent 3270 in HTML, JavaBeans, or XML form.
- Application Web enablement using the Java-centric WebSphere Application Server (WAS). (BEA's WebLogic and iPlanet's Application Server are comparable offerings.)
- TCP/IP messaging via the WebSphere MQ family (formerly MQSeries).

IBM has also produced a 422-page Red Book called *IBM Communications Controller Migration Guide* that covers all of the items in the above list, and more, in great detail, but in a somewhat disjointed and choppy manner. It's nonetheless a good reference to have, and, given its extensive length, can cover more esoteric and marginal topics (eg Network Routing Facility (NRF) and Non-SNA Interconnection (NSI)) than can be addressed in this relatively short article.

LET'S NOT FORGET DLSW

The 'everything but the kitchen sink' list from IBM above, which covers all of its current offering irrespective of their exact relevance, does not, however, mention Data Link Switching (DLSw), the most widely-used SNA-IP integration technology, introduced by IBM in 1992. The reason is that IBM no longer markets any bridge/routers and DLSw is a bridge/router-based technology. Enterprise Extender (EE), which is available all the way into the mainframe via the various platform-specific Communications Server offerings, is now positioned by IBM as the strategic replacement for DLSw. However, this is not really the case, and reflects a bit of 'sour grapes' mentality from IBM. While EE is undoubtedly a powerful SNA-IP integration scheme, it can be overkill in many situations where DLSw (still readily available from Cisco on its bridge/routers) would be a much better solution.

EE is undoubtedly better than DLSw in two relatively specialized scenarios. One of these applies only if you have users who regularly switch between applications running on mainframes that are located far apart (eg North America and Europe, or the East and West coast of the USA). The other is if you really rely on SNA LU 6.2-based Class-of-Service (COS) traffic prioritization from the desktop to the mainframe. These two scenarios, where EE excels, can be described as follows:

1 EE's native support of APPN/HPR Network Node routing ensures optimal end-to-end paths for users who must regularly switch between applications running on different data centres. DLSw doesn't support NN or ACF/NCP-like routing and will first route SNA traffic to a pre-determined SNA 'routing gateway' (eg 3745 or ACF/VTAM). So with DLSw, traffic routing between dispersed data centres may occur in two steps: first to the SNA 'routing gateway' and from there to the target data centre if the application happens to be running at another data centre.

2 EE supports LU 6.2 COS traffic prioritization all the way from the

desktop to the mainframe. DLSw does not recognize LU 6.2 COS. This type of LU 6.2 end-to-end COS, though not widely relied upon, has been exploited in certain, typically OS/2-oriented, financial applications. If such prioritization is imperative, then DLSw is not an option.

DLSw will work in both of these cases, but may not deliver the optimum results, and could impact performance and response times. But DLSw and EE are both IP-based transport mechanisms. The move to IP is invariably accompanied by a healthy increase in overall network bandwidth, with 56Kbps links considered the bare minimum whereas 19.2Kbps was considered plenty for SNA links. This increase in network bandwidth will usually compensate for DLSw's non-optimum 'routing' in these two scenarios. If, however, you don't have one of these two requirements, you should start by evaluating DLSw (or DLSw+ as Cisco calls it) ahead of EE.

CAREFUL PLANNING IS THE TICKET

In the end, DLSw and EE will still only be interim solutions, in that they transport SNA traffic across an IP network. The long-term objective is to keep all of your SNA within the data centre and have an IP-only network. This is where some of the other items that appear in IBM's list – such as tn3270(E) server, Host On-Demand, Host Publisher, WebSphere Application Server, and MQSeries – come in.

tn3270(E) makes the move towards an IP-only network possible through a client/server mechanism whereby IP-based 3270 emulation clients (eg Host On-Demand, IBM's PComm, NetManage's RUMBA, etc) interact with a data-centre-resident tn3270(E) server (eg Communications Server, Microsoft SNA Server, or ICOM's Winpass Tnserver). tn3270(E) is a widely-used (over 12 million users today), industry-standard-based solution which permits the mix-and-match of servers and clients from multiple vendors.

Whereas tn3270(E) eliminates SNA from the network, Host Publisher, WAS, and MQSeries go even further by eliminating 3270 datastream from the network. With these solutions, both SNA and 3270 are confined to the data centre. Rather than sending 3270 datastream to the client, the 3270 is converted to HTML, XML, or a JavaBean that can be embedded inside a Web page that supports Java Server Pages (JSP) object technology. This type of approach removes the need for 3270-specific emulators on the client systems. Instead, SNA/3270 applications can be accessed directly from standard Web browsers. Although technology and proven solutions (from over 70 vendors) abound, moving from SNA networking to IP is not something that can, or should, be done in a hurry without much planning, because of the mission-critical nature of the applications and traffic involved.

The good news is that you still have a few years to achieve this move, in a slow, deliberate, and systematic manner, even if, bucking the trend, you haven't done anything over the last five years to slowly converge towards an IP-centric network. Depending on the exact nature of your current SNA network, the conversion from SNA to IP networking will involve some relatively easy tasks, most likely intermingled with more difficult and intractable ones. Figures 1, 2, and 3 categorize the tasks most often encountered into three types, based on the degree of difficulty, and show the most often used techniques to achieve the migration to IP. All of the suggested solutions assume the implementation of a bridge/router-based IP network to take over from today's SNA network. Note that the Figures aren't comprehensive. Rather, they provide a snapshot of the most commonly encountered migration tasks and show some of the popular solutions. IBM's Red Book goes into considerably more detail, and covers other scenarios.

IMPACT ON MAINFRAME CAPACITY

The term 'FEP' (Front End Processor) in the case of 3745s running ACF/NCP was very appropriate and descriptive. The 37*x*5s, starting with the 3705, were IBM's only true, long-term 'off load' processors for mainframes. ACF/NCP performs various address conversion, message chaining, Request/Response Header (RH) validation, and data retransmission functions that preclude ACF/VTAM on the mainframe from having to do them. Bypassing ACF/NCP means that these functions will now have to be done on the mainframe. A rule-of-thumb is that this is likely to increase mainframe CPU cycle consumption by around 20%. Though not dramatic, this could start pushing up against the upper limits of current capacity plans.

This 20% increase doesn’t include doing tn3270(E) serving – especially with Secure Sockets Layer (SSL)-based encryption/decryption – on the mainframe. tn serving will further increase CPU usage noticeably even if you use the numerous hardware cryptographic co-processors now available to ‘off-load’ some of the security-related processing.

The bottom line here is that getting rid of the 374*x*s will increase your mainframe workload. Since we’re talking four to six years down the road, the chances are that you’ll be planning to get a mainframe

Task 1 Eliminate 374*x*s being used as Remote Communications Controllers (RCPs).

Solution DLSw or EE with SDLC, BSC, ALC, and start-stop link consolidation using serial ports on the bridge/router. BSC, ALC, etc will require other IP-tunnelling schemes such as Cicso’s BSTUN.

Task 2 Move or replace SDLC or BSC attached 3*x*74 type controllers.

Solution 1. Use the SDLC or BSC capability described above with DLSw and BSTUN.

2. If 3*x*74s are being used as SNA gateways for LAN-attached PCs, get rid of the 3*x*74s and convert PC software from ‘full-stack’ SNA (eg IBM PComm, Attachmate EXTRA!) to tn3270(E) client. Note that most ‘fat client’ emulators capable of acting as ‘full-stack’ clients can also work as tn3270(E) clients.

Task 3 Replace remote SNA gateways such as Novell’s NetWare for SAA, Microsoft’s SNA Server, or IBM’s Communications Server.

Solution Rather than using a remote SNA gateway that is SNA-based, move to a centralized, data-centre-, or mainframe-resident tn3270(E) server, and use IP to transport ‘tn’ traffic to/from tn3270(E) clients. Note that most clients that worked with remote SNA gateways can be reconfigured to work as tn3270(E) clients.

Task 4 Start channelling your LAN-based SNA (including DLSw) and IP traffic to/from the mainframe using OSA-2/OSA-Express or Cisco CIP/CPA.

Solution This could be done by reconfiguring your data centre routers and will enable you to start migrating traffic away from 374*x*s towards the IBM-endorsed interfaces for mainframe connectivity.

Task 5 Eliminate data centre-resident SNA gateways and tn3270(E) servers.

Solution The tn3270(E) server component of Communications Server for System/390 is the most scalable and resilient (ie parallel sysplex) way to implement a ‘tn’-server. It also eliminates the need to move SNA in or out of the mainframe. SNA becomes confined to the mainframe. 3270 traffic to/from the mainframe will all be in TCP/IP form, and, as such, is ideally suited for routing across OSA-Express or Cisco CIP/CPA.

Figure 1: Relatively easy tasks

upgrade by then. Now, when calculating the potential capacity for the upgrade you'll have to factor in the new networking-related workloads.

This move away from SNA, however, should not be all about increased costs and efforts. Many of the new technologies, especially the so-called 'thin-client' (eg Java or ActiveX) tn3270(E) emulators and

Task 1	Migrate away from X.25 or frame relay connections to the 374*x.*
Solution	Rather than interfacing these packet switching networks to a 374x, move the connections to a bridge/router network. Cisco in particular will help you achieve this. You might have to get third-party protocol converters if you also used X.25-centric protocol conversion to SNA.
Task 2	Convert an APPN/HPR network to IP – as an interim measure.
Solution	EE is tailor-made for this, although you could also use DLSw. However, the longer-term goal would be to replace the remote SNA 'clients' – including applications – with IP equivalents.

Figure 2: Slightly harder tasks

Task 1	Get rid of SNA Network Interconnection (SNI).
Solution	SNI is contingent on the availability of at least one ACF/NCP SNI gateway. ACF/NCPs run only on 3745s. To get rid of 3745s altogether, you have to come up with a different way to interconnect the networks. If the networks are owned and run by different companies, this will require significant discussions and collaboration. The best option may be to plan for a host integration solution using products such as IBM's Host Publisher or NetManage's OnWeb 5.2 in conjunction with WAS, which will enable you to easily reuse the existing business logic in mainframe SNA applications to create new, Web-oriented e-applications or even Web services.
Task 2	Get rid of BSC/3270 connections.
Solution	While you can use BSTUN to get BSC traffic to the data centre, you need ACF/NCP or EP on a 374*x* to get it into the mainframe. The best solution is to use a BSC/3270 to SNA/3270 protocol converter or see if you can convert the clients to tn3270(E).
Task 3	Supporting non-3270 BSC connections.
Solution	1. If you have an 'older' System/390 mainframe you may be able to use the Integrated Communications Adapter (ICA) – the precursor to OSA – for some of the connections. The problem is that ICA-capable mainframes like 3745s will be rather old by 2007. 2. Investigate protocol converters from specialized vendors. IBM no longer makes any.

Figure 3: Hard tasks

zero footprint solutions (eg 3270-to-HTML conversion), will be less expensive than the ‘fat client’ emulators in the past. In some cases, your existing maintenance contracts with the likes of Attachmate will entitle you to migrate to some of this newer Web-to-host technology at a relatively low cost. Maintenance on ACF/NCP and 374*x*s, though not wholly unreasonable, is fairly costly. Moving away from these now rather old technology boxes will allow you to spend these maintenance dollars on other projects.

BOTTOM LINE

IBM’s withdrawal of the 3745 and 3746 is its tacit admission that SNA networking has finally entered the ‘end game’ phase. This isn’t to say that SNA applications won’t be around for another decade. This play is just about the migration of corporate networks to IP. In reality, this move to IP has been occurring inexorably since the mid-1990s, and this is just the final chapter. It also doesn’t mean that 374*x*s will disappear within a few years. On the contrary. 374*x*s will still be around in some data centres when 2010 comes along. But this is the time to start planning. There are many options and issues to be carefully evaluated. Talk to others in user groups such as SHARE. Consult IBM’s Red Book. Get proposals from IBM and Cisco – at least. There’s still plenty of time to realize the final migration, but the time for procrastinating about the future of SNA networking is now at an end.

Anura Gurugé
Strategic Consultant (USA)

XML – the next step

This article on XML builds on the concepts and topics discussed in 'XML – a 'first-cut' tutorial' in TCP/SNA Update *45 (March 2002, pp 42-55).*

XML NAMES, SPECIAL SYMBOLS, AND ENTITIES

The basic building blocks of XML are elements, which are made up of a name and some content. The content of an element is defined and delimited by a start tag followed by an end tag. The start tag, which appears in angle brackets (like HTML tags), gives the element its name. For example:

```
    <e-mail>anu@wownh.com</e-mail>
```

In this case, the element name is 'e-mail', and its content is 'anu@wownh.com'. True to its claim of being extensible, XML doesn't specify the tags (and hence the element names) you can use within an XML document – in marked contrast to HTML, there are no pre-defined tags.

There are, however, strict rules as to the composition of XML names. For a start, XML names must begin with either an alphabetic letter (ie A to Z) or the underscore character ('_'). It's important to remember that XML names cannot start with numeric digits. XML names can be of any length, and, after the first character, can include:

- Alphabetic letters
- Numeric digits
- Underscore character
- Dot character ('.')
- Hyphen character ('-')
- Colon character (':'). Note, however, that this is a special, reserved character associated with XML namespaces.

Note that spaces are not valid within XML names. The only other restriction is that names cannot start with the string 'xml', which is reserved for use by the XML specification itself. The ‘xml’ string may, however, occur within a name. It’s also very important to note that, unlike in HTML, XML names are case-sensitive. This means that the following XML names are all different and therefore cannot be used interchangeably: <zipcode>, <ZIPCODE>, <Zipcode>, and <ZipCode>.

This case-sensitivity has given rise to two popular conventions (not rules) when it comes to XML names. The first is that, where possible, people stick with using just lowercase. If the name consists of multiple words and it helps to separate them out, hyphens are used between the words – for example, <product-item-number>. Others prefer what’s referred to as the ‘Camel Case’ convention, with a capital for the first letter of each word – for example, <ProductItemNumber>. Again, remember that these are conventions and not rules.

By carefully restricting the special characters that may appear within names, XML avoids the issue of restricted characters (eg angle brackets) that may appear within XML names. However, XML cannot control the characters that may appear within the content (or data portion) of an XML element. Given XML’s tag-oriented syntax, the appearance of restricted characters such as angle brackets within the content of an element could wreak havoc with the XML-related structure of that document. There are therefore five special symbols in XML that have to be entered differently. In essence, there are ‘escape sequences’ assigned to these five special symbols so that their presence does not disrupt the syntax of XML. XML handles this ‘escape sequence’-to-special symbol mapping by a generalized reference insertion mechanism built into XML known as ‘Entities’.

In XML, an entity is a symbol that represents – or identifies – a pre-defined resource, which may be a file or a text character. Entities are included within a document via entity references. An entity reference is defined using an ampersand (&) at the beginning and a semicolon (;) at the end – for example, &copyright;, &UK;, &NH;, or &wstp;. An XML parser will automatically replace the entity reference according to the value assigned to that entity. Values are assigned to entity references through an entity declaration that has the following form:

```
<!ENTITY  entityname  entitydefinition>
```

Some of the entities mentioned above could therefore be defined as follows:

```
<!ENTITY  UK  "United Kingdom">
<!ENTITY  NH  "New Hampshire">
<!ENTITY  WSTP  "IBM WebSphere Transcoding Publisher">
<!ENTITY  copyright  "&#169;">
```

Unlike the others, the copyright entity is non-intuitive, in that it in turn illustrates another XML feature. This feature is the ability to directly enter character references in the form of Unicode character references. In this example, the ‘169’ represents the character code for the copyright symbol (c). If you’re using Windows, use the ‘Character Map’ utility, found under ‘Programs’ and then ‘Accessories’ off the Windows Start button, to find these character codes. Character code references are prefaced by &#. Hence the ‘&#169’. The character code for an ‘e with an acute accent’ (‘é’) is 233, so you could define an XML entity reference for it that went:

```
<!ENTITY  eacute  "&#233;">
```

These entity references could be used in an XML document as follows:

```
<article-body>
What you always wanted to know about XML
&copyright; Anura Gurug&eacute;
</article-body>
```

Once the workings of entity references is understood, the way that XML handles the five special characters becomes obvious. XML just predefines five entities to represent these special characters as follows:

- Left-angle bracket or less than symbol (<) as &lt.
- Right-angle bracket or greater than symbol (>) as &gt.
- Ampersand symbol (&) as &amp.
- Double quote symbol (") as &quot.
- Apostrophe symbol (’) as &apos.

Thus in XML, the company name Johnson & Johnson will have to appear as:

```
<CompanyName>Johnson &amp; Johnson</CompanyName>
```

There are just two other related concepts that need to be addressed within this concept of names and entities, and they have to do with language specification and how to denote space characters that are ‘meaningful’. There is a special attribute, xml:lang, provided within XML that enables you to specify the language in which the content that follows it is written. In essence, you can have an xml:lang attribute per element that specifies the language in which the content of that element is written. So, for example, you could distinguish between UK-English and US-English as follows:

```
<articlebody xml:lang="en-GB">The foreground colour is green.</
articlebody>
<articlebody xml:lang="en-US">The foreground color is green.</
articlebody>
```

The codes that can be used to specify the ‘root’ language – in this case English (‘en’), but it could equally well have been French (‘fr’) or Hindi (‘hi’) – are defined by ISO 639. Visit http://lcweb.loc.gov/standards/iso639-2/englangn.html for more information. In the case of languages with ‘dialects’, a subcode, in our example ‘GB’ and ‘US’, can be used to get very specific. In the case of French, Canadian French can be specified as xml:lang="fr-CA". The subcodes for this purpose are defined by ISO 3166. Visit http://www-old.ics.uci.edu/pub/ietf/http/related/iso3166.txt for more details.

The other special attribute that can be used within an element is xml:space. It can have one of two values: preserve or default (with default obviously being the default if this attribute is not included within an element). Stating xml:space="preserve" instructs XML applications that the spaces appearing within the content of this element are meaningful and should be preserved.

XML NAMESPACES

Extensibility is the beauty and the bane of XML. Although desirable, enabling users to define tags and as such element names at will can lead to ambiguity and misunderstandings – especially if the same names appear in different XML documents to mean different things. DTDs and XML schema are not the answer here, since they tend to be document specific. One option would be to implement a global naming registry as IBM tried to do to ensure unique network identifiers (NETIDs) for SNA and APPN networks, or the one used for Web addresses (URLs). However, this would be unwieldy and impractical, and would compromise the underlying principles and flexibility of XML.

Let’s assume that you create an XML document listing your favourite PC games and post this on a Web site as a public domain document. One of your friends could take this list and decide to rate the various games. To do this, your friend might add a new ‘rating’ element. Another one of your friends might decide to rate the games by their suitability for various age groups using the General (G), Parental Guidance (PG), PG-13, etc rating system used in North America for movies, TV shows, and even some PC and video games. Given that this is also a rating system, they too may decide to add this classification using a ‘rating’ element. However, these two ‘rating’ elements, which apply to the same base document, mean very different things.

One way to overcome such a conflict problem is to qualify (or prefix) the various element names – for example, ‘fun-rating’ and ‘age-rating’. Athough this works, it can be limiting because qualifying the element names depends on you anticipating potential conflicts. XML’s strategic solution for preventing this type of conflict is the use of namespaces. Namespaces is an elegant and non-intrusive mechanism to enable the unique identification of XML elements without in any way restricting the flexibility or extensibility of XML. Namespaces are implemented by attaching a prefix, identified by a colon (‘:’), to each element and possibly even to each attribute. Thus, with namespaces, the rating elements would be defined as fun:rating and age:rating, where the prefixes now refer to namespaces. These prefixes are mapped to what’s called a Unified Resource Identifier (URI). These URI mappings typically appear near the start of the XML document and have the following form:

```
<pc-games  xmlns:fun=http://www.wownh.com/funrating
                   xmlns:age= http://www.wownh.com/agerating
```

where xmlns identifies this as a XML namespace (ns) declaration. Note that this URI scheme is similar to how you define an external DTD as described in the previous article (*TCP/SNA Update* 45, March

2002, pp 42-55).

The exact use of URIs in XML is rather confusing, especially since there's no requirement that a URI should be valid! In other words, an URI doesn't have to point to anything. In the context of namespaces, URIs are used only as identifiers. Since URIs don't need to be valid, they are essentially treated as case-sensitive text strings. Having said that, however, it's fair to say that for practical purposes most XML 'developers' use valid URLs as their URIs. These URLs will then point to a file that contains the exact definition of the element being qualified. The Internet Engineering Task Force (IETF) is working on another alternative to URIs known as Uniform Resource Names (URNs). Whereas URLs usually start with a protocol designation such as http or ftp, URNs start with a 'urn:' prefix. URNs are supposed to define a unique, location-independent name for a resource that then typically maps to one or more URLs. The XML namespace recommendation can be found at: http://www.w3.org/TR/REC-xml-names/

XML APIS AND EDITORS

A free XML Editor that allows you to display and manipulate XML documents according to their XML structure is available from Microsoft. It's called XML Notepad, and can be downloaded from the following address:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnxml/html/xmlpaddownload.asp.

Other, but not free, XML editors are available from the likes of XML Spy at www.xmlspy.com and Corel/SoftQuad at www.xmetal.com. These editors come in two forms:

- XML Spy, like Microsoft's Notepad, deals directly with XML code and expects you to know XML.
- XmetaL, by contrast, hides the XML from you. You concentrate on the structure of the document using word-processor-like techniques, and the editor, behind the scenes, generates the XML on your behalf. These types of automatic XML 'generator' are

similar to MS FrontPage or MacroMedia DreamWeaver in the context of HTML. These user-friendly HTML tools allow you to compose Web pages using visual, drag-and-drop, graphical techniques. The HTML is generated automatically by the editor without the user having to know anything about it.

If you plan to use XML extensively you might want to get one of each of the two different editor types. Use the automatic generation one to get you started. Then use the XML-centric editor to refine, optimize, and customize the automatically-generated XML.

There are currently two popular XML APIs: Document Object Model (DOM) and Simple API for XML (SAX). These APIs allow applications to read XML documents independent of the XML syntax. Note that the APIs are geared towards different applications: DOM is geared for display type applications involving XML-oriented browsers and editors, while SAX is targeted for inter-program interactions.

CONCLUSION

These XML tutorials have attempted to provide a succinct overview of XML. Their goal is to provide a foundation and framework that you can build on using readily-available XML resources on the Web. You should now visit the Web sites mentioned in these two articles. The resources you'll find there, together with these articles, will get you firmly on the road to XML.

Anura Guruge
Strategic consultant (USA)

TCP/SNA news

IBM has made a numer of announcements, including the following:

- Tivoli NetView for z/OS V5R1, with enhancements in TCP/IP management, NetView Web Application, and broader Linux support. Amongst other things, as a stand-alone management application it manages TCP/IP resources and SNA resources and provides facilities to automate any network or system event. It dynamically discovers DVIPAs for all defined TCP/IP stacks and, for each DVIPA configured for a Communication Server for OS/390 or z/OS Communication Server TCP/IP stack, it provides configuration and status information.
- Version 4.1 of its Tivoli NetView for TCP/IP Performance, which provides automated TCP/IP management of zSeries systems and connected IP networks. Previously Tivoli NetView Performance Monitor for TCP/IP, it promises to help identify and reduce network resource congestion, optimize IP networks, minimize downtime, and increase application performance, availability, and service.
- V1.1 of its CICS Online Transmission Time Optimizer for z/OS, which identifies and removes repetitive data and compresses 3270 datastreams. This is designed to improve 3270 network resources utilization and response time.
- Plans to uprate VSE/ESA V2.7 with better interoperability features on zSeries boxes. VSE and Linux (both under z/VM or LPAR) will be consolidated on the same server. New support will include HiperSockets for high-speed memory-to-memory TCP/IP comms, and OSA Express support is planned to include non-QDIO support for SNA in addition to TCP/IP.

URL: http://www.ibm.com
URL: http://www.ibm.com/servers/zseries/os/vse
URL: http://www.tivoli.com
URL: http://www.tivoli.com/products

* * *

CommerceQuest has announced the CommerceQuest CICS Process Integrator which accesses, exposes, and re-integrates CICS transactions and VSAM files via XML, supporting existing applications and data while creating new and more modern interfaces without conversions or migrations.

URL: http://www.commercequest.com

* * *