41

TCP/SNA update

March 2001

In this issue

TCP/SNA Update

Published by

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

North American office

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

Subscriptions and back-issues

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

Editor

Fiona Hewitt

Disclaimer

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

Contributions

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 ask for a copy of our *Notes for Contributors*, or download a copy from www.xephon.com/contnote.html.

TCP/SNA Update on-line

Code from *TCP/SNA Update* can be downloaded from our Web site at http://www.xephon.com/tcpsnaupdate.html; you will need the user-id from your address label.

Printed in England.

What’s next for networking?

In the first of two articles on this subject in this issue, we take a broad overview of the networking challenges now facing IT departments.

Most IT departments have now managed to link their SNA networks with the TCP/IP networks that flourish in the user departments. They’ve set up Web access for users and their customers, and they’ve made sure that salespeople can log in remotely, and quickly access relevant data or upload sales information.

Are they finally on top of the situation? Or is there something new that they need to be preparing for?

As we all know, there is now a multitude of wireless devices, from Web (WAP) phones to palm-sized computers, and the bad news is that these are beginning to need access to the mainframe.

Luckily, the leading host-access vendors have solutions ready to go on the market. Unfortunately, there’s a number of other components that all have to fall into place before a mainframe will seamlessly talk to a PalmPilot, and much of it is beyond the control of IT managers. Such technologies include the acceptance of new standards and protocols, new client hardware, new client operating systems and ‘microbrowsers’, uniform signal coverage, and new middleware/EAI (Enterprise Application Integration) solutions.

Many Web-to-host vendors have announced support for WAP (Wireless Access Protocol), the set of standards subscribed to by Web phones and palmtop computers. The Wireless Mark-up Language (WML), used to create WAP pages, is a derivative of XML (eXtensible Mark-up Language), and is the language of wireless Internet applications. WML is optimized for compression over low-bandwidth connections, such as wireless Web access.

It looks like wireless access will rely entirely on XML, with IP as a communication protocol. In the meantime, WAP/WML and XML applications do not interoperate well. To move the data from OS/390 to a WAP-enabled phone, applications need to be converted from

HTML (HyperText Mark-up Language) into XML, then from XML into WML. The World Wide Web Consortium and other industry groups are working to bring these standards together, but this process may take time!

In addition, linking devices to host environments will require new middle-tier services, in the form of wireless ISPs (WISPs), which include hosting functions, as well as content translation services. It's likely that many current ISPs and ASPs (Application Service Providers) will add this functionality.

End user hardware is another issue with wireless devices. For example, most WAP phones function as dumb terminals, with no storage or memory capability. Signal-based access is far more unstable than access through a wire. Thus, if a user is out of range of the signal, he or she is no longer connected – not ideal for sales people visiting clients.

The industry has been fairly aggressive about rolling out the software that powers wireless devices. The microbrowsers that run on these devices, such as WAPMan, are easier on bandwidth, since they take on some of the processing tasks. The Netscape 6 browser, powered by Netscape's 'Gecko' technology, is designed for multiple devices, such as Web phones or Internet appliances.

It's likely that wireless Web-to-host transactions will take place within EAI applications that pull together data from various ERP and database sources and format the data for wireless transmissions immediately.

My guess is that within the next five years there will be direct wireless access to mainframe applications. In Web-to-host wireless access, the focus of the action will all take place in the middle, running on Unix or Windows NT/2000 servers.

Nick Nourse
Independent Consultant (UK)

The new chapter in Web-to-host integration

This article follows on from the last one, and looks in detail at new developments in Web-to-host integration.

Although it may seem hard to believe, Web-to-host technology, in proven production quality form, has now been around for over four years. Today's Web-to-host solutions provide such seamless and transparent integration with standard Web pages that it's almost impossible to believe that you might be interacting with an SNA application running on a mainframe or AS/400. Try it for yourself by visiting www.guthrietheater.org, the Web site of a large US theatre. Click on the 'tickets' button in the middle of the page, and when the calendar of events appears, click on one of the 'buy' links and pretend that you're going to buy a ticket on-line. At this point, you're talking to an SNA application running on an AS/400. Note that there are no 'green-on-black' screens anywhere. The host data appears in HTML form tightly integrated within a Web page, while the navigation through the SNA application has been simplified to a couple of 'proceed' and 'cancel' buttons. This AS/400-to-Web integration is being performed by iE ScreenSurfer, a Windows NT Server-based 3270/5250-to-HTML conversion product (www.screensurfer.com). In this instance, iE ScreenSurfer is being 'front-ended' by an e-business application developed to work with Allaire's ColdFusion Application Server (www.allaire.com). This on-line ticket system is a great example of what's possible today with Web-to-host technology.

Take a look too at taxweb.charlestoncounty.org, an automated, on-line tax system for Charleston County in South Carolina, USA. In this case, the host data being transparently accessed from a Web browser resides in a DB2 database running on a System/390 mainframe.

There are lots of such publicly accessible Web-to-host applications around (see 'Web-to-host integration in practice', *TCP/SNA Update*, Issue 39, September 2000, for more examples).

However, all of these examples represent what's possible with 'standard', first-wave Web-to-host technology – whether it be applet-based (ie Java or ActiveX), 'thin-client' terminal emulators (eg IBM's

Host On-Demand), or ‘on-the-fly’ 3270/5250-to-HTML conversion solutions (eg ResQNet’s ResQPortal). While this base technology, in many cases in third- or fourth-generation form, is still in high demand, 2000 saw the emergence of the next wave of the technology. The principal themes that characterize this new wave of Web-to-host technology are:

- The ascent of host integration, as opposed to straight host-to-HTML, solutions.
- Pervasive support for handheld, wireless devices including Wireless Application Protocol (WAP)-based smart phones.
- Support for bi-directional host-to-XML conversion, where Extensible Mark-up Language (XML) is an increasingly strategic tag-based ‘self-defining’ mark-up language, within the growing Generalized Mark-up Language (GML) family, which is used to provide data with meaning, structure, and context.

HOST INTEGRATION SOLUTIONS

Host integration solutions – including IBM’s WebSphere Host Publisher 2.2 and Attachmate’s HostPublishing System (HPS) 2.4 – can be thought of as host-to-HTML products on steroids. Another way to think of them would be as mini Application Servers that focus on providing access to mainframe and AS/400 data. The four defining characteristics of a host integration product are:

- Concurrent multi-host access capability to disparate host types with, at a minimum, support for mainframes, AS/400s, ‘VT’ (ie Unix/Linux, minicomputer), and SQL-based database access.
- Rapid e-business application (ie e-application) development facility, typically via an Integrated Development Environment (IDE) that supports highly visual, ‘drag-and-drop’ software development methodology.
- Inclusion of some level of object-oriented application development.
- Intrinsic, built-in support for HTML output/input that emphasizes Web-centricity and dictates deployment along with a standard

Web server (eg IBM’s WebSphere or Microsoft’s Internet Information Server).

Those of you familiar with traditional middleware software such as IBM’s MQSeries and CNT/Apertus’s Enterprise/Access might feel, quite rightly, that a host integration solution is just another middleware offering. The key differentiating feature, however, is that host integration solutions have been developed from the ground-up to support Web-based e-business applications and Web page-oriented transactions.

Host integration solutions enable the workflow and business logic of existing host applications to be modified and augmented. For a start, they permit host screens to be combined or bypassed, or accessed in a random order. This allows application workflow to be streamlined and optimized, and thus to exploit the ‘point-and-click’, HTML-based GUI that invariably ‘front-ends’ the application. Host integration products also enable data from multiple host applications, running on different and disparate systems (eg mainframe and Unix), to be easily synthesized to create a whole new e-business application. For example, you could take a ‘mail-order’, e-commerce product merchandizing application running on an AS/400 and extend it by adding an existing customer information DB2 database resident on a mainframe as well as a credit authorizing application running on a Unix system. Then, there’s object-orientation to facilitate new e-application development.

Let’s look more closely at these features using an actual product as an example. While IBM’s Host Publisher may be the most visible of today’s host integration solutions, I’ve chosen to focus on Zephyr’s ECLIPSE (www.zephyrcorp.com) because of its power, speed, ease-of-use, and mainframe emphasis. ECLIPSE, which came on the market in September 2000, was also the first Web-to-host product to feature a WYSIWYG ‘drag-and-drop’ host-to-HTML design tool. This eliminates any uncertainty as to what a completed Web page containing host data is going to look like.

ECLIPSE is written entirely in C++, whereas IBM’s Host Publisher 2.2 is totally Java-based. This C++ aspect gives ECLIPSE an interesting edge over comparable Java-products as well as NT-only solutions such as Attachmate’s HPS 2.4. C++ makes ECLIPSE platform-independent (at least from a pragmatic standpoint); it can be deployed on all major server platforms including mainframes (running OS/390, z/OS, or MVS), Sun Solaris servers, and Windows NT/2000 servers, with the conspicuous omission of AS/400s. ECLIPSE, consisting of compiled C++ code, is also extremely fast – typically 50% to 70% faster than a comparable Java offering.

The host systems currently supported by ECLIPSE with regard to host integration include:

- IBM (or compatible) mainframes running OS/390, z/OS, MVS, VM, or VSE
- AS/400s
- Unix systems
- Other host systems with VT-interfaces (eg HP or Unisys systems).

An SQL-based database access scheme will be available in a future release.

The key features related to program development that distinguish Eclipse include:

- ‘On-the-fly’ 3270/5250-to-HTML conversion, with built-in autoGUI capability, that ensures immediate, zero-effort Web-enablement of all host applications while new e-applications are being developed using the other programmatic features of the product.
- Automatic, behind-the-scenes JavaScript generation (when required) for use within output Web pages to facilitate better emulation of certain host-oriented data entry functions (eg behaviour of protected and unprotected data fields).
- Simple, easy-to-use object orientation in the form of Host Data Objects, which represent host-related units of work (eg a ‘Get_Account’ function), which are created as reusable objects, with clearly defined HTML-based inputs and outputs. These then expedite and simplify the integration of existing host functionality into new e-applications through an ‘object-linking’ scheme,

without the application designer needing to get involved with the host business logic. Java data objects are also supported.

- Integrated support for the emerging IETF Open Host Interface Object (OHIO) API specification that is backed by IBM and Attachmate. This is a strategic, object-oriented API that can be used with e-applications being developed in the form of Java servlets, JavaServer Pages (JSPs), or Microsoft Active Server Pages (ASPs).

An object lesson

Given that object-orientation is a pivotal feature of host integration solutions, it’s worth looking at Host Data Objects in more detail because they represent one of the more intuitive schemes on the market. Host data objects represent a particular host application function. As an example, let’s take a mainframe application that manages customer account details. Let’s also assume that the enterprise is developing a new set of e-applications that permit the general public, over the Internet, to order products, check product delivery status, and query account details. Rather than developing a new account management application, the enterprise, quite rightly, intends to use the proven host application as a transparent data ‘back-end’ for the new e-application suite. The host application will be expected to obtain and deliver customer account details when presented with a customer name or customer account number.

A developer familiar with 3270 terminal emulation and some amount of HTML, using the visual paradigm of the WYSWIG DESIGNER, can quickly develop a host data object called ‘Get_Accounts’. No programming language skills are required. The inputs to this ‘Get_Accounts’ host data object (or unit of work) could be the customer’s name or a customer account number. The output of this host data object would be the appropriate account details for that account (or an error message stating that the account could not be found). The input to this ‘Get_Accounts’ host data object will be defined in terms of an HTML Web page with the appropriate input fields. Data eventually entered into the HTML input field(s) by the end users of the new e-application suite will be passed onto the ‘Get_Accounts’ host data object by the ECLIPSE Server. Under the directions of the ECLIPSE Server, the ‘Get_Accounts’ host data object will interact with the host application (including opening a new session if required) and navigate through the necessary host screens to obtain the requisite account details.

The screen-to-screen navigation requirements of the host application would be included within the ‘Get_Accounts’ host data object as a ‘macro script’. The output of the ‘Get_Accounts’ host data object will also be an HTML Web page – in this case, one listing the account details (or an error message if the account could not be located).

IBM’s Host Publisher 2.2 and Attachmate’s HPS 2.4 also support objects; IBM calls them Host Integration Objects (HIO), and Attachmate QACOM – where ‘COM’ refers to the fact that they’re based on Microsoft’s Component Object Model (COM) architecture. The defining characteristic of Host Publisher is its tight integration with IBM’s strategic WebSphere application development and deployment foundation –- including the use of top-end WebSphere Studio Professional Edition as its design tool. On the other hand, HPS 2.4’s strength is the ‘NavDesigner’ component that automatically creates the relevant QACOM objects that represent the logic needed to realize unconstrained navigation and field selection with regard to a given host application.

The three host integration solutions described above, along with WRQ’s Apptrieve, represent the most widely used solutions in this field, and you should carefully evaluate them all.

WIRELESS-TO-HOST

The Internet is about to become ‘unplugged’. By 2004, there will be more wireless subscribers capable of Internet access than wired users. Today, there are around 41 million US households with Internet access, and 120 million US wireless subscribers (including cell phones, Palm Pilots, pagers, PDAs, and cellular modems for laptops). Web-to-host products are therefore being extended to support wireless-to-host interactions. Large financial institutions such as Fidelity Investments (www.fidelity.com) already offer no-charge real-time wireless-based transactions from PDAs, pagers, Internet phones, and cellular-modem equipped laptops for its customer base. Since Fidelity is a huge IBM mainframe user, many of these wireless transactions have to be processed on a mainframe via wireless-to-host technology. Three key products that facilitate wireless-to-host integration are IBM's WebSphere Transcoding Publisher, ResQNet's ResQ/ME, and Seagull's Wireless-to-Host.

Internet-based e-business is already transforming the world-wide commercial landscape, when it comes to supply chain fulfilment, customer relationship management, Internet financial transactions, Internet insurance, collaborative product development, mail order shopping, Web-based travel reservation, and real-time human resource fulfilment. However, none of these e-business applications can deliver their true potential until they can support wireless-based interactions. The reason for this is simple and obvious: today's professionals are highly mobile and spend a lot of time away from their desks. Wired access to the Internet for e-business is too constraining and impractical.

The good news is that host integration products and Web servers together ensure that you don't have to worry about the actual wireless protocols used by the various handheld devices. These server products handle those transparently – in some cases via device-specific wireless ISPs. In many cases, particularly so with Pocket PCs, these devices run an abridged version of a standard Web browser, and therefore support standard HTML Web pages, although some insist on Wireless Mark-up Language (WML), a variant of HTML.

The issue that has to be addressed, however, is that of dealing with the user interface of the wireless device. Wireless devices, even if they are Pocket PCs or PDAs, have considerably smaller screens than laptops and PCs. A host user interface designed to be used with a 3270 terminal or PC screen cannot be used unchanged with a wireless device. Not only will it not fit the screen (without the need to continually scroll up-and-down, and side-to-side), but the host dialog will invariably be too cumbersome and verbose for easy use from a wireless device. In this context, it's also worth noting that wireless devices do not offer many screen navigation (ie screen traversal) options. Consequently, for wireless-to-host interactions to be viable and meaningful, the host-to-user dialog has to be rationalized, simplified (eg replacing embedded images with links to those images) and restructured to work well within the constraints of the small screens and limited navigational latitude. WebSphere Transcoding Publisher and ResQ/ME offer leading-edge solutions to this user interface reengineering challenge, so if you're looking to implement a wireless-to-host solution start by looking at these products. But bear in mind that a lot of other host access vendors (including Seagull, Jacada, and iE) also now support wireless-to-host.

HOST-TO-XML CONVERSION

XML is being widely promoted as the strategic, if not the only, way to exchange information with new e-business applications. It would therefore help if existing mission-critical mainframe or AS/400 applications could talk XML (rather than 3270, 5250, or SNA LU 6.2), so that their proven business logic and functionality could be readily integrated with new e-applications. It's obviously not feasible to modify these old applications to talk XML; what's needed is a gateway function that would accept 3270 or 5250 datastream and automatically convert it to an appropriate XML schema. IBM (which obviously has a vested interest in making sure that mainframe and AS/400 applications are XML-capable) has already proposed a set of Document Type Definition (DTD) XML tags for host datastreams. Visit: http://swaydemo.dfw.ibm.com/dtd/xmlHostData.dtd.html

Given that we already have proven 'on-the-fly' 3270/5250-to-HTML conversion products, the obvious solution was to extend these gateways to also support bi-directional host-to-XML conversion using a set of agreed-upon DTD tags – such as those proposed by IBM. IBM was the first to offer this capability with its WebSphere Host Publisher 2.2 solution in July 2000. Since then, ResQNet's ResQPortal, iE ScreenSurfer 3.0, and Zephyr ECLIPSE have offered similar capabilities. Now, using this kind of host-to-XML feature, you can automatically represent your 3270 and 5250 screens in XML form, effortlessly, without any manual intervention.

SUMMARY

The first wave of applet-based terminal emulation and host-to-HTML conversion Web-to-host technology products has now been around for over four years. Examples of this technology abound on the

Internet, although the Web-to-host integration is so seamless that most people never realize that they’re interacting with a mainframe or AS/400 over the Internet.

2000 saw the evolution of the next wave of Web-to-host technology, the three key themes of which: host integration, wireless capability, and XML. In each case, there are numerous products that will meet your needs. The products mentioned in this article are by no means the only examples of this technology – at the last count there were over 60 active Web-to-host vendors. This a crowded and busy field, but the examples given here should give you a leg-up when it comes to evaluating the technology.

Anura Gurugé
Strategic consultant (USA)

Code from *TCP/SNA Update* articles

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

http://www.xephon.com/tcpsnaupdate.html

You will need the user-id shown on your address label.

3270 terminal emulation using Reflection

You would be hard-pressed today to find a 3270 terminal on any office desktop, but, whether they communicate via SNA or TCP/IP, there are very few on-line mainframe applications that do not use 3270 protocol. That real 3270 hardware has been replaced mostly by 3270 emulation software that runs on Windows workstations, with a few client/server applications using 3270 screen scraper technology.

But although there are a lot of 3270 emulators available, usability varies widely. We begin this series of articles on 3270 emulators with a look at Reflection from WRQ (http://www.wrq.com), and, more specifically, Version 8 of Reflection for IBM, used on a Windows 2000 workstation, with tn3270e protocol.

One word on terminology before going on. Although I normally stick to tn3270 and tn3270e when referring to Telnet and Telnet Extended in IBM mainframe (3270) environments, WRQ does not usually differentiate, referring to Telnet or Telnet Extended even in the 3270 environment; TN3270(E) is also occasionally used. You will therefore see all of these terms in this article.

Reflection for IBM includes both an AS/400 and IBM mainframe emulation client for the workstation. A menu item is added for each in the Start-Programs-Reflection menu, but Start-Settings-Control Panel-Add/Remove Programs shows only a single installed ‘Reflection for IBM 8.0.2’ item. Given this fact, you may be tempted to delete the AS/400 menu item during deployment to users. In fact, however, you’re better off removing both the IBM mainframe and AS/400 Start menu items, since users will be using shortcuts to Reflection Settings files to start Reflection, load relevant settings, and connect to the application of their choice.

CHECKING OUT THE REQUIREMENTS

Although a little confusing because it covers so many host environments, the Installation and Quick Start Guide manual is quite detailed, trying very hard to cover every possible hardware, software, and network protocol scenario.

This broad range of coverage leads to some confusion. My search for details relevant to my environment (IBM mainframe, tn3270e, and Windows 2000), finally led to the following:

“To use Telnet Extended, you must have TCP/IP networking software as explained in the ‘Telnet’ section above. Telnet Extended is for users of TCP/IP software who connect to their host through a Telnet gateway that implements RFC 1647. Such gateways, like the Apertus DataCenter Hub gateway, support the TN3270(E) protocol. Using TN3270(E) as a connection type allows you to specify the connection device name (also known as the LU name) and print mainframe jobs to a local or network printer. This connection type provides standard support for the Attention key, the SysReq key, and SNA response handling.”

The referenced Telnet section states: “To use Telnet for 3270 connections, you must have Windows Sockets compliant TCP/IP networking software. If you’re using a TCP/IP stack that is not Windows Sockets compliant, contact WRQ technical support.”

A lot more reassuring is the Support for Windows 2000 Features section on page 6 near the beginning of Chapter 2, which at least includes the fact that it is “...designed to take full advantage of the new Windows 2000 features, especially those related to installation and deployment”. This is in sharp contrast to shareware tn3270 products, which may list NT4 support, if you’re lucky. (That’s not to say, however, that there aren’t some that support Windows 2000.)

This section of the Reflection manual would seem the ideal place to indicate how many of the Telnet and Telnet Extended requirements are met by Windows 2000. But, most of all, I would like to see a clarification in the Telnet Extended section as to whether the Telnet gateway being referred to (RFC 1647) is on the host or client side of the connection. It would also be helpful to add some advice on how to easily check out these requirements, especially on the host end. In my case, I was given only an IP address and a TSO ID and password for a host 200 miles away.

Of course, you can always adopt my approach: try installing Reflection and see whether it works. Though, admittedly, I was buoyed by the confidence of having been through a SNA to tn3270 conversion with Reflection at my local telco, then reconfiguring for tn3270e and having it work.

SPECIFYING INSTALLATION OPTIONS

Although the ‘Installing Reflection to a Single PC’ section of the manual lists 19 installation steps, experience reveals that only 12 of these actually happen. Steps 10-16 relate to the Custom install option, where you choose exactly what gets installed. ‘Typical’ would be used for a single user install; ‘Complete’ for a system administrator.

Thirty seconds after loading the CD-ROM, a screen is displayed, with the option to install, which leads to the following options:

- Administrative Installation, for when when you want to deploy Reflection to other workstations.
- Workstation Installation for a single workstation install. If the workstation does not have the Windows Installer (standard with Windows 2000), the Reflection Interactive Installation program installs it, and requires a reboot right after.
- Additional components.

At this point, you are urged to shut down all other programs. Your operating system definition of Full Name and Organization are displayed, and you are given the chance to change them. In some organizations, to simplify operating system installs, all workstations are set to a single standard value, such as 999999 for Full Name.

Further down the screen, you are given the choice of installing for:

- Anyone who uses this computer
- Only for me.

Your workstation user ID is displayed with the second choice. In most organizations, it’s the same as your LAN ID. Step 1 of the installation process in the manual (page 22) indicates that you must be logged on with a workstation ID that has Administrator privileges, no matter which of these two choices you choose. In my case, I was logged on as Administrator, and chose ‘Anyone who uses this computer’.

DESTINATION FOLDER

I recommend accepting the default folder for any software you install, in this case C:\Program Files\Reflection. Years ago, in the days of MS-DOS, I used to always change the default folder to match my division of my hard drive into C:\DATA and C:\SOFTWARE, but ran into problems when some obscure component or update utility failed because I had not used the default install destination path. Today, so long as products stick to creating folders in C:\Program Files, I do not change the default.

Even then, changing directory names is generally a problem, since a product lacking the sophistication of C:\Program Files destinations likely also lacks the ability to allow the installer to change the installation directory name. You end up with a DOS-like 8 character folder name in the root directory, such as C:\ABBACUS.

It does not necessarily mean the product has never been tested outside of Windows 3.*x* environments. I am aware of two popular free products, one in the public domain from a major US university and the other freeware from a large software vendor. Both use this approach to deliver a simple single installation medium for everything from Windows 3.0 to Windows ME and 2000.

THE ACTUAL INSTALL

Unless you choose a Custom install, you can ignore steps 10-16 in the manual. The actual installation process begins as soon as you respond affirmatively to the prompt “Are you ready to start the installation?”

The installation display is quite informative, showing each file and its size as it’s installed from the CD-ROM. Even though the timing estimate began well under one minute, and stabilized at 42 seconds, it later settled in at a more accurate 5-6 minutes.

When installation is completed, no reboot is recommended or required.

SURVEYING THE DAMAGE

So, once installation is complete, what visible changes are there to the desktop? A grey pyramid icon labelled Reflection Links appears on the Windows 2000 desktop, and a Reflection folder is added to the Start-Programs menu. Nothing is added to either of the System Trays at the bottom of the screen.

If you choose the Complete install, as I did, you'll find that the Reflection folder contains: a Utilities folder, a Wizards folder, an FTP Client, Host – AS/400, and Host – Mainframe.

To avoid user confusion, your first instinct might be to remove the Host – AS/400 menu item from the Reflection folder, and to try to remove the AS/400 component altogether. And page 27 of the manual seems very promising: 'Adding and Removing Reflection Features from Your Installation'. However, you'll soon discover that mainframe and AS/400 are provided by the same Reflection components, and you cannot remove support for one without the other. Separate menu items for the AS/400 and mainframe are also irrelevant, since, as mentioned earlier, it is shortcuts to Reflection settings files, not these menu items, that users will use to start Reflection.

The FTP Client passed my sophisticated test with flying colours. In one keystroke, it deleted a folder with a FrontPage subfolder that contained a hidden file, with a dot ('.') as the first character of its name.

The Wizards folder contains just one entry: the Connection Wizard. And that's what you'll use next.

CONNECTION WIZARD

The 'Connecting to a Host' section of the manual begins on page 54. The first subsection is entitled 'Using the Connection Wizard', and that's usually the best place to start. But, as the last paragraph of the subsection indicates, for SNA connections, your links to the host must be configured before you run the wizard.

To begin: Start-Programs-Reflection-Wizards-Connection Wizard. Click the Next button in the Introduction dialogue box. The Session Type dialogue box is displayed next. It varies depending on the product you're installing. With Reflection for IBM, you'll see just two choices: terminal session and printer session.

Telnet Extended (tn3270e)

Select Terminal session, push Next, then Mainframe, and push Next in the IBM Host Type dialogue box. The Mainframe Connection dialogue box is next, with four choices: Telnet, Telnet Extended, Direct Connect SNA, and Gateway.

Since I was given an IP address on the Internet and a TSO ID and password, I knew it had to be a telnet connection, not SNA. But whether it supported tn3270e, rather than just tn3270, was unclear. Nonetheless, I chose Telnet Extended in the hope of getting advanced 3270 features such as colour and System Request key support. As far as I'm aware, CICS is the only environment where Telnet Extended may be the wrong choice even if it is supported – you can hang VTAM LUs if your choice does not match the CICS terminal definition.

Push 'Next', and the Host System dialogue box appears. Two fields need to be filled in. In my case, 'Which host or IP address would you like to connect to?' is merely the IP address I was given, four numbers separated by dots but no blanks. Do not be confused by the fact that the drop down list already has one entry: localhost.

The second field left me mystified. Titled simply 'Device Name', Help indicates it is the VTAM LU Name to connect to, and is applicable only to Telnet and Telnet Extended. Nonetheless, I left it blank and hoped for the best. It turned out to be the right choice.

AUTOLOGON

Hitting 'Next', the Connect dialogue box appears with the question: Do you want the Connection Wizard to remember your login? I responded Yes, hoping it might do what I'd seen another Reflection user at my local telco do, automate the logon process to TSO then ISPF.

At this point, the mainframe host should have responded. If not, it's worth hitting the workstation's Enter key just in case you are connected but the mainframe hasn't noticed you yet. Logging on to TSO through VTAM, then starting ISPF, a little window keeps reminding you to tell the Connection Wizard when you're logged on. It will then ask: 'Do your session settings appear to be correct?'

SAVING SETTINGS

Next, a Save Session dialogue box appears, asking ‘Do you want to save your login and session?’ Respond Yes, and a Save Settings File window is presented, revealing that part of the Install procedure created a Reflection folder in My Documents, at least in part for storing Settings files. (This is an excellent approach, and one that should prevent customizations and other personalizations being lost whenever a workstation or hard drive is replaced, or rehosted.)

In the Save Settings File, you’re positioned in the My Documents\Reflection folder, the file name is listed as *.rsf, and you’re warned No Shortcuts will be created. If you name the settings file after the host, you will remember where it connects you. Make sure that the extension is .rsf, or omit it altogether. Otherwise, you’ll end up with a file name that looks like it has two extensions.

If, at this point, you decide to make some changes to settings, be sure to save them, by hitting the Save icon (a diskettte symbol) on the toolbar at the top of the screen. Unlike many other products, the default is not to to Save settings or to Confirm exit – which proves that WRQ listens to customers. At my local telco, we used to complain bitterly when asked to confirm exit every time we terminated a host session. And if you deploy terminal emulators throughout an enterprise, you probably do not want any but your most sophisticated users changing settings, and then calling you for help because things no longer work properly. Note, however, that, even without Save setting or Confirm exit specified, you will still be prompted to save any changes to macros you may have made in the session.

(Editor’s note: there is an article on using and customizing Reflection on pages 21 to 32 of this issue.)

Armand Minet
(Canada)

3270 keyboard mapping

There may be no 3270 keyboards on desks these days, but the Windows workstation keyboards that have replaced them are transmitting 3270 keystrokes to on-line mainframe applications, no matter whether they're attached to SNA or TCP/IP networks. The workstations, of course, are running terminal emulation software that does the translation.

Until recently, my main criticism of mainframe terminal emulators was that the standard keyboard mapping was clearly designed by someone with very little hands-on 3270 keyboard experience. However, it's been a long time now since 3270 hardware has been on anyone's desk, and trying to match the position of 3270 keys on the PC keyboard is no longer that important. The biggest issue now is the appalling lack of 3270 knowledge on the part of mainframe users.

This may not matter for customer-facing staff running front-ends to multiple applications using screen scraper technology and/or multi-session managers. But it does matter for younger technical staff educated in Unix environments who have never seen a real 3270. Unaware of the relevant escalation process, they handle an infinite loop just as they do any terminal hang: by closing the terminal emulator window to exit from the product. The result is a lot more dead sessions chewing up computing resources, and often human resources too, as users wait for the session to time out so they can log on again, or call the Help Desk to get the session cancelled. This problem can be reduced by allowing multiple log-ons, but some environments (such as TSO) cannot support multiple log-ons to the same user ID.

The worst-case scenario is the runaway task. Infinite loops are the most obvious example. Closing a terminal emulator window may leave the session active, consuming resources until resource counts hit any limit that might have been set by the installation or specified by the user for CPU time, printed pages, etc. Limits, if any, may be so high that significant damage can be done. For example, few installations have DASD space limits on individual users. A runaway task could conceivably consume all available large chunks of contiguous DASD space. DFSMShsm may get triggered to solve the problem, but it could still cause real havoc to other jobs running in the meantime. And, ironically, this is one type of problem that improvements in technology have made worse, not better. Faster hardware means doing more damage in less time, giving Operations staff less time to take preventative action.

Since programmers doing testing are far more likely to create runaway tasks than end users, programmers are the ones who most need 3270 knowledge. And, of course, once you get a more educated 3270 user, you'll be faced with questions, perhaps even complaints, about how to use these 3270 keys with the current terminal emulator. You may even come face-to-face with the fact that tn3270 does not support the 3270 System Request key; tn3270e does. In this article, we use Reflection for Windows as an example, but the issues discussed could be just as easily applied to any other product.

ENTER

The 3270 New Line key is probably the most obvious problem. Although New Line and Enter are very different in a 3270 environment, PC keyboards have combined them into a single key. There's no perfect answer to this problem, but I opt for the 3270 positional approach, as I do with many other keys: position 3270 keys roughly where they're found on the last popular generation of 3270 terminals, such as the 3179 and 3180. That means the PC Enter key becomes the 3270 New Line key, and the right Ctrl key on the PC keyboard becomes the 3270 Enter key. The downside of this approach is that the user gets into the habit of using the right Ctrl key as Enter. But dialogue boxes within the terminal emulator, and, of course, other workstation applications, do not recognize the right Ctrl key as Enter.

KEYBOARD MAPPING CONSIDERATIONS

Although the position of a key on the original 3270 keyboard is important, it's been a long time since anyone I know has used a 3270 terminal. So for most 3270 keys, the most logical assignment is to a PC key with a similar name: Insert is Insert, Delete is Delete. It may be tempting to assign Erase EOF to the PC's End key, but I favour assigning the extremely useful 3270 End of Field key to PC End, and adopting a positional approach for Erase EOF. (If you weren’t aware that 3270s had End of Field keys, you’re not alone – neither was I back in my real 3270 days. No matter where you’re positioned in a 3270 field, the End of Field key moves you to the last character in the field, before any blanks or nulls that may be at the end of the field.)

Then, some keys should be assigned based on how easy it is to remember the key assignment. For example, Alt-C makes sense for the North American cent sign (‘¢’), since it looks like the letter C. But I also keep Reflection’s assignment of the left square bracket (‘[’) since the cent sign was also just right of the letter P on the 3270 keyboard. Assigning multiple PC keys to a single 3270 key is permitted by Reflection, and allows you to implement multiple approaches to aid your memory when the time comes to actually use the 3270 key in question.

The final consideration is whether mapping the PC key would inhibit its existing function. Alt-C and Alt-D are the Reflection keyboard shortcuts for Connect and Disconnect. By assigning Alt-C to the cent sign, the shortcut is lost. But I never use those keyboard shortcuts anyway, preferring the icon in the toolbar.

Using the Ctrl keys for 3270 Reset and Enter eliminates the possibility of mapping any Ctrl characters. For example, Ctrl-C may have been a better solution than Alt-C for mapping the cent sign.

MY CHOICES WITH REFLECTION

Although certainly not encompassing every 3270 key that exists, the chart shown in Figure 1 shows the keyboard mapping I have evolved to after three years – the first year using Rumba, and the last two with Reflection. Actually, my chart is a three-column list, showing workstation, 3270, and VAX keys. In Figure 1, the workstation key is listed first, before the hyphen, and then the 3270 key. Comments are included for clarification.

KEYBOARD SET-UP WINDOW

Reflection provides a Keyboard Set-up dialogue box offering three separate functions:

First, the Reflection keyboard defaults that work for me:

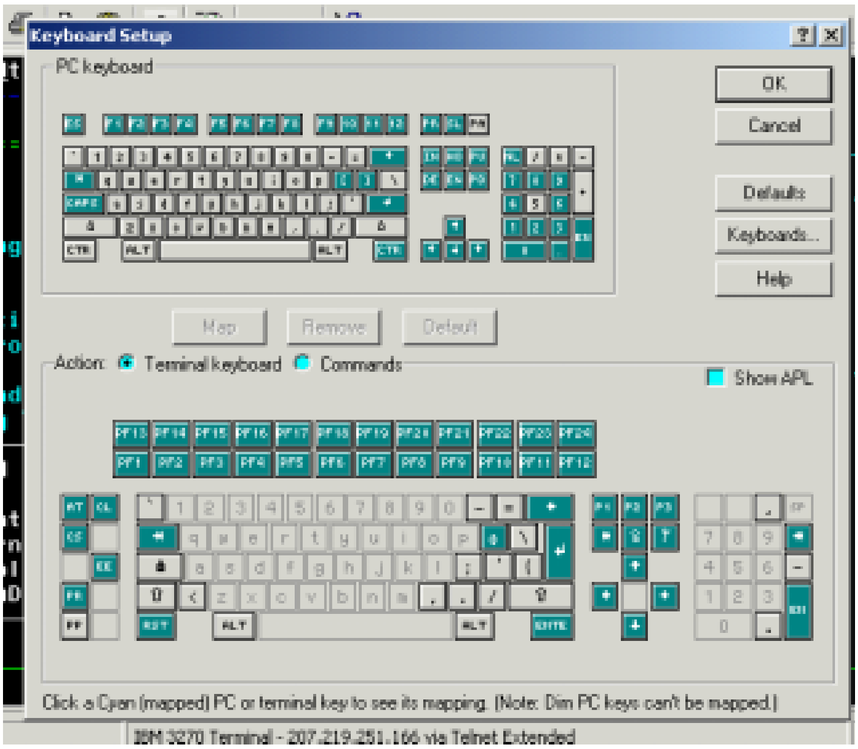
Fn	PFn, where n = 1 to 12
Shift + Fn	PFm, where n = 1 to 12, and m = n + 12
Grey Insert	Insert
Grey Delete	Delete
Grey Home	Home
Alt + Print Screen (labelled SysRq)	System Request
Shift + Tab	Back Tab
^ (Shift + 6)	logical not (‘¬’)
[	cent sign (‘¢’) (same position on keyboard)
Numeric keypad cursor keys	cursor keys
Numeric keypad Enter	Enter
Numeric keypad Home	Home
Numeric keypad Insert	Insert
Numeric keypad Delete	Delete

I add the following:

Enter	New Line
Right Ctrl key	Enter
Left Ctrl key	Reset
Escape	Clear
` (just below the Escape key)	Erase EOF
Grey End key	End of Field
Grey Page Up	PF7
Grey Page Down	PF8
_	|
Alt + Fn	PAn, for n = 1 to 3
Alt + C	cent sign (‘¢’), but that eliminates the Connect keyboard shortcut
Numeric keypad End key	End of Field
Numeric keypad Page Up	PF7
Numeric keypad Page Down	PF8
Numeric keypad /	PA1
Numeric keypad *	PA2
Numeric keypad hyphen (‘-’)	PA3
Numeric keypad plus sign (‘+’)	Attention

Figure 1: Keyboard mapping chart

- Changing the mapping of workstation (PC) keys to 3270 keys.
- Determining current mapping for any workstation key, in case you hit it by accident and need to know what 3270 key was pressed on your behalf.


Keyboard Setup
PC keyboard
OK
Cancel
Defaults
Keyboards...
Help
Map
Remove
Default
Action: ● Terminal keyboard ○ Commands
Show APL
IBM 3270 Terminal - 207.219.251.166 via Telnet Extended

Click a Cyan (mapped) PC or terminal key to see its mapping. (Note: Dim PC keys can't be mapped.)

Figure 2: Keyboard set-up

- Figuring out what workstation key (or keys) can be used to press a given 3270 key. There are two exceptions to this:
  - When no workstation key combination is mapped to the 3270 key
  - When the 3270 key is not shown on the keyboard.

To display the Keyboard Setup dialogue box, select Setup from the menu bar, and then Keyboard... from the drop-down menu. You will see two keyboards: a representation of the workstation ('PC') keyboard at the top of the window and a 3270 keyboard at the bottom (see Figure 2). Three greyed-out buttons separate the two keyboards:

- Map – to define which workstation key is translated into which 3270 key.
- Remove – to remove the definition of an existing mapping of a workstation key to a 3270 key
- Default – for a workstation key, to restore Reflection's default mapping to a 3270 key.

The three buttons are greyed out because, initially, no keys on either keyboard have been selected. Mapping of key combinations, such as Shift+F1, are also supported on both keyboards.

A column of buttons in the upper right corner of the dialogue box includes four worth noting:

- OK – to confirm all changes made since initiating Keyboard Setup (note that none of the changes are saved on disk until the Reflection Settings file is saved).
- Cancel – cancels all changes made since initiating Keyboard Setup.
- Defaults – restores Reflection's default mapping for all keys.
- Keyboards... – allows you to choose both the 3270 and workstation keyboard layouts displayed.

Finally, the top of the 3270 keyboard is labelled Action, and allows you to choose between the default view, Terminal keyboard, and Commands. Commands provides access to 3270 keys not found on the displayed keyboard, such as End of Field. It can also be used to assign Reflection actions and even Visual Basic for Applications (VBA) macros to PC keys.

KEYBOARD TYPE

The Keyboards... button takes you to the Keyboard Type dialogue box which allows you to choose the layout of workstation and 3270 keyboards. Drop down lists are provided for each, 'PC:' denoting workstation and 'Terminal:' 3270.

To choose the workstation keyboard that most closely matches your own, click and drag the Keyboard Type dialogue box's title bar to move it where you can also see the PC keyboard on the Keyboard Setup dialogue box. Single mouse click on any workstation keyboard from the drop-down list and you will see the layout instantly displayed in the Keyboard Setup dialogue box in the background.

The one exception occurs when you select a very different workstation keyboard if changes have already been made to Reflection's default keyboard mapping. A second Keyboard Type dialogue box will be displayed with the prompt ‘The new local keyboard is substantially different, reset keyboard maps to defaults?’ The OK, Cancel, and Help buttons normally work as you might expect, but, under certain circumstances, Cancel does not cancel your 3270 keyboard selection.

Another Cancel problem is that most changes made in the Keyboard Type dialogue box are not reset when you hit the Cancel button in the Keyboard Setup dialogue box. To cancel those changes, too, you need to stop and restart Reflection without saving the Settings file. This shows not only how important it is to back up Reflection Settings files, but also that any Save of the Settings file should be a carefully considered decision, and never done automatically.

SINGLE KEY MAPPING

Once you have the Reflection workstation keyboard set to one that closely matches your own, you can start mapping any key assignments you would like to change. First, start Reflection by opening the Settings file you created during installation. Start-Documents is the easiest method, if you installed Reflection or used the Settings file in the last few days.

Alternatively, though not recommended, you can start Reflection directly:

- Start-Programs-Reflection-Host Mainframe.
- Push the Cancel button in the Enter Host Name or IP Address dialogue box.
- From the menu bar, File-Open.
- Double click on the Settings file.

From the menu bar, Setting-Keyboard..., then

- On the PC keyboard at the top of the Keyboard Setup dialogue box, single mouse click on the workstation key you want to map. The key will turn black to show that it is depressed.
- If the Remove button is no longer greyed-out, push it to delete the existing mapping for this workstation key.

- On the Terminal keyboard at the bottom, single-mouse-click the 3270 key you want to map to.
- Push the Map button.
- Click both keys so that they are no longer depressed.
- Repeat the process for each workstation key that you want to map.
- When you have finished, push the OK button.
- You may wish to test the keyboard mapping in a live mainframe terminal session.
- Save the Settings files by pushing the Save icon (a diskette symbol) on the Reflection toolbar, or selecting File-Save .rsf from the menu bar.

Note that the most common keyboard mapping mistake is to forget to hit the Map button before hitting the OK button.

MULTI-KEY MAPPING

However, keyboard mapping is not always this simple. To map a multi-key combination on either keyboard, first click the key(s) you would hold down (Shift, Alt, Ctrl), then the regular key. Note that all references in the following examples are to the Shift key, never the Caps Lock key.

To assign Alt+C to the cent sign (‘¢’):

- Click the Alt key on the top keyboard.
- Click the C key on the top keyboard.
- The bottom keyboard has been replaced by a list of commands, with Connect selected, indicating that Alt+C is currently mapped to the Reflection Connect function.
- Push the Remove button.
- Click the Terminal keyboard bullet beside by Action: in the middle of the Keyboard Setup dialogue box.
- The 3270 keyboard should be displayed again.

- Click the cent sign (‘¢’) on the bottom (3270) keyboard.
- Push the Map button.
- Click the Alt key on the top keyboard and the cent sign on the bottom keyboard.
- Now no keys will show as depressed.

MISSING 3270 KEYS

Some 3270 keys are nowhere to be found on the 3270 keyboard, even when you click on the Shift and Alt keys. These missing keys must be mapped using Commands.

To assign the workstation’s grey End key to the 3270 End of Field key:

- Click the End key on the top keyboard.
- Push the Remove button.
- Find Action: in the middle of the Keyboard Setup dialogue box.
- Click the bullet just to the left of Commands.
- A new line will appear just below, titled Commands.
- Click the Terminal keys bullet.
- Scroll through the list below, find rcIBMEndOfFieldKey, and click on it.
- Note that TransmitTerminalKey rcIBMEndOfFieldKey is now displayed in the field above.
- Push the Map button.
- Click the depressed End key on the top keyboard.
- Click the bullet between Action: and Terminal keyboard.

CURRENT MAPPING

In order to determine what a specific workstation or 3270 key is currently mapped to:

- Go to Setup-Keyboard... from the Reflection menu bar.
- Click the key on the relevant keyboard (top is workstation, bottom is 3270).
- The other keyboard will show the key(s) it is mapped to.

For 3270 keys for which several workstation key sequences are mapped, only one is shown. In the one case I tested, where one mapping was a Reflection default and the other was my own mapping, mine was displayed.

Clicking on a workstation key will show as a Command:

- If the 3270 key does not exist on the bottom keyboard, such as the End of Field key, which shows as the TransmitTerminalKey rcIBMEndOfFieldKey command.
- Or if the workstation key initiates a Reflection command, such as Disconnect, which shows simply as Disconnect.

Unfortunately, the process is one way – clicking on a command will not show what workstation key maps to it.

KEY MEANING AND LOCATION

There are a number of factors that determine how much the key names are abbreviated when displayed in the top and bottom keyboards. But there is no denying they can be difficult to recognize.

The top keyboard pictures the workstation keyboard, and abbreviation is not a big problem because you have your own keyboard right in front of you where you can see less abbreviated names. However, it’s unlikely that you’ll also have a 3270 keyboard or IBM manual close by for reference.

You can use Reflection Help to locate a specific 3270 key on the lower keyboard, but it also describes the function of many of the 3270 keys:

- From the Keyboard Setup dialogue box, push the Help button.
- Click Default 3270 keyboard mapping.
- Click on any underscored 3270 key name for a description of what it does.

- The second column, labelled Keystroke, shows the default workstation key(s).
- The third column, Environment, lists Host for 3270 keys and Reflection for product actions.

Access to this list is also provided through Reflection’s normal Help facility:

- From the Reflection menu bar, Help-Help Topics.
- Click on the Contents tab.
- Double click on Reference, then Default Mappings, then Default 3270 Keyboard Mapping.

To locate the position of a 3270 key on the lower keyboard, find the 3270 key in the Help listing, and then note the workstation key (combination) listed in the Keystroke column. Click the workstation key (combination) and the 3270 key will show as depressed on the lower keyboard.

Another approach, the only one for 3270 keys not on the lower keyboard or not mapped by default, is to switch the lower half to Commands, select Terminal Keys, and find the key in the list. Reflection has assigned easily recognizable names to each key and preceded it with ‘rcIBM’ – eg rcIBMEndOfFieldKey. You can then map the command to a workstation key. If you click the workstation key twice, to release it and re-depress it, the bottom keyboard will display the 3270 key if it’s on the keyboard.

WHAT’S THE POINT?

Most mainframe users and administrators never even think about keyboard mapping. But consider the productivity angle. The most productive mainframe environment is one where you:

- Know what can be done and what 3270 keys to hit to do it.
- Minimize the time required to find and enter the workstation key sequence to generate that 3270 key.
- Enter the key sequence automatically.

If this sounds like touch-typing, you’re right – it is. Even those who hunt and peck remember where to look on the keyboard for keys they use often. The goal of keyboard mapping is to make hitting frequently-used 3270 keys a natural, automatic process. For example, I use Erase EOF a lot. My finger automatically moves to the left of the ‘1’ key to hit it. All I think is, ‘erase those extra characters after my cursor’. The rest just happens naturally.

Armand Minet
(Canada)

NvDM receive procedure

The NvDM RECEIVE procedure presented here is based on NetView Distribution Manager and was developed to manage the conversion, from NvDM repository to traditional mainframe datasets, of software previously received from other NvDM nodes in a host-to-host configuration.

The procedure is designed to make receiving software simple by transparently creating, executing, and controlling the NvDM utilities in order to manage the software reception phase. It also supplies a historical log of all the main functions carried out.

The procedure is simple to use, and requires no knowledge of the rules or syntax of NetView Distribution Manager. For the software reception phase, users must simply select the datasets/files that they wish to convert from the Resource Repository to traditional (partitioned, sequential, etc) mainframe datasets.

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

NvDM RECEIVE PROCEDURE: REXX EXECs

C-list NRPC000

```
/* REXX */
/* NvDM Receive Procedure
   C-List NRPC000
   Called by ISPF user panel or user command
   Execute procedure for receive of datasets between more OS/390 hosts
   The functions are:
      - Check user; - Allocation environment for execution Receive
Procedure
      - If user enabled then run clist main NRPC00
*/
trace ?o
libpan = 'NVDM162.TOOLS.PANEL' ; libexe = 'NVDM162.TOOLS.CLIST'
         /*         Insert here the user of projects                */
if userid() ¬= NDMOPR1 & userid() ¬= NDMADM ,
       & userid() ¬= XXXXXXX & userid() ¬= YYYYYYY
   then do
        say time() ' >>>'
        say time() ' >>> *-----  Distribution Software  -----*'
        say time() ' >>>       "NvDM Receive Procedure"'
        say time() ' >>>'
        say time() ' >>>          USER NOT AUTHORIZED.'
        say time() ' >>>' ; say time() ' >>>'
        end
   else do
        address tso
        'ALTLIB ACT APPLICATION(EXEC) DA('''libexe''') UNCOND'
        "ISPEXEC LIBDEF ISPPLIB DATASET ID("''''libpan''''")"
        call nrpc010
        'ALTLIB DEACT APPLICATION(EXEC)'
        ISPEXEC LIBDEF ISPPLIB DATASET
        end
say time() ; say time() ' Distribution software ended.' ; say time()
exit
```

C-list NRPC010

```
/* REXX */
/* NvDM Receive Procedure
   C-List main NRPC010  -  Called by c-list NRPC000
   Execute procedure for receive of datasets between more OS/390 hosts
   The functions are:
   - verification if user is enable to use NvDM functions;
   - init environment to receive datasets/files from Resource
Repository;
   - display contents of the resource repository;
   - call others routine to generate, submit and control the utilities
NvDM; */
Trace ?o
/************************   Variables main   ***************************
              wjdsn = Dataset of work-jobs
              wjske = User library with job skeleton
              wtbl  = Tables of the datasets and of the projects
              wlogs = Dataset name of log NvDM Send Procedure
 **********************************************************************/
wjdsn = userid()||'.NVDM.RECEIVE'
dd=OUTTRAP(dd.)
   "DELETE '"wjdsn"'"
   "ALLOC DA('"wjdsn"') DIR(30) SPACE(10,1) DSORG(PO)" ,
   "RECFM(F,B) LRECL(80) BLKSIZE(6160) UNIT(WORKA) NEW CATALOG"
dd=OUTTRAP('OFF')
if rc > 0 then do
   do a=1 to dd.0
      say dd.a ; end
   say '>>>' ; say '>>>  Create of "'wjdsn'" failed the Return Code is
'rc
   say '>>>  To control.' ; say '>>>' ; return
end
```

```
wjske = userid()||'.LIB.CLIST' ; ww="rca=SYSDSN('''"wjske"''')"
INTERPRET ww
if rca ¬= 'OK' then do
   say '>>> Status of the "'wjske'" dataset'
   say '>>> is "'rca'". To control.' ; return
end
wjske1 = wjske||'(NDMSKEL)' ; ww="rcb=SYSDSN('''"wjske1"''')"
INTERPRET ww
if rcb ¬= 'OK' then do
   say '>>> Status of the member "'wjske1'"'
   say '>>> is "'rcb'". To control.' ; return
end
wtbl1=wjske||'(NRPT000)' ; ww="rcc=SYSDSN('''"wtbl1"''')"
INTERPRET ww
wtbl2 = 'NVDM162.NRPT000.TABLE' ; ww="rcd=SYSDSN('''"wtbl2"''')"
INTERPRET ww
if rcc¬='OK' & rcd¬='OK' then do
   say '>>>' ; say '>>> LIBRARY LIST not found. To control.'
   say '>>> Status of "'wtbl2'" is "'rcd'".' ; say '>>>' ; return
end
if rcc ¬= 'OK' then wtbl1='***' ; if rcd ¬= 'OK' then wtbl2 = '***'
wtbl = wtbl2||'+'||wtbl1 ; wlogs = 'NVDM162.TOOLS.LOG'
wlogs1 = wlogs||'1'  ; ww="rce=SYSDSN('''"wlogs1"''')"
INTERPRET ww
if rce ¬= 'OK' then do
   say '>>>' ; say '>>> Status of the dataset "'wlogs1'"'
   say '>>> is "'rce'". To control.' ; say '>>>' ; return
  end
wlogs2 = wlogs||'2' ; ww="rcf=SYSDSN('''"wlogs2"''')"
INTERPRET ww
if rcf ¬= 'OK' then do
   say '>>>' ; say '>>> Status of the dataset "'wlogs2'"'
   say '>>> is "'rcf'". To control.' ; say '>>>' ; return
  end
wsynm = 'NVDM162.TABLE.SYNONYM' ; ww="rcg=SYSDSN('''"wsynm"''')"
INTERPRET ww
if rcg ¬= 'OK' then do
   say '>>>' ; say '>>> Status of the dataset "'wsynm'"'
   say '>>> is "'rcg'". To control.'
   say '>>> The procedure continue without table.' ; wsynm='***'
end
pw = '*'
Do forever
Call NRPC020 wjske,wsynm,VE,pw,pwok
if result = 100 then signal DELD
                else pw = result
parse var pw pw '+++' progge
Call NRPC030 wtbl,wlogs,wjske,pw,progge
if result = 100 then return
End
DELD:
```

```
/************************************************************/
/* Delete job output and work datasets                     */
/************************************************************/
if userid() \= NDMADM then do
  deld1=userid()||'.NVDM.RECEIVE' ;
deld2=userid()||'.NVDM.RECEIVE.RESREP'
  ADDRESS TSO "DELETE '"deld1"'" ; ADDRESS TSO "DELETE '"deld2"'"
 end
return
```

C-list NRPC020

```
/* REXX */
/* NvDM Receive Procedure
   C-List NRPCØ2Ø   -   Called by c-list main NRPCØ1Ø
   Execute procedure for receive of datasets between more OS/39Ø hosts
   The functions are:
    - generate and display the contents of the Resource Repository;
    - select objects from Resource Repository to convert;
    - display contents of the resource repository;                 */
Trace ?o
ARG wjske,wsynm,VE,pw
if wjske = '' | wsynm = '' | VE = '' | pw = '' then Return 1ØØ
volta=Ø
Ancora:
if VE = Ø then DO FOREVER
   Call Displ1
   if proj = 'rc=8' then Return 1ØØ
   Call VediRR
   if io = -1 then Return 1ØØ
   if io ¬= Ø then leave
END
else Do
   if ripeti ¬= 1 & pw = '*' then do
      Call Pswd
      if pw = 'rc=8' then Return 1ØØ
   end
   proj = '*'
   Call VediRR
   if io = -1 then Return 1ØØ
   Call Displ2
   if io = Ø then Return 1ØØ
END
DO FOREVER
   ripeti = Ø
   Call Displ3
   if toc = Ø & tod = Ø then do
      say ; say time() " Nothing was selected to convert."
      Say ; ripeti = 1 ; leave
   end
   loads = 0
   do a=1 to toc
      sps = translate(word(tbtoc.a,1),' ','.')
      thirdqual = word(sps,3)
      if thirdqual = 'LOADS' then loads = loads + 1
   end
   if loads > 1 then do
                      DS = '' ; RFM = ''
                      say
  say time() 'It is possible to convert only LOADS for time. To
control.'
                      Say ; end
   else leave
END
if ripeti = 1 then signal Ancora
if tod > 0 then Call ConfDel
if toc = 0 & tod = 0 then do
   say ; say time() ' Nothing delete was confirmed.'
   Say ; ripeti = 1
   Signal Ancora
end
io = 0
if toc > 0 then do a = 1 to toc
  io=io+1 ; tabout.io='C' tbtoc.a
end
if tod>0 then do
  a=1 to tod
  io = io + 1 ; tabout.io = 'D' tbtod.a
end
/*****************************************************/
/* Prepare work list of datasets to convert          */
/*****************************************************/
if io > 1 then do
  y = 1 to io - 1
  w = y + 1 ; ele1 = translate(tabout.w,'   ','.()')
  ele1 = word(ele1,2)||'&'||word(ele1,4)
  ele2 = translate(tabout.y,'   ','.()')
  ele2 = word(ele2,2)||'&'||word(ele2,4)
  if ele1 < ele2 then do
                       comodo = tabout.y ; tabout.y = tabout.w
                       tabout.w = comodo ; y = 0
                      end
end
dsnwls = userid()||'.NVDM.RECEIVE(WORKLIST)'
xx=OUTTRAP(DD.)
   ADDRESS TSO "DELETE '"dsnwls"'"
xx=OUTTRAP(OFF)
tabout.0 = io
"ALLOC DA('"dsnwls"') F(WLS) SHR REUSE"
"EXECIO * DISKW WLS (STEM TABOUT. FINIS"
"FREE F(WLS)"
```

```
Return pw||'+++'||proj
Displ1:
 ADDRESS ISPEXEC 'DISPLAY PANEL(NRPP000)'
 UPPER proj pw
 if rc = 8 then proj = 'rc=8'
Return
Pswd:
 ADDRESS ISPEXEC 'DISPLAY PANEL(NRPP001)'
 UPPER pw
 if rc = 8 then pw = 'rc=8'
Return
VediRR:
 if volta = 0 then do
    Call NRPC025 wjske,pw
    if result > 99 then do
       if result = 666 then do
          say time() ; say time() ' Resource Repository is empty.'
          say time() ; end
       io = -1
       Return
    end
    volta = 1
 end
   dsnrls = userid()||'.NVDM.RECEIVE(RRREP)'
   "ALLOC DA('"dsnrls"') F(RLS) SHR REUSE"
   "EXECIO * DISKR RLS (STEM LL. FINIS"
   "FREE F(RLS)"
   io = 0
   do a=1 to ll.0
      UPPER ll.a
      pconfr = word(translate(word(ll.a,3),' ','.'),1)
      if proj = '*' then pconfr = proj
      if pconfr ¬= proj then iterate
      io = io + 1 ; tbd.io = ll.a
   end
   if io = 0 then do
   say time()
   say time() ' Nothing object into RR for the project "'proj'".'
   say time() ; end
Return
Displ3:
 tabella = '@@P1T1@@'
 ADDRESS ISPEXEC 'TBOPEN 'tabella' NOWRITE'
 if rc = 8 then,
    ADDRESS ISPEXEC 'TBCREATE 'tabella ,
     'NAMES(TDAT TTIM TNAM TBLK TLRE TVAR REAL DS RFM) NOWRITE'
 do a=1 to io
    TDAT = word(tbd.a,1) ; TTIM = word(tbd.a,2) ; dnam = word(tbd.a,3)
    TBLK = word(tbd.a,4) ; TLRE = word(tbd.a,5) ; TVAR = word(tbd.a,6)
    REAL = word(tbd.a,3) ; ldnam = length(dnam)
    if substr(dnam,ldnam-1,1) = '.' & datatype(right(dnam,1))='NUM'
```

```
then,
       dnam = left(dnam,lastpos('.',dnam)-1)
    if pos('.LOADS',dnam) ¬= Ø then ,
TNAM = proj||'.SHIPMS.LOADS.'||word(translate(dnam,' ','.'),4)
      else do
        ld = length(dnam) ; b = lastpos('.',dnam)
        if b = Ø then do
            say time() ' >>>'
            say time() ' >>> *------- NvDM Receive Procedure -------*'
            say time() ' >>>'
            say time() ' >>>         Dataset name invalid.'
            say time() ' >>>         Contact System Support.'
            say time() ' >>>' ; say time() ' >>>' ; leave
                      end
        TNAM = left(dnam,b-1)||'.'||right(dnam,ld-b)
      end
      if right(TNAM,1) = '.' then TNAM = left(TNAM,length(TNAM)-1)
      ADDRESS ISPEXEC 'TBADD 'tabella
   end
   ADDRESS ISPEXEC 'TBSORT 'tabella' FIELDS(TDAT,C,D,TTIM,C,D,TNAM,C,A)'
   do a=1 to io
    ADDRESS ISPEXEC 'TBSKIP 'tabella' NUMBER(1)'
    find.a = TNAM
   end
   finds = io
   ADDRESS ISPEXEC 'TBTOP 'tabella
   toc = Ø ; tod = Ø ; csrr = 1
   DO Forever
      ADDRESS ISPEXEC 'TBDISPL 'tabella,
      ' PANEL(NRPPØØ3)',
      ' CSRROW('csrr') AUTOSEL(NO)'
      IF RC = 8 then leave
      if zcmd ¬= '' then UPPER zcmd
      if zcmd = '' & DS ¬= '' & RFM ¬= '' then ms = S
      if zcmd = 'ALL' then do
         if tod = Ø then do
            do a=1 to io
  tbtoc.a = WORD(tbd.a,3) WORD(tbd.a,4) WORD(tbd.a,5) WORD(tbd.a,6) DS
RFM
            end
            toc = io ; tod = Ø ; leave
         end
         else do
            ind = Ø
            do a=1 to io
               ugu = Ø
               do b=1 to tod
                  if tbtod.b = word(tbd.a,3) then do
                     ugu = 1 ; leave
                  end
               end
               if ugu = 0 then do
                  ind = ind + 1
                  tbtoc.ind = WORD(tbd.a,3) WORD(tbd.io,4)
               end
            end
            toc = ind ; leave
         end
      end
      else do
         ss = ztdsels + 0
         if ss ¬= 0 then,
         do a=1 to ss
          if a ¬= 1 then ADDRESS ISPEXEC 'TBDISPL 'tabella'
POSITION(CRP)'
           UPPER ms
           if zcmd = '' & DS ¬= '' & RFM ¬= '' then ms = S
           if ms = 'C' | ms = 'S' then do
             toc = toc + 1 ; tbtoc.toc = REAL TBLK TLRE TVAR DS RFM
           end
           if ms = 'D' then do
             tod = tod + 1 ; tbtod.tod = REAL
           end
           f = 0
           do b=1 to finds
             if find.b = TNAM then iterate
             f = f + 1 ; find.f = find.b
           end
           finds = f
           ADDRESS ISPEXEC 'TBDELETE 'tabella
         end
         if words(zcmd) = 2 & word(zcmd,1) = 'F' then do
            do y = 1 to finds
              if pos(word(zcmd,2),find.y) ¬= 0 then leave
            end
            ADDRESS ISPEXEC 'TBSKIP 'tabella' NUMBER('y-1')'
         end
      end
   end
   ADDRESS ISPEXEC 'TBEND 'tabella
Return
Displ2:
   tprg.1 = word(translate(word(tbd.1,3),' ','.'),1)
   qris.1 = 1 ; pros = 1
   do a=2 to io
      cprg = word(translate(word(tbd.a,3),' ','.'),1) ; ugu = 0
      do b=1 to pros while ugu = 0
         if cprg = tprg.b then do
            qris.b = 1 + qris.b ; ugu = 1
         end
      end
      if ugu = 1 then iterate
      pros = pros + 1 ; tprg.pros = cprg ; qris.pros = 1
   end
   if wsynm ¬= '***' then do
      "ALLOC DA('"wsynm"') F(ACR) SHR REUSE"
      "EXECIO * DISKR ACR (STEM ACRS. FINIS"
      "FREE F(ACR)"
   end
   else acrs.0 = 0
   do a=1 to pros
      acrn.a = '   -' ; mssg.a = 'Synonymous not found'
      do b=1 to acrs.0
         UPPER acrs.b
         if word(acrs.b,1) = tprg.a & word(acrs.b,2) ¬= '' then do
            acrn.a = word(acrs.b,2) ; mssg.a = '' ; leave
         end
      end
   end
   tabella = '@@P1T3@@'
   ADDRESS ISPEXEC 'TBOPEN 'tabella' NOWRITE'
   if rc = 8 then,
      ADDRESS ISPEXEC 'TBCREATE 'tabella ,
      'NAMES(ACRO PRGT QRES MESS) NOWRITE'
   do a=1 to pros
      ACRO = acrn.a ; PRGT = tprg.a
      QRES = qris.a ; MESS = mssg.a
      ADDRESS ISPEXEC 'TBADD 'tabella
   end
   ADDRESS ISPEXEC 'TBSORT 'tabella' FIELDS(QRES,N,D)'
   ADDRESS ISPEXEC 'TBTOP 'tabella
   abort = 'N' ; messag = '' ; b15a = left('>',15,'>')
   b15b = left('<',15,'<') ; csrr = 1
   DO forever
      ADDRESS ISPEXEC 'TBDISPL 'tabella,
      ' PANEL(NRPP002)',
      ' CSRROW('csrr') AUTOSEL(NO)'
      if RC = 8 then do
         abort = 'Y' ; leave ; end
      ss = ztdsels+0
      if ss = 1 then leave
      if ss > 1 then messag = b15a 'Select only one Project' b15b
   END
   proj = PRGT
   ADDRESS ISPEXEC 'TBEND 'tabella
   if abort = 'Y' then do
      io = 0 ; Return ; end
   ii = 0
   do a=1 to io
      cprg = word(translate(word(tbd.a,3),' ','.'),1)
      if cprg ¬= proj then iterate
      ii = ii + 1 ; tbd.ii = tbd.a ; end
   io = ii
```

```
Return
ConfDel:
tabella = '@@P1T2@@'
ADDRESS ISPEXEC 'TBOPEN 'tabella' NOWRITE'
if rc = 8 then,
   ADDRESS ISPEXEC 'TBCREATE 'tabella ,
  'NAMES(DNAM REAL) NOWRITE'
do a=1 to tod
   tnam = strip(tbtod.a) ; REAL = strip(tbtod.a) ; ltnam = length(tnam)
   if substr(tnam,ltnam-1,1) = '.' & datatype(right(tnam,1)) = 'NUM'
then,
      tnam = left(tnam,lastpos('.',tnam)-1)
   tnamb = translate(tnam,' ','.')
   if pos('.LOADS',tnam) ¬= Ø then ,
   DNAM = word(tnamb,1)||'.SHIPMS.LOADS.'||word(tnamb,4)
   else do
     lt = length(tnam) ; b = lastpos('.',tnam)
     DNAM = left(tnam,b-1)||'('||right(tnam,lt-b)||')'
   end
   if right(DNAM,1) = '.' then DNAM = left(DNAM,length(DNAM)-1)
   ADDRESS ISPEXEC 'TBADD 'tabella
end
ADDRESS ISPEXEC 'TBSORT 'tabella' FIELDS(DNAM,C,A)'
ADDRESS ISPEXEC 'TBTOP 'tabella
csrr = 1 ; all = Ø ; todd = Ø
DO Forever
   ADDRESS ISPEXEC 'TBDISPL 'tabella,
   ' PANEL(NRPPØØ4)',
   ' CSRROW('csrr') AUTOSEL(NO)'
   IF RC = 8 then leave
   if zcmd ¬= '' then UPPER zcmd
   if zcmd = 'ALL' then do
     all = 1 ; leave
     end
   ss = ztdsels+Ø
   if ss ¬= Ø then do a=1 to ss
      if a ¬= 1 then ADDRESS ISPEXEC 'TBDISPL 'tabella' POSITION(CRP)'
      todd = todd + 1 ; tbtodd.todd = REAL
      ADDRESS ISPEXEC 'TBDELETE 'tabella
   end
end
ADDRESS ISPEXEC 'TBEND 'tabella
if all = Ø then do
  do a=1 to todd
    tbtod.a = tbtodd.a
    end
  tod = todd
end
Return
```

C-list NRPC025

```
/* REXX */
/* NvDM Receive Procedure
   C-List NRPCØ25    -  Called by c-list NRPCØ2Ø
   Execute procedure for receive of datasets between more OS/39Ø hosts
   The functions are:
    - generate and submit job to report of the Resource Repository;   */
Trace ?o
ARG wjske,pw
jc = 'R' ; jobendmsg = 'OUTPUT QUEUE' ; result = Ø ; jobtype = 'PRINT
DIR'
jname = userid()||jc ; dsnjsk=wjske||'(NDMSKEL)'
"ALLOC DA('"dsnjsk"') F(JSK) SHR REUSE"
"EXECIO * DISKR JSK (STEM JD. FINIS"
"FREE F(JSK)"
dsnrrd = userid()||'.NVDM.RECEIVE.RESREP'
dsnout = userid()||'.NVDM.RECEIVE.OUTPUT'
say time() ; say time() ' ... Create job 'jobtype ; say time()
jn = left(userid()||jc,9) ; jd.1 = '//'||jn||substr(jd.1,12)
do a=2 to jd.Ø
  pn = pos('*FUNCT*',jd.a)
  if pn ¬= Ø then leave
end
jd.a = left(jd.a,pn-1)'MAINTDAT'substr(jd.a,pn+7)
jd.a = left(jd.a,lastpos(',',jd.a)-1)||"'"
do a=2 to jd.Ø
  pn = pos('*OUT*',jd.a)
  if pn ¬= Ø then leave
end
dsn = dsnout ; jd.a = left(jd.a,pn-1)||dsn||substr(jd.a,pn+5)
jd.a = left(jd.a,8Ø) ; a = a + 2
jd.a = '//  UNIT=339Ø,DISP=(NEW,CATLG)' ; dpd = 'DSXPRINT DD'
do a=2 to jd.Ø
  if pos(dpd,jd.a) ¬= Ø then leave
end
dsn = dsnrrd ; jd.a = '//'dpd' DSN='dsn',' ; jd.a = left(jd.a,8Ø)
a1 = a + 1 ; sysin = jd.a1 ; a = a + 1 ; b = a - 3 ; jd.a = jd.b
a = a + 1 ; b = a - 3 ; jd.a = jd.b ; a = a + 1 ; jd.a = sysin
a = a + 1 ; jd.a = ' PRINT DIRECTORY TYPE=DATASET'
jd.a = left(jd.a,8Ø) ; a = a + 1 ; jd.a = ' END' ; jd.a = left(jd.a,8Ø)
a = a + 1 ; jd.a = left('/'||'*',8Ø) ; a = a + 1
jd.a = left('//BATCHPW DD *',8Ø) ; a = a + 1 ; jd.a =
left('<PASSWORD>',8Ø)
a = a + 1 ; jd.a = left('/'||'*',8Ø) ; jd.Ø = a
dsnrrj = userid()||'.NVDM.RECEIVE(JOBDIR)'
xx=OUTTRAP(DD.)
  ADDRESS TSO "DELETE '"dsnrrj"'"
xx=OUTTRAP(OFF)
"ALLOC DA('"dsnrrj"') F(RRJ) SHR REUSE"
"EXECIO * DISKW RRJ (STEM JD. FINIS"
"FREE F(RRJ)"
xx=OUTTRAP('jobm.')
  ADDRESS TSO "DELETE '"dsnrrd"'" ; ADDRESS TSO "DELETE '"dsnout"'"
xx=OUTTRAP('OFF')
/****************************************************************/
/* Call routine to submit job utility                           */
/****************************************************************/
Call NRPC035 dsnrrj,jname,jobtype,dsnout,pw
if result = 100 then Return result ; else wrkjn = result
result = 0
if left(wrkjn,3) = '+++' then do
  wrkjn = substr(wrkjn,4) ; ddd = 666 ; Signal Iesci
end
"ALLOC DA('"dsnrrd"') F(RRD) SHR REUSE"
"EXECIO * DISKR RRD (STEM RR. FINIS"
"FREE F(RRD)"
str.1 = 'Resource type' ; str.2 = 'Resource name' ; str.3 = 'Complete.'
str.4 = 'Creation date' ; str.5 = 'Creation time'
str.6 = 'Physical record length' ; str.7 = 'Logical record length'
str.8 = 'Variable record length' ; qstr = 8 ; e = 1 ; it = 0
do a=1 to rr.0
  if pos(str.e,rr.a)¬=0 then do
    if e = 1 then do
                   it = it + 1 ; lrr.it = '' ; end
    lrr.it = lrr.it word(rr.a,words(rr.a))
    e = e + 1
    if e > qstr then e = 1
   end
end
io = 0
do a=1 to it
   if words(lrr.a) ¬= qstr | word(lrr.a,3) = 'NO' then iterate
   io = io + 1
   lrr.io = word(lrr.a,4),
            word(lrr.a,5),
            word(lrr.a,2),
            word(lrr.a,6),
            word(lrr.a,7),
            word(lrr.a,8)
end
if io = 0 then do
  ddd = 666 ; Signal Iesci
end
dsnlst = userid()||'.NVDM.RECEIVE(RRREP)'
xx=OUTTRAP(DD.)
  ADDRESS TSO "DELETE '"dsnlst"'"
xx=OUTTRAP(OFF)
"ALLOC DA('"dsnlst"') F(LST) SHR"
"EXECIO "io" DISKW LST (STEM LRR. FINIS"
"FREE F(LST)"
ddd = 0
```

```
Iesci:
xx=OUTTRAP('dd.')
  ADDRESS TSO "CANCEL "wrkjn" PURGE"
xx=OUTTRAP('OFF')
Return ddd
```

C-list NRPC030

```
/* REXX */
/* NvDM Receive Procedure
   C-List NRPC030    -  Called by c-list main NRPC010
   Execute procedure for receive of datasets between more OS/390 hosts
   The functions are:
    - generate and submit job of Convert object from Resource
Repository; */
Trace ?o
ARG wtbl,wlogs,wjske,pw,progge
jobendmsg='OUTPUT QUEUE'
Call NRPC040 wtbl,progge
if result = ko then do
     say time() '>>>' ; say time() '>>> The "'progge'" project' ,
                     'is not managed in this environment.'
     say time() '>>> Null it will be converted.' ; say time() '>>>'
     return ; end
tipo = '' ; dst = '' ; p = 0 ; qc = 0 ; tmploads = '' ; libloads = ''
nomloads = '' ; tuttok = 0 ; dsnwkl =
userid()||'.NVDM.RECEIVE(WORKLIST)'
"ALLOC DA('"dsnwkl"') F(WKL) SHR REUSE"
"EXECIO * DISKR WKL (STEM WL. FINIS"
"FREE F(WKL)"
do a=1 to wl.0
      qc = qc + 1 ;  w6 = word(wl.a,6)
      if w6 = 'PO' then do
         w2 = word(wl.a,2) ; w2a = strip(translate(w2,'   ','.()'))
         w2b = words(w2a)-1 ; w2m = word(w2a,w2b)
         w2c = words(w2a)-2 ; w2t = word(w2a,w2c)
         w2lm = length(w2m)+3 ; w2ll = length(w2)
         w2lib = substr(w2,1,(w2ll-w2lm)) ; lb.a = w2lib ; me.a = w2m
         ti.a = w2t ; end
      else do
         w2 = word(wl.a,2) ; w2ll = length(w2)
         w2lib = substr(w2,1,(w2ll-2)) ; lb.a = w2lib ; me.a = ''
         ti.a = '' ; end
      nc.a = w2 ; bl.a = word(wl.a,3) ; rl.a = word(wl.a,4)
      vl.a = word(wl.a,5) ; do.a = word(wl.a,6) ; rf.a = word(wl.a,7)
   end
do b=1 to a-1
   if do.b ¬= 'PO' then call allocaps
               else if tipo ¬= ti.b then call allocapo
                                  else lb.b = lb.b||'.TEMP'
```

```
end
tabella='@@NST1@@'
ADDRESS ISPEXEC 'TBOPEN 'tabella' NOWRITE'
if rc=8 then,
   ADDRESS ISPEXEC 'TBCREATE 'tabella ,
    'NAMES(TDLB TDST) NOWRITE'
do j=1 to p
   TDLB = dlb.j ; TDST = dst.j
   ADDRESS ISPEXEC 'TBADD 'tabella
   ADDRESS ISPEXEC 'TBSKIP 'tabella' NUMBER('j-1')'
end
   ADDRESS ISPEXEC 'TBTOP 'tabella
   csrr = 0
   DO Forever
      ADDRESS ISPEXEC 'TBDISPL 'tabella,
      ' PANEL(NRPP005)',
      ' CSRROW('csrr') AUTOSEL(NO)'
      IF RC=8 then leave
   end
ADDRESS ISPEXEC 'TBDELETE 'tabella
ADDRESS ISPEXEC 'TBEND 'tabella
CreaJob:
/******************************************************************/
/* Generate Convert job                                           */
/******************************************************************/
dsnskj = wjske||'(NDMSKEL)'
"ALLOC DA('"dsnskj"') F(SKJ) SHR REUSE"
"EXECIO * DISKR SKJ (STEM JD. FINIS"
"FREE F(SKJ)"
jc = 'W' ; jname = userid()||jc
dsnout = userid()||'.NVDM.RECEIVE.OUTPUT' ; jtype = 'CONVERT'
say time() ; say time() ' ... Create job 'jtype
jn = left(jname,9) ; jd.1 = '//'||jn||substr(jd.1,12)
do a=2 to jd.0
   pn = pos('*FUNCT*',jd.a)
   if pn ¬= 0 then leave
end
jd.a = left(jd.a,pn-1)'MAINTDAT'substr(jd.a,pn+7)
jd.a = left(jd.a,lastpos(',',jd.a)-1)||"'"
do a=2 to jd.0
   pn = pos('*OUT*',jd.a)
   if pn ¬= 0 then leave
end
dsn = dsnout ; jd.a = left(jd.a,pn-1)||dsn||substr(jd.a,pn+5)
jd.a = left(jd.a,80) ; a = a + 2
jd.a = '//  UNIT=WORKA,DISP=(NEW,CATLG)'
do a=2 to jd.0
   if left(jd.a,7)='//SYSIN' then leave
end
sysin = jd.a
a = a - 1 ; nn = 0
```

```
if qc > 0 then do
   do b=1 to qc
      LL  =  POS('LOAD',LB.B)
      if me.b = '' & LL = 0 then do
                      nn = nn + 1 ; a = a + 1
                      jd.a = '//USER'||left(nn,4)||' DD DSN='||lb.b||','
                      if length(jd.a)+8<72 then jd.a = jd.a||'DISP=SHR'
                      else do
                            a = a + 1 ; jd.a='//         DISP=SHR'
                           end
                      end
      if me.b ¬= '' & LL = 0 then do
                      nn = nn + 1 ; dsnm = lb.b||'('me.b')'
                      a = a + 1
                      jd.a = '//USER'||left(nn,4)||' DD DSN='||dsnm||','
                      if length(jd.a)+8<72 then jd.a = jd.a||'DISP=SHR'
                      else do
                            a = a + 1 ; jd.a = '//         DISP=SHR'
                           end
                 end
      if LL > 0  then do
                      tmploads = userid()||'.NVDM.RECEIVE.LOADS.TEMP'
                      libloads = lb.b ; nomloads = nc.b
                      nn = nn + 1 ; a = a + 1
                      jd.a = '//USER'||left(nn,4)||' DD
DSN='||tmploads||','
                      a = a + 1
          jd.a = '//
UNIT='SYSUNIT',VOL=SER='SYSVOLUME',DISP=(NEW,CATLG),'
                     jd.a = '// UNIT=3390,DISP=(NEW,CATLG),' ; a = a + 1
                      jd.a = '//
DCB=(RECFM=VS,LRECL=32760,BLKSIZE='bl.b'),'
                      a = a + 1 ; jd.a = '// SPACE=(CYL,(20,5))'
                      end
   end
a = a + 1 ; jd.a = sysin ; cont = copies(' ',15)
if qc > 0 then do b=1 to qc
   id = nc.b
   a = a + 1 ; jd.a = '  CONVERT DATASET FCTLNAME=NDM,ORIGIN=DSX,'
   jd.a = left(left(jd.a,71)||'X',80) ; a = a + 1
   jd.a = cont||'ID='id',' ; jd.a = left(left(jd.a,71)||'X',80)
   a = a + 1 ; jd.a = cont||'DESTYPE=PS,OUTDD=USER'||b
   jd.a = left(jd.a,80)
end
a = a + 1 ; jd.a = left(' END',80)
a = a + 1 ; jd.a = left('/*',80) ; a = a + 1
jd.a = left('//BATCHPW DD *',80) ; a = a + 1
jd.a = left('<PASSWORD>',80) ; a = a + 1
jd.a = left('/*',80) ; jd.0 = a
dsnwkj = userid()||'.NVDM.RECEIVE(WORKJOB)'
```

```
xx=OUTTRAP(DD.)
   ADDRESS TSO "DELETE '"dsnwkj"'"
xx=OUTTRAP(OFF)
"ALLOC DA('"dsnwkj"') F(WKJ) SHR REUSE"
"EXECIO * DISKW WKJ (STEM JD. FINIS"
"FREE F(WKJ)"
xx=OUTTRAP('jobm.')
   ADDRESS TSO "DELETE '"dsnout"'"
   if tmploads ¬= '' then ADDRESS TSO "DELETE '"tmploads"'"
xx=OUTTRAP(OFF)
/********************************************************************/
/* Call routine to submit job                                       */
/* Management files type LOAD                                       */
/********************************************************************/
Call NRPCØ35 dsnwkj,jname,'CONVERT',dsnout,pw
if result = 1ØØ then signal Iesci
else wrkjn1 = result
if tmploads ¬= '' & libloads ¬= '' then do
   jobtype='IEBCOPYs' ; jc = 'I' ; jname = userid()||jc
   say time() ; say time() ' ... Create job 'jobtype
   ji.1 = left('//'||jname,11)||substr(jd.1,12)
   do a=2 to 6
      ji.a = jd.a
   end
   ji.7 =  '//STEP1   EXEC PGM=IEBCOPY'
   ji.8 =  '//IDD      DD DSN='tmploads',DISP=SHR'
   ji.9 =  '//ODD      DD DSN='libloads',DISP=SHR'
   ji.1Ø = '//SYSUT3   DD UNIT=339Ø,SPACE=(CYL,(1Ø,1))'
   ji.11 = '//SYSUT4   DD UNIT=339Ø,SPACE=(CYL,(1Ø,1))'
   dsnout = userid()||'.NVDM.RECEIVE.OUTCOPY'
   ji.12 = '//SYSPRINT DD DSN='dsnout',DISP=(NEW,CATLG),'
   ji.13 = '//'||'  DCB=(DSORG=PS,RECFM=FBA,BLKSIZE=121Ø,LRECL=121),'
   ji.14 = '//'||'  SPACE=(CYL,(1,1,Ø)),UNIT=339Ø'
   ji.15 = '//SYSUDUMP DD SYSOUT=*'
   ji.16 = '//SYSIN    DD *'
   ji.17 = '    COPY INDD=((IDD,R)),OUTDD=ODD'
   ji.18 = '/'||'*' ; ji.Ø = 18
   dsnieb = userid()||'.NVDM.RECEIVE(IEBCOPY)'
   xx=OUTTRAP(utt.)
      ADDRESS TSO "DELETE '"dsnieb"'"
   xx=OUTTRAP(OFF)
   "ALLOC DA('"dsnieb"') F(IEB) SHR REUSE"
   "EXECIO * DISKW IEB (STEM JI. FINIS"
   "FREE F(IEB)"
   xx=OUTTRAP('jobm.')
      ADDRESS TSO "DELETE '"dsnout"'"
   xx=OUTTRAP('OFF')
/********************************************************************/
/* Call routine to submit job for load type                         */
/* Control outcome job                                              */
/********************************************************************/
```

```
Call NRPCØ35 dsnieb,jname,jobtype,'***','***'
  if result = 1ØØ then signal Iesci ; else wrkjn2 = result
say time() ; say time() ' ... Control outcome 'jobtype ; say time()
"ALLOC DA('"dsnout"') F(IEBOUT) SHR"
"EXECIO * DISKR IEBOUT (STEM OJ. FINIS"
"FREE F(IEBOUT)"
if oj.Ø = Ø | oj.Ø = 'OJ.Ø' then do
   say
   say ">>>>>>>>  The job "jobtype" is ended with errors. To control."
   Say ; Signal Iesci ; end
tms=Ø ; iebrc = '***ABEND***'
do u=1 to OJ.Ø
  if word(oj.u,1) = 'IEB154I' then do
    tms = tms + 1 ; mms.tms = word(oj.u,2)
  end
  if word(oj.u,1) = 'IEB147I' then do
    iebrc = word(oj.u,6)
      end
   end
  if iebrc ¬= 'Ø' & iebrc ¬= '4' then do
  say '>>>>>>' ; say '>>>>>> The Return Code of 'jobtype' job is
"'iebrc'".'
  say '>>>>>> Verify the dataset "'dsnout'".' ; say '>>>>>>'
  Signal Iesci ;  end
end
tuttok = 1
/*************************************************************/
/* Update log activities dataset                             */
/*************************************************************/
Call Writlog
say time() ; say time() " The Receive function has been ended
correctly."
say time() " Read contents of the Resource Repository." ; say time()
Iesci:
if tuttok = 1 then result = Ø ; else result = 1ØØ
Return result
Writlog:
log1 = wlogs||'1' ; log2 = wlogs||'2'
"ALLOC F(LLOG1) DA('"log1"') SHR"
"EXECIO 1 DISKR LLOG1 (STEM LR FINIS"
if rc ¬= Ø then do
                 w1 = Ø ; frc.1 = 1
                end
           else do
                 frc.1 = Ø ; w1 = word(LR1,1)
                 if datatype(w1) ¬= 'NUM' | length(w1) ¬= 11 then w1 = Ø
                end
"FREE F(LLOG1)"
"ALLOC F(LLOG2) DA('"log2"') SHR"
"EXECIO 1 DISKR LLOG2 (STEM LR FINIS"
if rc ¬= Ø then do
```

```
                 w2 = Ø ; frc.2 = 1 ; end
           else do
                 frc.2 = Ø ; w2 = word(LR1,1)
                 if datatype(w2) ¬= 'NUM' | length(w2) ¬= 11 then w2 = Ø
                end
"FREE F(LLOG2)"
if w2 > w1 then log = 2 ; else log = 1
namelog = wlogs||log
if frc.log = 1 then Call Primi2
righe.1 = 'User: 'userid()'  -  Date: 'date('E')'  -  Time: 'time()
rr = 1
if qc > Ø then do a = 1 to qc
   q3l = word(strip(translate(lb.a,'   ','.()')),3)
   if pos('LOAD',q3l) ¬= Ø then iterate
   if q3l = old then iterate
   rr = rr + 1 ; righe.rr = '   Received into dataset "'lb.a'" :'
   old = q3l
   do b=1 to qc
      ricb = strip(translate(nc.b,'   ','.()'))
      q3d = word(ricb,3)
      if q3l = q3d then do
         rr = rr + 1
         if do.b ¬= 'PO' then
                        riga = '      '||word(ricb,1)||'.'||,
                            word(ricb,2)||'.'||q3d||'.'||word(ricb,4)
                         else
                         riga = '      '||word(ricb,1)||'.'||,
                          word(ricb,2)||'.'||q3d||'('||word(ricb,4)||')'
         righe.rr = riga
      end
   end
end
if libloads ¬= '' then do
   bload = strip(translate(nomloads,'   ','.()'))
   rr = rr + 1 ; righe.rr = '   Received into dataset "'libloads'" :'
   rr = rr + 1
   righe.rr = '      '||word(bload,1)||'.SHIPMS.LOADS.'||word(bload,4)
   do b=1 to tms
      rr = rr + 1 ; if b=1 then righe.rr = '         Members: '||mms.b
             else righe.rr = '                 '||mms.b
   end
end
aa = rr + 1 ; righe.aa = left('-',7Ø,'-')
ww="xx=LISTDSI('''"namelog"''')"
interpret ww
if SYSEXTENTS=16 & SYSUSED=SYSALLOC then do
   oldlog = namelog
   if log = 1 then log = 2 ; else log = 1
   namelog = wlogs||log
   say time() ; say time() ' ATTENTION:  dataset of log "'oldlog'" is
full.'
```

```
   say time() '             Switch to log "'namelog'".' ; say time()
   Call Primi2
end
do xx=1 to 500
   "ALLOC F(LLOGH) DA('"namelog"') MOD"
   if rc = 0 then leave
end
"EXECIO "aa" DISKW LLOGH (STEM RIGHE. FINIS"
zz = rc
"FREE F(LLOGH)"
if zz ¬= 0 then do
   say '>>>>>' ; say '>>>>>  Write error for log dataset "'namelog'".'
   say '>>>>>  Return Code is 'zz'. To control.' ; say '>>>>>'
end
Return
Primi2:
DaTi = left(date('B'),6,'0')||left(time('S'),5,'0')
rige1 = DaTi||' - Init '||date('E')' time 'time()'.'
rige2 = left('-',70,'-')
"ALLOC F(LLOGN) DA('"namelog"') SHR"
"EXECIO 2 DISKW LLOGN (STEM RIGE FINIS"
zz = rc
"FREE F(LLOGN)"
if zz ¬= 0 then do
  say '>>>>>' ; say '>>>>> Init error for log dataset "'namelog'".'
  say '>>>>> Return Code is 'zz'. To control.'
  say '>>>>>' ; signal Iesci
  end
Return
allocapo:
tipo = ti.b ; p1 = POS('.',LB.B) ; p2 = p1 - 1 ; alias =
SUBSTR(LB.B,1,P2)
call VAIPO
return
VAIPO:
lb.b = lb.b||'.TEMP' ; lenrf = length(rf.b)
if lenrf = 3 then do
  rf1 = delstr(rf.b,2,2) ; rf2 = substr(rf.b,2,1)
  rf3 = delstr(rf.b,1,2) ; recfm = rf1||','rf2||','rf3
  end
 else do
  rf1 = delstr(rf.b,2,1) ; rf2 = delstr(rf.b,1,1)
  recfm = rf1||','rf2
  end
DD = POS('LOAD',LB.B)
 if DD > 0 then do
   xx=OUTTRAP(dd.)
      ADDRESS TSO "DELETE '"lb.b"'"
      "ALLOC DA('"lb.b"') F(DESTPO) DIR(50) SPACE(10,10) CYL" ,
      "RECFM(U) LRECL(0) BLKSIZE("bl.b") DSORG(PO) UNIT(WORKA) NEW
CATALOG"
```

```
   xx=OUTTRAP(OFF)
   p = p + 1
   end
   else do
    xx=OUTTRAP(dd.)
       ADDRESS TSO "DELETE '"lb.b"'"
       "ALLOC DA('"lb.b"') F(DESTPO) DIR(50) SPACE(10,10) CYL" ,
       "RECFM("recfm") LRECL("rl.b") BLKSIZE("bl.b") DSORG(PO) " ,
       "UNIT(WORKA) NEW CATALOG"
    xx=OUTTRAP(OFF)
    p = p + 1
    end
dst.p = 'Partitioned'
if rc > 0 then do
   do k=1 to dd.0
      say dd.k
   end
   say '>>>>>' ; say '>>>>> Allocation error for "'lb.b'".'
   say '>>>>> Return Code is 'rc'. To control.' ; say '>>>>>'
   end
else
   dlb.p = lb.b
ADDRESS TSO "FREE F(DESTPO)"
RETURN
allocaps:
p1 =  POS('.',LB.B) ; p2 = p1 - 1 ; alias = SUBSTR(LB.B,1,P2)
call VAIPS
return
VAIPS:
lb.b = lb.b||'.TEMP' ; lenrf = length(rf.b)
if lenrf = 3 then do
  rf1 = delstr(rf.b,2,2) ; rf2 = substr(rf.b,2,1)
  rf3 = delstr(rf.b,1,2) ; recfm = rf1||','rf2||','rf3
  end
 else do
  rf1 = delstr(rf.b,2,1) ; rf2 = delstr(rf.b,1,1)
  recfm = rf1||','rf2
  end
DD = POS('LOAD',LB.B)
 if DD > 0 then do
   xx=OUTTRAP(dd.)
    ADDRESS TSO "DELETE '"lb.b"'"
    "ALLOC DA('"lb.b"') F(DESTPS) SPACE(10,10) CYL" ,
    "RECFM(U) LRECL(0) BLKSIZE("bl.b") DSORG(PS) UNIT(WORKA) NEW
CATALOG"
   xx=OUTTRAP(OFF)
   p = p + 1
   end
   else do
      xx=OUTTRAP(dd.)
       ADDRESS TSO "DELETE '"lb.b"'"
       "ALLOC DA('"lb.b"') F(DESTPS) SPACE(10,10) CYL RECFM("recfm")" ,
       "LRECL("rl.b") BLKSIZE("bl.b") DSORG(PS) UNIT(WORKA) NEW CATALOG"
      xx=OUTTRAP(OFF)
      p = p + 1
      end
 dst.p = 'Sequential'
if rc > 0 then do
   do k=1 to dd.0
      say dd.k
   end
   say '>>>>>' ; say '>>>>> Allocation error for "'lb.b'".'
   say '>>>>> Return Code is 'rc'. To control.' ; say '>>>>>'
   end
else
   dlb.p = lb.b
ADDRESS TSO "FREE F(DESTPS)"
RETURN
```

C-list NRPC035

```
/* REXX */
/* NvDM Receive Procedure
   C-List NRPC035    -  Called by c-list NRPC030, NRPC025
   Execute NvDM utilities
   The functions are: - submit jobs of utilities;
        - verification the corrected execution of the utilities; */
Trace ?o
ARG tosub,jn,jtype,jout,pw
jobendmsg = 'OUTPUT QUEUE'
if pw ¬= '***' then do
  "ALLOC DA('"tosub"') F(SUB) SHR REUSE"
  "EXECIO * DISKR SUB (STEM LL. FINIS"
  ll0m1 = ll.0 - 1 ; ll.ll0m1 = pw
  "EXECIO * DISKW SUB (STEM LL. FINIS"
  "FREE F(SUB)"
  end
say time() ; say time() ' ... Submit job 'jtype
xx=OUTTRAP('jobm.')
   ADDRESS TSO "SUBMIT '"tosub"'"
xx=OUTTRAP('OFF')
if pw ¬= '***' then do
  "ALLOC DA('"tosub"') F(SUB) SHR REUSE"
  "EXECIO * DISKR SUB (STEM LL. FINIS"
  ll0m1 = ll.0 - 1 ; ll.ll0m1 = ' <PASSWORD>'
  "EXECIO * DISKW SUB (STEM LL. FINIS"
  "FREE F(SUB)"
  end
do a=1 to jobm.0
   if pos(jn,jobm.a) ¬= 0 & pos('SUBMITTED',jobm.a) ¬= 0 then do
       sp = pos('(',jobm.a) ; lu = pos(')',jobm.a)-sp+1
       jid = substr(jobm.a,sp,lu)
      end
end
wrkjn = jn||jid ; ended=0
do until ended=1
   xx=OUTTRAP('jobx.')
      ADDRESS TSO "STATUS "wrkjn
   xx=OUTTRAP('OFF')
   do z = 1 to jobx.0
     if pos(jid,jobx.z) ¬= 0 & pos(jobendmsg,jobx.z) ¬= 0 then ended = 1
   end
end
if jout = '***' then Return wrkjn
say time() ; say time() ' ... Control outcome 'jtype ; say time()
ww="xx=SYSDSN('''"jout"''')"
INTERPRET ww
if xx ¬= 'OK' then do
  say '>>>>>' ; say '>>>>> Status output dataset "'jout'"'
  say '>>>>> is "'xx'". To control. End Procedure.' ; say '>>>>>'
  Return 100
  end
"ALLOC DA('"jout"') F(DD1) SHR REUSE"
"EXECIO * DISKR DD1 (STEM REC. FINIS"
"FREE F(DD1)"
/*****************************************************************/
/* Routine of search error messages into NvDM utility output     */
/* dataset. It selects only error records.                       */
/*****************************************************************/
E = 0 ; PD = '' ; nores = ''
DO I = 1 TO REC.0
  A = POS('NDM',REC.I)
  if A = 0 then Iterate
  B = substr(REC.I,A+3,4)
  if DATATYPE(B) ¬= 'NUM' then Iterate
  C = substr(REC.I,A+7,1)
  if C ¬= 'E' & c ¬= 'T' then Iterate
  D = substr(REC.I,A,8)
  if D = 'NDM3324E' | D = 'NDM5116E' then Iterate
  if D = 'NDM2647E' then do
                          pd = d ; Iterate
                         end
  if D = 'NDM2605E' & pd = 'NDM2647E' then do
                                           nores='+++' ; Iterate
                                           end
/*****************************************************************/
/* Message Errors found.                                         */
/* Delete job output and work datasets                           */
/*****************************************************************/
  E = E + 1 ; ER.E = strip(rec.i)
END
```

```
IF E ¬= Ø THEN do
   say '>>>>>'
   say '>>>>> The job 'jtype' has not been concluded correctly.'
   say '>>>>> The errors messages found are:'
   do a=1 to e
     say er.a
   end
   say '>>>>> To control.' ; say '>>>>>'
if userid() \= NDMADM then do
    deld1=userid()||'.NVDM.RECEIVE' ;
deld2=userid()||'.NVDM.RECEIVE.RESREP'
    ADDRESS TSO "DELETE '"deld1"'" ; ADDRESS TSO "DELETE '"deld2"'"
    say time() ; say time() 'Procedure ended.' ; say time()
   end
   E = 1ØØ
end
if e ¬= 1ØØ then e = nores||wrkjn
Return E
```

C-list NRPC040

```
/* REXX */
/* NvDM Receive Procedure
   C-List NRPCØ4Ø    -   Called by c-list main NRPCØ1Ø
   Execute procedure for receive of datasets between more OS/39Ø hosts
   The functions are:
    - it verifies the existence of the project to convert and if
      it is managed from that node NDM;                               */
Trace ?o
ARG wtbl,progge
parse var wtbl nlib1 '+' nlib2
"ALLOC DA('"nlib1"') F(LLNS) SHR REUSE"
"EXECIO * DISKR LLNS (STEM LL. FINIS"
"FREE F(LLNS)"
do w=1 to ll.Ø
   alias = substr(ll.w,1,8)
   if alias = progge then do
                           result = ok ; leave
                          end
                     else result = ko
end
return result
```

NVDM RECEIVE PROCEDURE: SAMPLE TABLE

```
*  NvDM RECEIVE Procedure
*  Table NRPTØØØ to management of the project and the relative datasets/
files:
 *   Insert here High level qualified of projects for NvDM Receive
Procedure
ProjA
ProjB
ProjC
Projx
NDMADM
```

NvDM RECEIVE PROCEDURE: ISPF PANELS

Panel NRPP000

```
)ATTR
/*
*/
/* NvDM Receive Procedure -  Panel NRPPØØØ
*/
 @ TYPE(TEXT) INTENS(LOW)
 $ TYPE(INPUT) INTENS(NON)
)BODY EXPAND(//)
%=/=/= NvDM ***  RECEIVE SOFTWARE =/=/=
+
+COMMAND ===>_ZCMD
+
+
+
+
+
+  Insert%high level qualified of the project _PROJ    +,
+
+  insert%NvDM password $PW      +and Press ENTER.+
+
+
+
+
+
+
+
+
+
+  %PF1+= HELP              %PF3+= END
)INIT
  .CURSOR = PROJ
  &ZCMD = ' '
  &PROJ = ' '
  &PW   = &PW
)REINIT
  &ZCMD = ' '
  &PROJ = &PROJ
  &PW   = &PW
  .ATTRCHAR(@) = 'INTENS(HIGH)'
)PROC
 VER (&PROJ,NONBLANK)
 VER (&PW,NONBLANK)
)END
```

Panel NRPP001

```
)ATTR
/*
*/
/* NvDM Receive Procedure -  Panel NRPPØØ1
*/
 @ TYPE(TEXT) INTENS(LOW)
 $ TYPE(INPUT) INTENS(NON)
)BODY EXPAND(//)
%=/=/= NvDM ***  RECEIVE SOFTWARE =/=/=
+
+
+
+
+
+
+
+
+                  Insert the%NvDM password+===>$PW      +
+
+
+                  or Press %PF3+to return TSO
)INIT
 .CURSOR = PW
 &PW   = &Z
)REINIT
 &PW   = &PW
)PROC
 VER (&PW,NONBLANK)
)END
```

Panel NRPP002

```
)ATTR
/*
*/
/* NvDM Receive Procedure -   Panel NRPPØØ2
*/
 @ TYPE(OUTPUT) INTENS(LOW)
 ¬ TYPE(OUTPUT) INTENS(HIGH)
 _ TYPE(INPUT)  CAPS(ON)
 $ TYPE(INPUT)  CAPS(ON) PADC('_')
```

```
)BODY EXPAND (##)
%=#=#= NvDM ***  RECEIVE SOFTWARE =#=#=
+COMMAND ===>_ZCMD
+
%-#-#- PROJECTS FOUND INTO RR -#-#-
+
%OPTIONS: +Insert%S+near project selected
          +Press%PF3+to return TSO
+
          %Attention:+Select%only one +project
+
¬MESSAG
+
%     S    SYNONYMOUS         PROJ-ID         RESOURCE
)MODEL
     $Z+  @ACRO            + @PRGT      +    @QRES +   @MESS
+
)INIT
  .ZVARS = '(MS)'
  .CURSOR = MS
  &MS = &Z
  &ZCMD = &Z
)REINIT
  &MS = &Z
  &ZCMD = &Z
  REFRESH(MS)
)PROC
 IF (&ZTDSELS ¬= 0000)
    VER(&MS,LIST,S)
)END
```

Panel NRPP003

```
)ATTR
/*
*/
/* NvDM Receive Procedure  -  Panel NRPP003
*/
 @ TYPE(OUTPUT) INTENS(LOW)
 _ TYPE(INPUT)  CAPS(ON)
 $ TYPE(INPUT)  CAPS(ON) PADC('_')
)BODY EXPAND (##)
%=#=#= NvDM ***  RECEIVE SOFTWARE =#=#=
+COMMAND ===>_ZCMD
%-#-#- RESOURCE LIST for PROJECT -#-#-
%OPTIONS: +Insert%C+or%S+near objects to convert from Resource
Repository
          +Insert%DSORG(PO o PS) and RECFM(FB,FBA,U,VB,VBA,ect.)
          +Press%PF3+to continue Convert procedure.
+
%        INTO RR
%S   DATE      TIME    RESOURCE NAME
DSORG RECFM
)MODEL
$Z+ @TDAT     @TTIM   @TNAM                                        $DS
$RFM
)INIT
  .HELP = NRPPØØ3H
  .CURSOR = &ZCMD
  &MS = &Z
  &DS = &DS
  &RFM = &RFM
  &ZCMD = &Z
  .ZVARS='(MS)'
)REINIT
  &DS = &DS
  &RFM = &RFM
  &MS = &Z
  REFRESH(MS)
)PROC
 IF (&ZTDSELS ¬= ØØØØ)
    VER(&MS,LIST,S,C)
    VER(&DS,NB,LIST,PO,PS)
    VER(&RFM,NB,LIST,VB,VBA,FB,FBA,U)
)END
```

Panel NRPP003H

```
)BODY EXPAND (//)
+
%=/=/= NvDM ***  RECEIVE SOFTWARE PROCEDURE =/=/=
+
+ This is the list of objects received from other NvDM nodes. This list
+ is relative to identification project chosen by user from previous
+ panel.
+ Possible options are:
+
% S+or%C+ On the line object to convert from NvDM Resource Repository
+ into user dataset.
+
% DSORG+and%RECFM+On the line object to convert; these parameters must
+                 be specified.
+                 Specify the correct Data Set Organization and the
+                 correct Record Format to obtain a valid conversion.
+
+
+
+  Press PF3 to continue with the function requested.
)END
```

Panel NRPP004

```
)ATTR
/*
*/
/* NvDM Receive Procedure -  Panel NRPP004
*/
 @ TYPE(OUTPUT) INTENS(LOW)
 _ TYPE(INPUT)  CAPS(ON)
 $ TYPE(INPUT)  CAPS(ON) PADC('_')
)BODY EXPAND (##)
%=#=#= NvDM ***  RECEIVE SOFTWARE =#=#=
+COMMAND ===>_ZCMD
+
%-#-#- CONFIRM CANCEL RESOURCES From Resource Repository -#-#-
%OPTIONS: +Insert%D+near objects to delete
+          Insert%ALL+on command line to confirm delete for%all+objects
+
+Press%PF3+to continue procedure
+
%  D     NAME RESOURCE
)MODEL
+ $Z+   @DNAM
+
)INIT
  .CURSOR = &ZCMD
  &MS = &Z
  &ZCMD = &Z
  .ZVARS='(XS)'
)REINIT
  &MS = &Z
  REFRESH(XS)
)PROC
 IF (&ZTDSELS ¬= 0000)
    VER(&XS,LIST,D)
)END
```

Panel NRPP005

```
)ATTR
/*
*/
/* NvDM Receive Procedure  - Panel NRPP005
*/
 @ TYPE(OUTPUT) INTENS(LOW) COLOR(GREEN)
 $ TYPE(OUTPUT) INTENS(LOW) COLOR(YELLOW)
)BODY EXPAND (##)
%=#=#= NvDM ***  RECEIVE SOFTWARE =#=#=
+COMMAND ===>_ZCMD
%-#-#- RECEIVE LIBRARY LIST for PROJECT -#-#-
+
+    Press PF3 to continue.
+
%    The selected resources will be converted into following datasets:
+
)MODEL
+   @TDLB                                +- $TDST
)INIT
  .HELP = NRPPØØ5H
  .CURSOR = &ZCMD
)REINIT
)PROC
)END
```

Panel NRPP005H

```
)BODY EXPAND (//)
+
%=/=/= NvDM ***  RECEIVE SOFTWARE PROCEDURE =/=/=
+
+
+
+
+ This is the datasets list allocated by "Receive Procedure" in dynamic
+ mode. The datasets will be used to convert the objects selected
+ previous. The datasets are allocated on "work disk" with a retention
+ period of one day.
+
+
+
+ Press PF3 to obtain the conversion.
)END
```

Espedito Morvillo
(Italy)

3270 datastream basics – correction

In the *3270 datastream basics* article published in the last issue, Figure 1, on page 24, was reproduced incorrectly. The correct Figure is shown below.

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

Figure 1: Attribute bits to byte value

George Walker
(Canada)

Information point – reviews

The topic for this issue is networking dictionaries and glossaries on the Internet. An obvious place to start is a dictionary portal:

http://www.yourdictionary.com

Portals attempt to give the visitor an organized view of the chaotic world of Internet Web sites, and yourDictionary.com (YDC) claims to have the widest and deepest set of dictionaries on the Web (more than 1,500 dictionaries representing more than 230 languages).

For more information, click the *About* link at the top of the home page. Further down the About page, you will read about the Advisory Council of Experts. They ensure the quality of sites being added, but slow down the addition of new sites. And a site flagged as new, *Microsoft computer dictionaries*, has a bad link.

The end result is a very incomplete list of computing dictionaries and glossaries. But some of those that are listed are interesting – for example, there is one each on spam and hacker terms. To get there from the home page, click on *Speciality* in the light brown bar near the top of the page. Scroll down through the alphabetical list until you find *Computing*, and click on it. Alternatively, a right sidebar on the home page usually has Computing in the list of Top Speciality Dictionaries in English. Or you can get there directly:

http://www.yourdictionary.com/diction4.html#computing

Given the wide breadth of coverage, it comes as no surprise that the closest it comes to a networking-specific dictionary is one by the Storage Networking Industry Association. But many of the other 18 dictionaries listed in the computing category include networking terms.

It would be easy to walk away at this point, thinking that the site has a lot of potential but realizing most of it is only outside of computing. But the left sidebar on the home page includes a link to *Multi-lingual Dictionaries*. Here, you'll find more than 30 dictionaries listed alphabetically, making it easy to overlook the following:

- *ITU Telecommunication Terminology Database* in English, French, Russian, and Spanish.
- *Multilingual Computer Dictionary* in English, Estonian, French, and Finnish.
- *NETGLOS Multilingual Glossary of Internet Terms* in Chinese, Dutch, English, French, German, Greek, Norwegian, Serbo-Croatian, and Spanish.

Note that the list of languages above is based on actually looking at each dictionary, and may contradict what yourdictionary.com indicates.

If your needs go beyond English, there are more computing dictionaries. From the home page, click on *Language Dictionaries* at the top of the left sidebar. Then click on the language of interest from the alphabetic list of more than 200. If the section begins with the heading General Dictionaries, scroll down for a Speciality Dictionaries section. Scan through it looking for Computing dictionaries.

Finally, the most convenient feature of the site is the on-line dictionary in the middle of the home page. Type in a word and click the GO button. It's powered by Merriam-Webster's Collegiate Dictionary, and includes many common computing terms.

LOOKING ELSEWHERE

The light brown bar near the top of the home page also includes an *Other Indexes* link to a number of other dictionary lists on the Internet, including Yahoo's. But there are better ways to find dictionaries and glossaries covering specific areas of computing, namely using search engines and searching vendor sites.

To pick a vendor, choose the largest in the area you're most interested in. For example, you could go to the Cisco home page at http://www.cisco.com, type glossary in the purple *Search* box in the upper left corner of the home page, and push the Go button. Even though you get 28,800 results, the first 10 include several that might be of interest:

- Internetworking Terms and Acronyms
- ISDN Glossary

- xDSL Glossary
- IP Telephony Network Glossary
- Glossary of Optical Terms.

Even though the tenth item is labelled simply ‘Glossary’, the short snippet of contents shows the first term as ‘3270 datastream’, making it certainly worth a look too.

IBM

More often than not, IBM will be the vendor of choice. At:

http://www.ibm.com/ibm/terminology/

you’ll find the famous *IBM Dictionary of Computing,* modestly listed as the *Glossary of Computing Terms*. There are also glossaries for AS/400, AIX, DB2, CICS, DCE, and Tivoli.

Many IBM manuals also have glossaries of relevant terms in the back, and these may be just the subject-specific dictionaries you’re looking for. From:

http://www.s390.ibm.com/os390/bkserv

begin by selecting an operating system. One approach is to *List all bookshelves,* click *Search* text, then type *glossary* in the search box and hit the *Search* button.

If you’re not sure which bookshelf contains manuals relevant to your area of interest, begin again just after you selected an operating system. Beside *Search,* click on *Text,* then type *glossary,* and hit the *Search* button. Using your browser’s search function, find manual titles of interest, eg – in IE, Edit-Find for *SNA*. Then click on each manual to see if it has a glossary listed (or mentioned).

Two things to note:

- In VM and VSE, this process may have to be repeated, as manuals are split across multiple (CD-ROM) Disk images.
- In OS/390, be sure you’re working with the elements and features bookshelf.

SEARCH ENGINES

With search engines, pick a word or phrase you’d expect to find in the dictionary you’re looking for, but that’s unlikely to be used in other contexts, and search for it on the same Web page as the word ‘glossary’ or ‘dictionary’. For example, from

http://www.alltheweb.com/cgi-bin/advsearch

change *Search for* to *any of the words*, type *glossary dictionary* into the field just to the right. Then under *Word Filters*, just to the right of *Must include*, type your word or phrase, and push the *FAST Search* button.

A final note: none of these are precise processes, and you’ll get a lot of false hits. But if less than 10% to 20% look vaguely interesting, you may wish to rethink your search terms and try again.

Jon E Pearkins
(Canada)

Contributing to *TCP/SNA Update*

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

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

If you have ever experienced any difficulties, or made an interesting discovery, you could receive a cash payment, a free subscription to any of our *Updates*, or a credit against any of Xephon's wide range of products and services, simply by telling us all about it. For a copy of our *Notes for Contributors*, which explains the terms and conditions under which we publish articles, please go to www.xephon.com/contnote.html.

TCP/SNA news

Landmark Systems has announced general availability of its TMON for TCP/IP, for monitoring and managing the availability of network devices and overall network performance.

For further information, contact:
Landmark Systems, 8000 Towers Crescent Drive, Vienna, VA 22182, USA.
Tel: (703) 902 8000.
Software Products, Systems House, 27 Spon Street, Coventry, CV1 3BA, UK.
Tel: (1684) 833700.
URL: http://www.landmark.com

* * *

IBM has announced Version 2.6 of its Tivoli NetView Performance Monitor. It has also announced Version 1.2 of NPM for TCP/IP, for managing TCP/IP OS/390 systems and the connected TCP/IP network.

For further information, contact:
Tivoli Systems, 9442 Capital of Texas Highway North, Arboretum, Austin, TX 78759, USA.
Tel: (512) 436 8000.
URL: http://www.tivoli.com

* * *

IBM has unveiled its Host Access Client Package, which replaces and upgrades the Personal Communications and Host On-Demand bundle. It provides access to applications and data residing on AS/400 (5250), System/390 (3270), and DEC/Unix (VT) hosts for traditional and Web users in SNA and intranet environments.

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

* * *

MacKinney Systems' JES Queue Client for Printers now includes TCP/IP printer support.

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

* * *

IBM has upgraded the VM/ESA platform in Capacity Services' Managed Application Server (MAS) outsourcing option, upgrading TCP/IP for VM/ESA V3R1 to V3R2.

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