

Personal Systems

IBM's MAGAZINE FOR TECHNICAL COORDINATORS

SEPTEMBER/OCTOBER 1994

REXX

**DEVELOPING
APPLICATIONS
WITH
VISUAL REXX**

**"SNEAKER NET" OR
SYSTEMS MANAGEMENT:
THE HARD WAY OR
THE EASY WAY**

**A REXX
HANDS-ON
TUTORIAL**

**UPGRADING
FROM MICROSOFT
LAN MANAGER
TO IBM
LAN SERVER 3.0**

**Visual REXX Products:
The Right One for Your Job**



BULK RATE
U.S. POSTAGE
PAID
PERMIT 1808
DALLAS, TEXAS

Watcom™ VX•REXX™

Visualize the Power of OS/2

NEW
CLIENT/SERVER EDITION

- Easily bind objects directly to your database
- Quickly connect to your database
- Visually generate and test your SQL query
- Rapidly create more than a dozen chart types with over 150 display options

PLUS

All of the Powerful Features
of the New VX•REXX
Version 2.1

NEW

VX•REXX
VERSION 2.1

- Over 2 dozen objects, including CUA'91 Containers, Notebooks, Pop-up Menus and more
- Integration and control of existing applications through DDE, keystrokes or REXX API's
- Easy to learn event-driven programming model with complete on-line documentation
- Context-sensitive help
- Powerful drag-and-drop programming techniques simplify programming
- Develop professional multi-threaded, multi-windowed and drag-and-drop enabled applications
- Code reusability through section sharing
- Include OS/2 style help and hints in your applications
- Advanced interactive source-level debugger
- System Object Model (SOM) based object manager
- Package your application as an EXE or PM macro for royalty-free distribution
- Integrated console window support simplifies migration of existing REXX programs

Watcom VX•REXX is an award winning, easy to use visual development environment for creating OS/2 applications with rich graphical user interfaces. VX•REXX combines a project management facility, visual designer and an interactive source-level debugger to deliver a very approachable and highly productive visual development environment.

Design Applications Visually

Create rich graphical applications quickly and easily using the visual design environment. With the visual designer, you can graphically create CUA'91 Presentation Manager interface objects, quickly customize their properties, and easily attach REXX procedures using powerful drag-and-drop programming techniques.

Integrated Development Environment

Build, test and debug your application without leaving the development environment. Then package your application as an EXE file or PM macro for royalty-free redistribution. The power of the integrated development environment and debugger can also be used with your existing REXX applications.

Powerful Open Environment

Enjoy the simplicity of event-driven programming together with the global editing capabilities essential for professional project management. Watcom VX•REXX is open and extensible through IBM's object-oriented System Object Model (SOM) technology. You can access all standard REXX API's including DB2/2, because VX•REXX is based on the OS/2 2.x standard system REXX.

Experts Agree... "Using VX•REXX for development is like driving a Porsche:

It's fast, it's compact, everything's in the right place, and it makes us look good, too."

Peter Coffee, PC WEEK, May 2, 1994.

"(VX•REXX) applications can be multithreaded and REXX is probably the easiest language in which to learn the OS/2 thread model."

Software Development, November, 1993.

"VX•REXX is a great tool; it's fun and productive." PC Techniques, Dec/Jan. 1994.



for OS/2



February 8, 1994

Watcom VX•REXX, Version 1.01



"The best new development tool of 1993"

Watcom VX•REXX

Version 2.1.....**Special Price \$99**
Client/Server Edition.....**Special Price \$299**

1-800-265-4555

Watcom

A Powersoft Company

Watcom International 415 Phillip Street, Waterloo, Ontario, Canada. N2L 3X2 Phone: (519) 886-3700 Fax: (519) 747-4971

*Prices and specification are subject to change without notice. Price does not include freight and taxes where applicable. Prices quoted in US dollars. Watcom, the Lightning Device, and VX•REXX are trademarks of Watcom International Corporation. Other trademarks are the properties of their respective owners. ©Copyright 1994 Watcom International Corporation.

Please circle #1 on reader service card.

**Power to run OS/2® and Windows™
with Reply's new 32-bit PowerBoard™
complete system upgrade**

Save your PS/2

Nobody could have guessed that the IBM PS/2 you bought a few years ago would turn out to be a dinosaur. But with new power-hungry Windows applications users want more processing power—now.

Meanwhile you're faced with computer hardware that hasn't been fully depreciated and an investment in expensive Micro Channel networking cards that aren't compatible with new PCs.

**Reply has the solution!
A complete, affordable motherboard upgrade**

Our upgrades are a cost-effective, complete solution. Starting at only \$495, you can bring a ponderous PS/2 up to speed. We provide a **total motherboard replacement**. In addition to the powerful 486 microprocessor, you get accelerated local-bus video,

increased memory capacity and the latest in affordable hard disk technology—all the components it takes to run OS/2 and Windows.

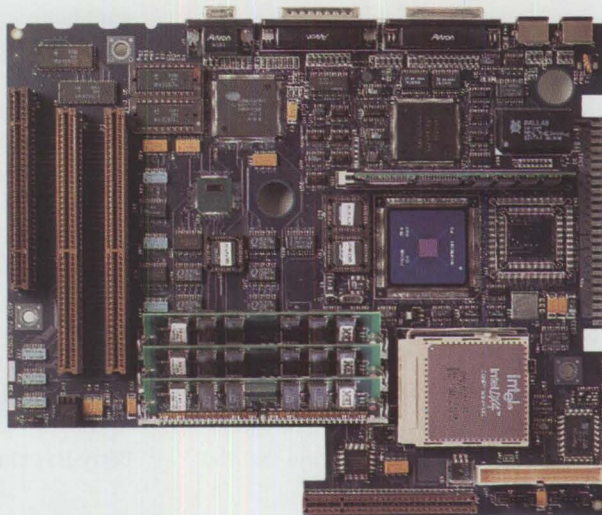
And you can still use your Micro Channel cards.

With Reply's TurboProcessor and PowerBoard upgrades, your sluggish PS/2s evolve into fast cutting-edge productivity tools, and your equipment budget stays in check. Upgrades are available for Models 25, 30, 50, 50Z, 55SX, 60, 65, 70 and 80.

from extinction

New Reply PowerBoards

Intel 486 processor
32-bit data path
Memory capacity to 64MB
64-bit local-bus SVGA video
with GUI accelerator
Enhanced IDE and existing
PS/2 hard drive support



"Reply Corp.'s PS/2 Model 60/80 motherboard upgrade replacement kit will breathe new life into those old Model 80s while protecting a company's investment in Micro Channel add-in boards."
—PC Week

reply CALL

1-800-955-5295

or fax 1-408-956-2793 to find out more about Reply System Upgrades along with our 30-day money-back guarantee and 3-year warranty

All registered trademarks are the property of their respective owners

Please circle #2 on reader service card.

Technology or Magic? Or Maybe Both!



As an editor of a magazine, I don't get out much! The number and frequency of the deadlines involved in the publication business sometimes makes me think *Personal Systems* gets published every other week instead of every other month. In today's voice mail and e-mail environment, I can go for months without seeing or speaking with a real person!

So, when the opportunity presents itself to actually meet one of *Personal Systems*' authors face to face, I'm certainly not going to turn it down. Last month, just such an opportunity arose when Phil Lieberman came to Dallas to teach a class. You know Phil—he's the guy who has made OS/2 threads and the Distributed Computing Environment understandable in the last couple of issues of *Personal Systems*.

So Phil and I met at a local refreshment establishment (LRE) after his class. If you've read Phil's biography at the end of his articles, you can imagine the topic of our conversation—magic! Phil told me early on that he belonged to the “other IBM”—the International Brotherhood of Magicians. So, naturally, I asked him to show me his stuff. Well, the rest of the time I sat there with my mouth open, asking over and over, “How did you do that?” He's good! Just ask the refreshment servers who neglected their other customers the whole time Phil was performing.

Phil said he started performing magic in the classes he teaches in order to provide some relief during the long, intensive days and the in-depth technical topics (can you imagine anyone needing any relief from that?). It's certainly an ice-breaker!

Phil performs magic with his articles, too. Back when he and I first discussed his OS/2 threads article (May/June 1994 issue), I told him that the

topic for which I got the most requests was REXX. REXX, as it turns out, is one of Phil's specialties, and it comprises a significant portion of a four-day course he designed and teaches: *OS/2 2.x: A Comprehensive Hands-On Workshop*.

His hands-on REXX tutorial is featured in this issue, along with another article in which Phil describes and compares visual REXX development environments. To round out our feature on REXX, we've also included several articles by REXX product developers describing their latest products.

So read on and soak up the magic of OS/2 REXX and its easy-to-use visual development environment.

What's Next?

We've got some great issues planned for you in the coming months: pen, multimedia, workplace, security, databases. These issues will also include our regular features on OS/2 and local area network products/solutions, technical Q&As, and the latest fixes.

If there's something else you need, tell us! If you want more information from our advertisers, tell them! We've included one insert in the magazine with a couple of different reply cards. Circle numbers on the reader service card to get more information from the advertisers on products in which you are interested. Tell us how we're doing on the editorial evaluation card—and include topics on which you want more information. Most of the articles in *Personal Systems* are driven by your requests!

Betty Hawkins, Editor

Looking For A LAN Fax Solution?



The FaxWorks™ line of fax software products offers state-of-the art fax power, ease-of-use and advanced fax administration capabilities.

Meet the FaxWorks family of LAN fax software products. An award-winning, value-rich line of fully-integrated PC-based fax software unmatched in its category. FaxWorks provides fax communications products for single and multi-user environments on DOS, Windows, and OS/2 platforms. The FaxWorks family is loaded with values: Including a common user-friendly interface. Outstanding viewing speed and power. Complete scalability. Broad fax hardware support. Ease of installation. And with prices starting as low as \$199, it's no question, the FaxWorks line offers exceptional value.

The Ultimate Fax Power For NetWare! FaxWorks Pro Server is an enterprise-wide fax solution that combines

two award-winning software products - the fax server power of FACSys® with the user-friendly client interface of SofNet's FaxWorks Pro standalone software. Supporting both DOS and Windows clients, FaxWorks Pro Server provides ease of use, FaxTracker™ fax management for compressing and assembling faxes, advanced viewing speed, fax annotation tools, and exceptional document management technology. FaxWorks Pro Server includes multi-phone line support, inbound routing and E-Mail integration.

FaxWorks Pro LAN runs on virtually any network without requiring a dedicated fax server. Send faxes from DOS and Windows with blazing fax viewing speed and powerful fax management capabilities.

FaxWorks Pro LAN provides users with many features including a cover sheet creator, OCR, scanner support, fax logs, phonebooks, annotation tools, and more!

The market leading OS/2 fax solution, FaxWorks OS/2 LAN allows you to fax from any DOS, Windows or OS/2 application. Combine documents from multiple applications, create your own customized cover sheets, have an unlimited number of phonebooks and more! Option available for up to 32 channels.

VAR Value Paks available now to resellers. Call SofNet or your distributor today.

1-800-FaxWorks

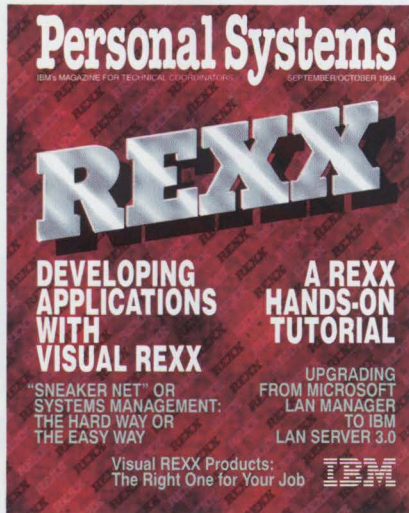
SOFNET

1110 Northchase Parkway
Suite 150
Marietta, GA 30067
(404) 984-8088
Fax: (404) 984-9956



Pricing in U.S. dollars, is for U.S. and Canada only. FaxWorks and FaxTracker are trademarks of SofNet, Inc. FACSys is a registered trademark of Optus Software, Inc. All other products are trademarks or registered trademarks of their respective manufacturers. Developer tested only. Novell makes no warranties with respect to this product.

Please circle #3 on reader service card.



ABOUT THE COVER

"Look far ahead as a racing motorist looks...at the horizon which he will have reached before he knows it," said essayist Arnold J. Toynbee on looking ahead. With the capabilities of visual REXX products, the ease of programming we thought was far in the future is here today!

Editor and Publisher
Betty Hawkins

**Business Manager/
Editorial Coordinator**
Donna Su

Editorial Assistant
Lia Wilson

Database Manager
Van Landrum

Subscription Coordinator
Rose McAlister

Production Coordinator
Elisa Davis

**Publication Services,
Typesetting, and Design**
Terry Pinkston/Corporate Graphics
Arlington, Texas

Illustrator
Bill Carr
Dallas, Texas

Printing
Dave Willburn/Motheral Printing
Fort Worth, Texas

Editorial Services
Mike Engelberg/Studio East
Boca Raton, Florida

Manager
Dan Russell



Printed on recycled paper and may be recycled.
Printed in U.S.A.

Contents

FOCUS

8

"Sneaker Net" or Systems Management?

Managing and supporting large installations of personal computers can be an overwhelming task. This article describes the value a good systems management product can provide your organization.

12

Like Father, Like Son

Todd Watson, a regular contributor to *Personal Systems*, provides one of those "lite" breaks from the weighty technology as only he can! This article provides a humorous look at the endless possibilities of "talking" to your computer as speech technology comes into its own.

15

The Book Shelf

To start our focus on REXX and REXX products, Lia Wilson reviews two prominent books by Hallett German on the REXX command language: *Command Language Cookbook for Mainframes, Minicomputers, and PC's* and *OS/2 2.1 REXX Handbook: Basics, Applications, and Tips*.

17

Cajun Electric Cooks Up OS/2 GUI with VisPro/REXX!

This article describes the speed with which the programming staff at Cajun Electric was able to develop an OS/2 graphical user interface with the help of HockWare's VisPro/REXX product.

TECHNICAL

29

A Hands-On Primer for REXX

This article is a tutorial about using REXX. It covers extensions, data types, operations on the data, assignments and console displays, getting and using input, using functions, repetitive operations, libraries of extensions, writing REXX programs, and graphical REXX environments.

34

Visual REXX Development Environments

Visual REXX development environments provide a fast, inexpensive way of developing OS/2 applications. They are quickly becoming a popular method of creating conceptual prototypes as well as line-of-business applications. This article provides a look at several of these products.

40

CID Installation of OS/2 2.11 and LAPS

This article explains how to add configuration, installation, and distribution (CID) support to the remote installation code server. It explains a CID installation of OS/2 2.11 and LAN Adapter and Protocol Support (LAPS).

45

Upgrading from Microsoft LAN Manager to IBM LAN Server 3.0

With the release of IBM's LAN Server 3.0 Advanced and related products, many organizations are upgrading from Microsoft's LAN Manager to IBM's LAN Server 3.0. This article details the steps necessary to convert LAN Manager servers to LAN Server 3.0 servers.

LITTLE SOLUTIONS

67

Questions and Answers

This issue includes a variety of REXX questions and answers.



Call us at 1-800-547-1283

IBM Personal Systems Technical Solutions is published bimonthly by End User and Software Solutions Services and Support, International Business Machines Corporation, Roanoke, Texas, U.S.A.

To make name or address changes, or to correspond with *Personal Systems*, write to the editor: IBM Corporation, Internal Zip 40-B3-04, One East Kirkwood Blvd., Roanoke, TX 76299-0015, fax to (817) 961-7218, or use Internet psts@vnet.ibm.com. For editorial questions, IBMers can reach the editor at DALVM41B(HAWKINSB); for other questions, IBMers can reach the business manager at DALVM41B(DONNASU).

Note to IBM personnel: IBM employees in divisions 12, 23, and 72 can subscribe to this publication using OV/VM (PROFS) and the INEWS facility: select hard copy for the PS_SOL category. IBMers in other divisions can call (800) 678-8014. Canadian and international subscribers can call (214) 550-0433.

© Copyright 1994 International Business Machines Corporation

BPA International Consumer Publication Membership applied for April, 1994.

21

Application Development by Program Integration

Technology exists today that makes it practical to build programs by integrating other programs. This article describes WATCOM International's VX-REXX product as a robust multi-purpose development tool for creating applications.

24

IBM REXX for NetWare

Easily customized and rich in function, IBM REXX for NetWare provides an easy-to-use interpreted language for building tools and utilities for NetWare.

25

GammaTech REXX SuperSet/2—Give Your REXX Programs the Power of C

This article describes the extensive set of external library functions for REXX provided by GammaTech REXX SuperSet/2, and it describes how this product can reduce and simplify your REXX development time.

27

BranchCard: A Viable Option to Stand-Alone Hubs

BranchCard, an intelligent hub product from LanOptics Inc., provides effective network management. This article describes the features of the product and elaborates on the problems an intelligent hub product can solve.

49

Stretching Your LAN with LAN Distance

Trying to maintain user productivity when users connect remotely into your LAN can be challenging. This article discusses IBM's LAN Distance technology and explains how LAN Distance provides remote access to local area networks.

54

DB2/2—More Than Ever Before!

DB2/2, IBM's relational database engine for OS/2, has become a much stronger offering with the introduction of many enhancements and the availability of a number of products that improve its value. This article looks at the recent enhancements and at one of those complementary products.

60

NetBIOS, SNA, and NetWare IPX Coexistence Under OS/2

This article describes NetWare's two different interfaces, ODI and NDIS, and how they can coexist using a single LAN adapter. Easy steps are provided for setting up coexistence in three different ways.

69

Corrective Service Information

Refer to this section for the latest maintenance release levels and other software service information.

Personal Systems Advertising Contacts

Personal Systems accepts paid advertising for applications, products, or services that run on or complement IBM's personal systems hardware and software products. To obtain a media kit and advertising rate information, contact one of the *Personal Systems* advertising sales representatives listed below.

Northeast Office

Peggy Schlatter
422 West Congress
Detroit, MI 48226
Phone (313) 963-8500
Fax (313) 963-0464

Southeast Office

Tom Flynn
290 Jade Cove Drive
Roswell, GA 30075
Phone (404) 993-2451
Fax (404) 993-4423

Central Office

Laura Stevens
422 West Congress
Detroit MI 48226
Phone (810) 661-9283
Fax (810) 661-8746

West Office

Kyle Walkenhorst
5394 Kenwood Drive
Salt Lake City, UT 84107
Phone (801) 262-1786
Fax (801) 262-1886

Q What's the best way to set up, administer, and troubleshoot your LAN Server domains?

A ICU for your LAN—
LAN "Intensive Care Utilities"
for IBM LAN Server 3.0

Are you spending too many hours adding users and assigning applications and resources?

Rx With our LAN **import tool**, you just create a simple text file with the information and actions you want performed, and the tool does the rest!

Do your users complain that they can't get to resources?

Rx With our **analyze tool**, you can check and repair your entire user domain for missing assignments, aliases, permissions and more—all based on user group memberships.

Have you lost sleep at night worrying about restoring a LAN—complete with corrupted DCDB or NET.ACC file?

Rx You can now save the entire domain in an editable ASCII file with our **export tool**.

Ever find yourself playing game after game of Solitaire when you *know* you should be doing cross domain administration?

Rx By using our import and export tools, you can capture vital information, edit it, then move things from place to place with simple batch procedures.

★ ★ ★ ★ ★ ★ ★ ★
Introductory Price — \$499.00
★ ★ ★ ★ ★ ★ ★ ★

Lieberman and Associates Design & Engineering Group

221 N. Robertson Blvd., Suite C

Beverly Hills, CA 90211

Phone: (800) 829-6263
Fax: (310) 550-1152
BBS: (310) 550-5980

IBMLink: DEV2203
CompuServe: 76426,363
OS2BBS1: LANUTIL

Getting IBM software support has just gotten simpler.



Before



After

We asked what you wanted in IBM software support, and the first thing we heard was, "Make it easier, quicker, more direct." So now it is.

Now, a single phone number gets you access to the right IBM experts fast, whether it's for simple usage questions or in-depth systems analysis, for almost any IBM system you have.

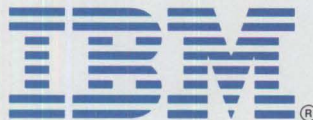
You can build your own support plan, choosing from the full

range of IBM Support Family services, and pay only for what you need.

You can get the same level of support across your entire business with just one signature on one contract.

And you said you were serious. So are we. Satisfaction is guaranteed or your money back.

To learn more, call us today. We think you'll like what you hear from us, because it's what we heard from you.



For more information about IBM software support call:

1 800 274-0015

©1994 IBM Corporation

Please circle #5 on reader service card.

“Sneaker Net” or Systems Management?

Managing and supporting large installations of personal computers can be an overwhelming task. Reactive management—responding to whomever yells first and loudest—can consume needless support personnel hours and create dissatisfaction among your customers (end users). Proactive management—responding to situations before they become problems—eliminates frustrations for both support personnel and end users. This article describes the value a good systems management product can provide your organization.

Technical support groups often use the humorous phrase “sneaker net” to refer to how reactive technical support is provided to a large group of computer users. Technicians “run” from user to user, workstation to workstation, floor to floor, and even building to building to support their end users or their workstations. Wearing sneakers helps a “runner” run to the next problem.

While the sneaker net analogy creates an amusing mind picture, it also describes an all too common method of providing technical support to users. Often, help desks are simply call-tracking clearinghouses through which the help desk staff receives phone calls from users, collects information about the user and the problem, then “attempts” to describe the problem (in writing) to pass it on to a technician who then visits the user to solve the problem.

Systems Management Tools Make Dollars and Sense!

Unlike the sneaker net, systems management tools enable the help desk staff to provide immediate assistance to users

when they first call. When your help desk provides this “first contact” level of assistance, users feel confident with help desk services. In turn, your technical staff will be available to address more complex, time consuming problems. For example, they will be able to provide a more personalized support service. Instead of correcting or solving one problem, then running to the next, the technician might provide information to the user on how or why the problem happened and ways to avoid or prevent the problem in the future.

Systems management tools also help local area network (LAN) administrators manage elements of a LAN, including LAN-connected workstations. Additional functions help LAN administrators to provide reliable and effective management services, including services for configuring a server or client workstation; distributing, installing, and managing software on client workstations; monitoring the performance of many LAN elements; and generating alerts when things go wrong. A good, proactive systems management solution shatters the



sneaker net paradigm and demonstrates competent technical support to users and management.

From Reactive to Proactive

Systems management is defined as "all the actions and procedures to accomplish the business-support activity of making information-systems services available." (*IBM LAN NetView Manage Administration Guide* [S96F-8492]).

Sneaker net is actually the first, primitive form of systems management. The sneaker net analogy describes a reactive approach to technical support, an approach in which LAN administrators react to problems after they occur. They react to performance issues after users start complaining. They react to disk storage problems after the disk becomes full and users can't save their day's work. The operative word here is "after"—not a particularly strategic move.

How can the needs of both the information systems (IS) department and end users be satisfied? The IS department wants reliability, security, and the ability to meet end-user needs in a planned and controlled way, while end users want the freedom to access data anywhere and anytime.

Customer satisfaction is a key indicator when IS managers measure the benefits of their technical support services. IS managers are continually challenged to maintain and improve on these measurements while economics and down-sizing continue to erode their resources.

How do you reduce your technical support staff's ever increasing workload while continuing to improve those same services? Is the pace of technological change outpacing your support staff's abilities? As technology advances, it becomes more and more complex to manage, maintain, repair, and support. Do you remember when applications were simple to install and run? When changing the hardware and/or software configuration had little or no impact on performance and reliability? Now, even the simplest changes have a tremendous impact on workstation functions and performance.

In the past, businesses considered personal computers disposable office

equipment. Today, personal computers are regarded as business systems. The services provided using PCs and LANs can make the difference between success and failure of your business. Managing these "PC assets" has created a myriad of "opportunities" equivalent to mainframe operations support. In other words, if time is money, then timely and reliable information is the key to your success!

An Overview of Systems Management Solutions

Systems management solutions provide

tools to proactively identify, monitor, and address network concerns. Configuration management, hardware and software management, security, performance monitoring, plus trend analysis and capacity planning are components of systems management that can give your organization the leading edge in responsive and reliable information-systems services access.

Configuration Management

One of the most powerful components of systems management is the ability to manage workstation configuration. With

Supercharge Your REXX!

GAMMATECH™

REXX SuperSet/2

Supercharge your REXX with the most complete set of REXX external functions available - GammaTech REXX SuperSet/2.

With over 300 functions from seven DLLs, the GammaTech REXX SuperSet/2 will initiate File and System operations, issue Network commands, execute Video I/O, manipulate Processes and Semaphores, regulate the MacroSpace and perform Math calculations.

Call us today at (405) 947-8080 and you'll see what REXX can do with some extra power!

- Perform EXECIO functions
- Manage LAN Server users by group or domain
- Perform LAN Server permission functions
- Manipulate LAN Server files, sessions and shares
- Interface with NetBIOS, TCP/IP and Communications Manager/2
- Open, close, query and create semaphores
- Start a thread
- Destroy a process or thread
- Copy, move or rename individual or mass files
- Dump variable pools or MacroSpace to a file
- Load, drop or reorganize functions in MacroSpace
- Perform logarithmic and trigonometric calculations
- Query hardware components
- List current system environment variables
- Write character and attribute strings
- Obtain and set cursor type and screen mode

STTS

**SofTouch Systems, Inc.
Workstation Division**

1300 S Meridian, Suite 600, Oklahoma City, OK 73108
(405) 947-8080 • Fax (405) 632-6537

©GammaTech, Inc. 1991-94 All Rights Reserved

Please circle #6 on reader service card.

configuration management tools, you can quickly and conveniently access detailed information about workstation hardware and software configurations.

Software installation programs sometimes change configuration files without the knowledge of the person installing the new software. How many times have your users reported a problem with their workstations while denying that they changed the AUTOEXEC.BAT, CONFIG.SYS, or a system .INI file? Searching for and discovering these changes takes a lot of time. Many astute technicians keep hard copy printouts of all their vital configuration files to use as comparison and reference points when problems appear. Others slyly "write-protect" configuration files so accidental and sometimes intentional changes are minimized. LAN administrators who have the ability to remotely view, compare, and change these system files can provide a very positive impact to the IS services your staff provides to your business.

Hardware and Software Management

Systems management provides tools giving the LAN administrator the ability to gather and display a broad variety of information about the hardware and software installed on LAN-connected workstations. Examples of system information that can be collected include:

- Drive information, including file system type (High-Performance File System [HPFS] or File Allocation Table [FAT]), available space on the drive, total drive size, and partition layout (physical or logical)
- Adapter identity; configuration information for installed adapters
- Memory configuration and usage, total physical memory, installed memory module identification
- Mouse type and settings
- Operating system information, version, and corrective service diskette (CSD) levels
- Printer and printer queue status information
- Workstation model, microprocessor information, model name, processor type and speed, and BIOS date
- Application software information

Automated asset management provides timely and easily obtained information for IS managers trying to balance technology, staff, budgets, and other resources.

Network-attached printers also impact office productivity. When do papers jam, toner and paper run out, and mechanical problems occur? These problems invariably occur when anyone in the office needs to print! Because of noise, odor, heat, and traffic, network printers are commonly located in rooms convenient to, but away from, employee work spaces; therefore, printer problems often go unnoticed or fail to get reported to the help desk until someone doesn't get his or her print-outs. Correcting paper jams usually requires more than basic knowledge of the printer. Printer ribbons and/or toner can be volume-purchased to obtain cost savings, and the support organization can install the printer supplies and recycle old toner cartridges.

LAN administrators can receive and process application-generated alerts. Alert logging, pop-up messages, program execution, and forwarding alerts to another managing system are some of the many actions that can be taken in response to alerts. If a "Toner Low" message is received, the technical group can send someone to replace the toner cartridge. If a paper jam or mechanical malfunction alert is received, LAN administrators can re-route print jobs to another printer. A message regarding the printer's status and including a repair time estimate can then be sent to users. Automatically creating a help desk repair request based on the alert information received is an example of a proactive (though unattended) management service solution.

System notifications can be used when selected LAN-connected systems go online or offline. Remote systems management tools automatically check each of the systems in the network to see if they are active according to a predetermined presence check interval. Alerts are also generated when predetermined threshold levels are exceeded for any of the monitored systems. Hard disk space on workstations and servers can be monitored with threshold management, providing a notification to the user of a workstation and/or to the LAN administrator before disk space is exhausted.

Software management can also include license management. IS managers are concerned about software that is legally and/or illegally installed. Good software management provides clues about whether or not an application's license count should be adjusted—providing the opportunity to compare usage/need volume reports with the pricing structures of application software. This information can also be used to determine if an application has become obsolete and should be removed to reclaim hard disk space.

Security

Most available network operating software has a certain level of built-in security. Network operating system security usually relates to granting access to the drives and directories that contain user data, public software applications, and LAN-attached peripherals. Referred to as *storage management*, data backup services are another form of "security." What is the impact on your business when data is lost? Depending on the type of data, the impact can be minor, severe, or even devastating.

In today's business environment, information is stored on workstations and LAN file servers across the enterprise. Leaving the protection of information to your end users may not guarantee that data will be backed up regularly. In many cases, the process of backing up multiple workstations and file servers has become labor intensive, time consuming, and costly. Protecting corporate data through automated backup software has become an essential part of systems management security. (See "Back Up for the Future" in the May/June 1994 issue of *Personal Systems*.)

Performance Monitoring

How many times have you heard "The LAN is really slow today"? Or "I keep getting this nasty message saying that an application is not available at this time, and try again later"? Since LAN operations are so complex, many factors impact performance—including network operating system and hardware tuning, network traffic, number of logged-on users, number of users concurrently using a single application, and so on.

A good systems management solution gives LAN administrators the tools to monitor both server and workstation performance. The LAN administrator can then use this information to prevent network problems. Measuring CPU utilization, memory, and hard disk usage are the basic tools in addressing performance issues.

Trend Analysis and Capacity Planning

Now that you have collected or gathered all this great information about your LAN, and you've checked everything that needs attention, is the information worth keeping? Can this information be useful in the future?

Yes! All the information collected by the systems management tools (configuration management, hardware and software management, security, and performance monitoring) can be kept, stored, and analyzed for trends and capacity planning both now and in the future.

Alerts and threshold alarms can be logged into a file or database. Sorting this information can easily provide clues to repetitive problems, software licensing, usage trends, CPU utilization, and security violations. You can then exploit this information to determine if more memory is needed or if a new PC with a faster, more powerful processor may be required.

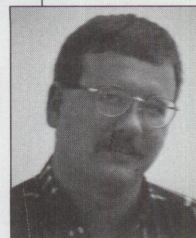
So . . . Now What?

Today, by combining systems management tools with the multitasking power of OS/2, LAN administrators can quickly, easily, and automatically collect information from servers and workstations connected locally and/or across the enterprise. Data is collected with no interaction from the workstation user. Information and error messages can be captured for accurate problem identification. Your help desk staff can quickly and easily determine the cause of the problem and resolve it—all while the user is still on the phone. Investing in any or all of the systems management components can provide immediate, positive benefits to the productivity of your business.

Systems management provides the management services that control the information technology used to meet your business needs and priorities. They help you plan, coordinate, operate,

and support the heterogeneous information systems across your enterprise, including mainframes, midrange computers, and LANs.

Contact your IBM services representative for specific information on some exciting new IBM systems management solutions and their outstanding services and support that can benefit your business.



Tom DeWispelaere is a marketing support representative in the IBM Personal Systems Competency Center in Roanoke, Texas. He provides technical marketing support on OS/2 systems management products. Tom joined IBM in 1993.

Direct your comments and questions about this article to psts@vnet.ibm.com.

OnCmd[®] xBase for OS/2 ... A Winning Combination!

OnCmd harnesses the power and the speed of OS/2, enabling you to develop new applications or convert existing xBase applications, such as those developed in FoxPro[®], Clipper[®], and dBase[®].



Preserving Your xBase Investment

OnCmd can quickly convert your xBase applications to run directly in OS/2's 32 bit Presentation Manager. This minimizes development time and preserves your current xBase investment. Text applications are automatically converted to a native Presentation Manager application that is both multi-user and network capable. With simple code changes, you can add buttons, check boxes, and drop-down menus to give your application more functionality.



Applications Development

OnCmd's screen painter, client-server facilities, "event driven" programming, and Dynamic Link Libraries (DLL) all add to the development power that you need.

A Performance Winner

OnCmd is a performance winner! No windows emulation overhead. Disk requirements less than 1 MB. Installation under 5 minutes. In addition, it typically indexes large databases twice as fast as the competition in half the disk space. For more benchmark details, call or email us.

*Go for the gold.
Order your
copy today!*



ON-LINE

ON-LINE DATA
5 Hill Street, P.O. Box 65, Kitchener, Ontario, Canada N2G 3X4
Tel: (519) 579-3930 Fax: (519) 579-2130
CompuServe: 70022,104 Internet: oncnd@onlinedata.com

* Prices are in US funds. Add \$249 per 10 user blocks (Regular \$498).

1-800-265-2455

Product also available from: Indelible Blue Inc. 1-800-776-8284 (USA) 1-800-672-4255 (CDN.)
Programmer's Paradise 1-800-445-7899 The Software Store (IBM) 1-800-342-6672
Online Data recognizes the following trademarks: FoxPro (Microsoft Corporation), dBase (Borland International Inc.), Clipper (Computer Associates International Inc.), OnCmd (Online Data).

Please circle #7 on reader service card.

Like Father, Like Son

The pointy-headed bit twiddlers in the development labs tell me I'm going to be able to start talking to my computer soon. They call it "speech recognition." They say I'll be able to say all sorts of things to it and make it do things I've never been able to make it do before.

I hate to burst their bubble, but I've been talking to my computer for years. Or I should say, I've been *yelling* at it.

Most of the things I've yelled at my various computers aren't fit to be printed in this magazine; suffice it to say that most of those verbal denunciations didn't include syntax you'd want to use at Grandma's house.

No, the technodroids say, now your computer will *understand* what it is that you're saying.

"Will it understand what I'm yelling?" I shriek.

They stare at me dumbfounded, shaking their heads as they slink back into the dark confines of their labs, as if I don't appreciate the thirty-odd years of research it has taken them to perfect this speech recognition technology.

They don't understand. I *want* my computer to understand what I'm yelling at it. I can't tell you the number of times I've cursed one of the five computers I've owned in my brief but byte-ridden life and *prayed* that it would understand what I was saying.

Computers *should* listen to us. We *created* them. They should do exactly what we say, how we say, and when we say to do it.

They're kind of like kids in that respect. I don't have any kids myself. But I was

a kid once, and I remember what it was like.

When I was growing up, my dad didn't have a computer to yell at, so he yelled at me instead. I can't repeat most of the things he yelled, either.

He didn't have to yell at me. He could've yelled at the TI calculator sitting on his desk. It looked a little like a computer. He certainly used it like one. But this was before most people could afford to own a computer they could yell at, or even realized that computers were good for venting your frustrations.

I kind of respected my dad for not yelling at his calculator. He would've looked awful funny, the thing was so small. And what if the other kids in the neighborhood had found out? *Whew*. I never would have lived that one down.

Then the first IBM PC came out.

My dad's an independent insurance agent. The IBM PC came along well before the 1986 Texas oil bust, when he was still raking in the dough, so he emptied my college fund and bought himself a PC. I was just a teenage punk; I didn't know he was spending my college money. He probably figured I wasn't smart enough to go to college.

Things got crazy when the new PC arrived. It was like the stork dropped off another rug rat, the attention he paid to it. Mom shook her head—first golf on Wednesday afternoons and all day Saturdays and Sundays, poker on Wednesday nights, and now computers. She could already hear *her* friends, too: "Did you hear about poor Mrs. Watson? Her husband got one of those fancy new calculators." Widowed at an early age.

Dad talked to his new PC gently at first, just like a man should talk to his newborn infant. He'd lean in real close to the screen and whisper, making goo-goo ga-ga



sounds as his breath fogged up the glass. I think he would have picked it up and rocked it in his arms if it wouldn't have given him a hernia, the thing was so heavy.

But after the first couple of weeks, the new wore off. Just as if he were ordering his loving teenage son to empty the compactor, he'd tell it to go do something productive. The computer would just sit there, cursor blinking, a cute babbling, saliva-dribbling ball of plastic protoplasm.

"Why won't you open my #&%# file?!" he'd scream.

This was before I knew what a computer "file" was. I'd stare at Dad's file cabinets for hours, waiting for one of them to open magically. It never happened. Dad kept screaming.

The moment Dad started yelling at an inanimate object I figured I was off the hook. He wouldn't have time to yell at me any more. He was going to be too busy trying to make that PC thing open up his file cabinets.

Talk about counting your virtual chickens before they're hatched.

See, this was all before I learned how to use the PC. Heck, I was 15—I'd barely figured out how to use my hormones.

Then I read Hemingway.

And I learned all about hormones, and about World War I, and deep sea fishing, and Spanish wine, and bullfighting, and that I wanted to become a writer. Dad's computer seemed the perfect tool to help me in my new endeavor.

At first, the PC was off limits to me. Like Dad's Limited Edition Liberty Bicentennial Celebration Lynx golf clubs, I could look but not touch.

Slowly but surely, though, he'd let me get closer to the computer until I, too, was able to lean over and fog up the screen.

Like father, like son.

One day I was toiling away at the PC when I realized I was having my first case

of writer's block. I'd read Hemingway had had such a condition on and off throughout his life. His solution was to drink heavily. That wasn't an option for me—at least not yet, anyway—so instead I started fiddling around with the computer, trying to get the file cabinets to open. Nothing I did had any effect—until I came across the "format" command, that is.

You could hear dad's screams all through the neighborhood. Mine weren't far behind.

My reasoning was sound for a trade-school bound 15-year-old writer wannabe: Format seemed as likely a command as any for opening the file cabinets.

Far from it. Dad was now yelling at both of us—the computer and myself, that is. This was my first experience with multitasking—Dad yelling at me from one side of his mouth and at his beloved IBM PC from the other side.

That was the first time I wished the *computer* could talk. Maybe it could have explained that I was merely trying

INDELIBLE BLUE

THE NEXT GENERATION OF SOFTWARE

Special REXX Offer

- **GammaTech REXX Superset/2** by SofTouch Systems, Inc. **NEW!** Complete REXX external functions in one package; 7 DLLs, over 300 functions. Initiate file & system operations, video I/O, much more! **MSRP \$79.95**
 - **VisPro/REXX Gold** by HockWare
Using this visual programming tool for OS/2, even novices become adept in no time. Quickly increase your programming productivity! **MSRP \$299.00**
 - **REXX Reference Summary Handbook** by Goran
A quick reference pocket guide for all OS/2 2.1 REXX commands and syntax, newly updated. Winner of OS/2 Magazine Editors Choice Award. **MSRP \$19.95**
- BUNDLE PRICE: \$280.00 SAVE \$119.00**

Offer expires 11/30/94

VALU Discount Program

- **Great savings!** Discounts up to 65% on additional licenses for IBM software
- **Discounts** based on 2-year volume forecast, no minimum purchase quantities
- **Easy budgeting!** Prices protected against increases for 2 years
- **Convenience!** Post-installation reporting eliminates lengthy ordering procedure
- **NEW!** TCP/IP products now eligible for VALU

- **Indelible Blue, Inc.** is "your single source for OS/2 solutions". In our full color catalog, you'll find hundreds of IBM (and third party) applications.



OS/2 and IBM are trademarks of IBM Corporation. All other registered trademarks belong to their owners.

**FULL COLOR CATALOG AVAILABLE
ORDERS: 1-800-776-8284**



Please circle #8 on reader service card.

to write the great American novel, á la Ernest Hemingway. An admirable notion, right? But that PC wasn't saying a word—I'd erased all its file cabinets.

When I realized I had not only *not* opened the file cabinets, but also completely erased them, I decided that writing was too dangerous an occupation and I'd be better off pursuing less risky employment—like being a bike messenger in New York City.

After I got hit by the cab on Park Avenue, I decided that bike messengering was too dangerous an occupation, and in order for me to live past 25, I should probably go back to writing. Live and learn.

Despite my tragic formatting experience, both for my psyche and my rear end, I became enthralled with computers. Right before I became intimate with the New York taxi, and just as I was making the transition back to writer wannabe, I bought my first laptop computer. The year was 1987. Made by a small Japanese company called Sord, the laptop I bought used a small microcassette as its storage medium. The idea seems laughable now, but back then I thought I'd gotten a really good deal.

My next PC purchase was a Kaypro lug-gable that used CP/M as its operating system. Of course, I hadn't yet figured out what an operating system was—I didn't even know such a thing existed. To my mind, DOS was Spanish for "two."

However, I *had* finally figured out what a "file" was, much to my relief. Jiminy Cricket, I was well down the road to computer literacy.

My Kaypro was a trusty servant indeed. When the writing wasn't going well, I could yell at it all I wanted and it would sit there and take all the abuse I could dish out. Never mind the fact that it didn't have a hard drive. Or the fact that I still didn't know what a hard drive was.

Like father, like son.

But alas, CP/M was destined for the great computer operating system junkyard in

the sky, and there I stood, now fully comprehending of my hormones, but still an unsullied innocent at understanding the wider ramifications of the true meaning of DOS.

Though ignorance may indeed be bliss, the ever-changing information revolution just couldn't stand for my technidiocy. My saving grace came in the form of following the timeless advice of my loving, strong-lunged father: "Get a job!"

What he really meant, and what his PC would gladly have told me had it yet learned how to speak, was that he was no longer paying my bills and that it was time for me to move out into the real world. *What?* Had I been living in some nondescript virtual reality without even knowing it? Was risking my life riding around the streets of NYC on a mission for the rapid deployment of Madison Avenue's latest advertising *blitzkrieg* not a real job?

And wait a minute—virtual reality hadn't even been invented yet.

The inevitable compromise came in the form of becoming a master word processor and graphics whiz in both the Macintosh and PC operating environments. Suddenly, operating systems and applications became part of my everyday lexicon. DOS, System 7, and I all became good buddies. WordPerfect was a virtual next-door neighbor. Microsoft Word and I drank at the same watering hole.

All this before I had begun wandering aimlessly around the hallowed halls of Big Blue.

I can only tell you that it's been uphill ever since. If it's not speech recognition, it's Workplace OS. If it's not Workplace OS, it's asynchronous transfer mode. If it's not ATM, it's the microkernel. And so on, *ad infinitum*.

At home, I now drive a 486 66 MHz DX2 with a 340 MB hard drive, 16 MB of RAM, a 1 MB video cache, and a double-speed CD-ROM with a SoundBlaster 16 sound card that can create virtual experiences that are about as real as I care to get. It's like I'd been driving a Yugo all these years and somebody just handed me

the keys to my own candy apple red Porsche 944.

I named her Bessy.

I talk real gently to ol' Bessy. See, my whole life revolves around Bessy's whirling CD-ROM drive, her torturously fast hard drive, her mind-bending flash access RAM, her 14.4 modem connection to the world outside my apartment.

Bessy's my front-end to the universe.

So I don't yell at her. Not like I used to yell at my other computers. Now I just. . . whisper.

Yes, they say I'll be able to start talking to my computer soon, but I say to them I've been talking to my computer for years.

And the possibility that Bessy will soon understand what it is that I'm telling her no longer frightens me as it once did. Because I know that one thing is certain—while she may understand what I say, she will never, *ever*, understand what I really mean.

And somehow it seems to me that we ought to keep it that way.

Anyway, it doesn't really matter. I've already made up my mind. Old habits may be hard to break, but I'm not going to yell at any of my computers anymore.

I'm going to have me some kids.

Like father, like son.



Todd Watson has worked for IBM since 1991 as the assistant editor for */AIXtra: IBM's Magazine For AIX Professionals* and will soon complete his master's degree in Mass Media

Studies at the University of North Texas at Denton. He received his BA there in 1991, concentrating in composition and literature.

Direct your questions and comments about this article to Watson at radar@vnet.ibm.com or psts@vnet.ibm.com.

The Book Shelf

Command Language Cookbook for Mainframes, Minicomputers, and PC's: DOS/OS/2 Batch Language, CLIST, DCL, Perl, REXX by Hallett German

Author Hallett German serves up a winning dish with his *Command Language Cookbook for Mainframes, Minicomputers, and PC's: DOS/OS/2 Batch Language, CLIST, DCL, Perl, REXX*. This extraordinarily well-organized, reader-friendly text covers the major command languages, comparing their strengths and weaknesses. Emphasizing early that this book is neither "a tutorial nor a reference on command language syntax," the author explains that it is "a how-to reference once you have selected your command language and type of application." This book is designed to help you choose the most appropriate language for your task.

Stressing that by knowing which command language to use, you can complete one-third of your application before you write even a single line of code, German includes a handy reference for choosing a command language. This reference asks a series of pertinent questions concerning the application development process, then lists the various chapters and sections related to each step.

You won't get stranded when you reach for this book; "Chapter 1: A Road Map" provides you with clear directions for using the book as well as outlines the book's goals:

- "To offer a one-stop guide to the major command languages. . ."
- "To provide, for the first time, a reference that objectively compares the strengths and weaknesses of the major command languages."
- "To provide an approach for developing command language applications that can be used with any command language, an approach that also follows the structure of the book."

- To provide ". . . 'real world,' ready-to-run examples using handy sequences needed to do the job."

Although written for those with some basic knowledge and experience with a command language, this book covers all the bases. Beginning with a definition of the term "command language" and an introduction to command languages and applications, the book then devotes a chapter to each of the major command languages: CLIST, REXX, DCL, DOS and OS/2 Batch Language, and Perl, focusing heavily on REXX. Two of the most valuable aspects of the book are the "ready-to-run examples using handy sequences needed to do the job" and the appendix, which covers commercial software for different command languages. Those eager for more information will be delighted at the extensive list of "suggested readings" located at the end of each chapter.

Since we're focusing on REXX in this issue of *Personal Systems*, I'll give you some highlights from the chapter on REXX. Although when referring to REXX's flexibility as a command language, German's statement that REXX is "a floor wax AND a dessert topping!" may be stretching things a bit, he does provide examples of the many situations where you might want to use REXX, for example, when you must quickly create an application prototype, perform date and time operations, build your own system/utility commands, or need an alternative to the DOS batch language.

Further in the chapter you'll read about front-end REXX applications, system/utility REXX applications, self-contained REXX applications, and other REXX exec concerns such as debugging. You'll also learn how to develop your own "good exec style." To help you do this, German suggests practical ways to make your REXX execs readable and effective through tips such as "Minimize the number of multiple clauses appearing on the same line to enhance readability and ease of debugging" and "Don't use a SIGNAL as a GOTO

and try to prematurely exit a DO loop/flow control structure."

To order this book, call Van Nostrand Reinhold toll free at (800) 842-3636.

Published by Van Nostrand Reinhold
New York, New York
ISBN Number 0-442-00801-5
IBM Pub Order Number SR28-4401

OS/2 2.1 REXX Handbook: Basics, Applications, and Tips by Hallett German

In keeping with the overall REXX flavor of this issue of *Personal Systems*, we bring Hallett German's latest book, *OS/2 2.1 REXX Handbook: Basics, Applications, and Tips*, to your attention. You'll find this book useful whether you're new to OS/2 and REXX or are familiar with REXX but not with how it works with OS/2. If you're a system integrator converting REXX procedures between different platforms, a programmer, or end user, you'll find helpful information in this book. "This book is a great source of information on programming with REXX on OS/2," states Patrick Mueller in the Foreword, and he's right!

Hallett German immediately points you in the right direction with "Chapter 1, A Road Map." How should you read the book? Do the book's goals meet your needs? Spend a few minutes reading this chapter, and you'll know exactly how to use and benefit from this book. Chapter 1 clearly outlines the goals of the book—to offer a reference, tutorial, and idea book for OS/2 REXX—as well as explains who should read the book and how.

You can access the information in the *Handbook* through one of two approaches: either by type of audience (for example, whether you are a new OS/2 and REXX user or a REXX user who wants to create portable procedures) or by type of question (for example, "When should you use REXX to develop an OS/2

application?" or "How can you interface REXX with third-party shareware and commercial software?"). A handy little reference helps you quickly choose the chapters covering information that best meets your needs.

Here's a sample of just a few of the topics covered in this book:

- REXX Essentials
- OS/2 REXX: Unique Features

- OS/2 REXX: IBM Extensions
- OS/2 REXX: A Third-Party Interpreter

For new OS/2 and REXX users, "Chapter 2: REXX Essentials" is the best place to start. Here you'll obtain a knowledge base on which to build when reading the remainder of the book. Then you'll move on to read about "standard OS/2 REXX." German examines IBM extensions and focuses on three areas: extended services, networking products, and Visual REXX and IBM copyrighted free-ware.

Later in the book you'll find a chapter devoted exclusively to Personal REXX for OS/2 by Quercus Systems, "... the only available third-party OS/2 REXX interpreter." In addition to covering Personal REXX's unique features, German compares and contrasts the IBM and Quercus REXX interpreters through comparison charts and benchmarks.

Based on the premise that you can "expect third-party OS/2 REXX products to continue to arrive in healthy numbers in future years," German spends some time on third-party non-interpreter products including editors, communication packages, and programming tools from heavyweights such as Borland and Lotus.

This comprehensive look at REXX is completed with a look at some other related topics covered in the Appendices. You'll find everything from a reader comments page to REXX Error Messages and REXX BBSS, from REXX Shareware to a review of VX-REXX from Watcom.

To order this book, call Van Nostrand Reinhold toll free at (800) 842-3636.

Published by Van Nostrand Reinhold
New York, New York
ISBN Number 0-442-01734-0
IBM Pub Order Number SR28-5250



Lia Wilson, editorial assistant for *Personal Systems* magazine, has worked for IBM for five years. She has a BA in English and is currently completing a masters degree in English at the University of North Texas at Denton.

NEW VERSION FOR OS/2

DESCRIBE® Word Processor 5.0

The highest quality, NATIVE, 32-BIT, OS/2 wordprocessor available. Developed for OS/2 since 1989, "DeScribe is

the best written software in my department...including OS/2." Ron Garrett, IBM Corp., NY. *Buy DeScribe today.*

\$299 DeScribe 5.0 SE — Subscription Edition includes complete software and documentation, PLUS free upgrades and support through Dec. 31, 1995.

\$159 DeScribe 5.0 CX — includes complete 5.0 software and documentation, free installation support and access to Pay Support Services.

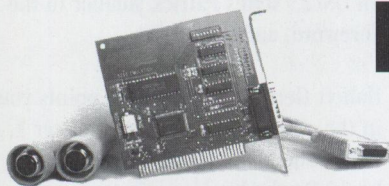
\$29 Lights — Continuous CPU Monitoring Utility. Automatically included in DeScribe SE and CX products.

To order, call: DeScribe, Inc. **1-916-646-1111** (West Coast)
CREDIT CARDS ONLY **1-813-732-5500** (East Coast)

*Execute REXX procedures from within DeScribe. Pass up to 100 variables.
*LAN, Site License, Windows™ and Windows NT versions available.

Please circle #9 on reader service card.

WHY Pay too Much for PC Emulation?



5250 EMULATION

\$185

Great Performance + Great Price = COMMON CENTS

IBM COMPATIBLE

- USA Made
- 100% IBM Compatible
- Runs all IBM 5250 Emulation Software
- Runs all IBM PC/Support Software
- 5 sessions under PC/Support
- 2 Year Warranty

ALSO AVAILABLE

- Micro Channel
- DOS Software
- Windows Software

TO ORDER CALL:

1-800-542-9144

FREE
15 Day
Evaluation

K G A
TECHNOLOGIES
C O R P

Cajun Electric Cooks Up OS/2 GUI with VisPro/REXX!

On a steamy summer night, Don Babcock of Cajun Electric examined a sticky problem. Outside his office, a million Louisiana residents kept cool with energy provided by his employer. Inside, his team of COBOL programmers were sweating it out, trying to produce OS/2 graphical user interface (GUI) applications. They had never worked with OS/2 and were unfamiliar with PC development.

Cajun Electric is a \$400-million-a-year power cooperative with 580 employees. It's a conventional IBM shop, operating a 4381 mainframe using DATABASE 2 OS/2 (DB2/2), Customer Information Control System (CICS), and an IBM token-ring with 250 workstations. Fifty of the workstations run OS/2 (Cajun is upgrading the remainder to OS/2). These users needed OS/2 applications to easily access the company's data for administrative tasks. Data was handled by DB2, which worked well for them, but as Babcock puts it, "Without a way to get at that data, the company would have been better off without it." His team needed to write applications around the data, and their old COBOL techniques were too difficult and time-consuming.

As client services manager, Babcock was responsible for finding a way to program OS/2 applications around DB2/2. His search began with a trial run of a KnowledgeWare CASE tool. Unfortunately, because of the high cost, only one team member could use the product, and it ultimately left them with third generation COBOL code, which didn't really solve the problem. Even the C generator products available were cumbersome in terms of development. It seemed that the products Babcock looked at were either too restrictive for his development needs or they lacked power. Most were out of his price range.

Then Babcock came across HockWare's VisPro/REXX, the first visual REXX programming product available for OS/2. The product had a reputation for being extremely easy to use, yet very powerful. What raised Babcock's interest was the fact that it used systems application architecture (SAA) REXX (a relatively easy language), offered royalty-free distribution to users, and required no runtime dynamic link libraries (DLLs) to distribute the applications. Moreover, Babcock could outfit his entire development team with VisPro/REXX for less than he could provide a single developer with any of the products he had considered. He especially liked that his programmers needed no formal training on the product.

Workplace Shell Integration Made It Easy

Within a week of introducing VisPro/REXX, Babcock's team was programming OS/2 GUI applications. After three months, the department delivered its first OS/2 application to the company's staff. Babcock says there was no real training period among the programmers; they learned as they went or, rather, produced as they went. The programmers learned REXX as they used the product and produced their own applications. Babcock says HockWare's technical support was a great help, answering questions readily and even calling back to confirm solutions. The transition to GUI programming was tremendously uplifting for the seven-person development team who had at one time viewed learning GUI programming as a monumental task. Babcock credits VisPro/REXX's easy-to-use interface for his team's enthusiasm and productivity.

"HockWare did it right," he says, "They designed VisPro from the ground up, drawing on all the advantages of OS/2." It's true. Ignoring the old Visual Basic paradigm, HockWare gave VisPro/REXX an interface appropriate for an OS/2

product by integrating it into the Workplace Shell. The result is that VisPro/REXX looks and feels just like the Workplace Shell, giving it, as Mike Gove, editor of *OS/2 Pointers*, remarked, "a definite sense of rightness." The ease and consistency of design puts programmers at the wheel and allows them to get on with the business of doing what they are there for: programming GUI applications.

The Ease of Objects and Multiple Views

The consistency of design is apparent in VisPro/REXX's objects and the multiple views available for working with the objects. All the objects behave similarly to the VisPro/REXX programmer. This is not the case in Presentation Manager programming, where the behavior of each object is unique and must be learned. A good case in point is VisPro/REXX's full-motion video multimedia object. It could take weeks for even a seasoned PM programmer to learn the behavior of this object. HockWare made its unique behavior transparent to the programmer by structuring the object the same as any other VisPro/REXX object.

Objects can be displayed within different CUA '91 views, depending on the task being performed. To design the application, the Layout view is used. It includes icons of the available objects (push buttons, radio buttons, etc.) and a canvas, which represents the window (called a form) of the finished product. The programmer drags the desired objects to the canvas. To specify what happens when an event occurs (for example, a push button being selected), programmers simply get a pop-up menu on the object and make their choice. The Layout view is best suited to the design phase of the application. VisPro/REXX includes four other views, appropriate to other programming tasks.

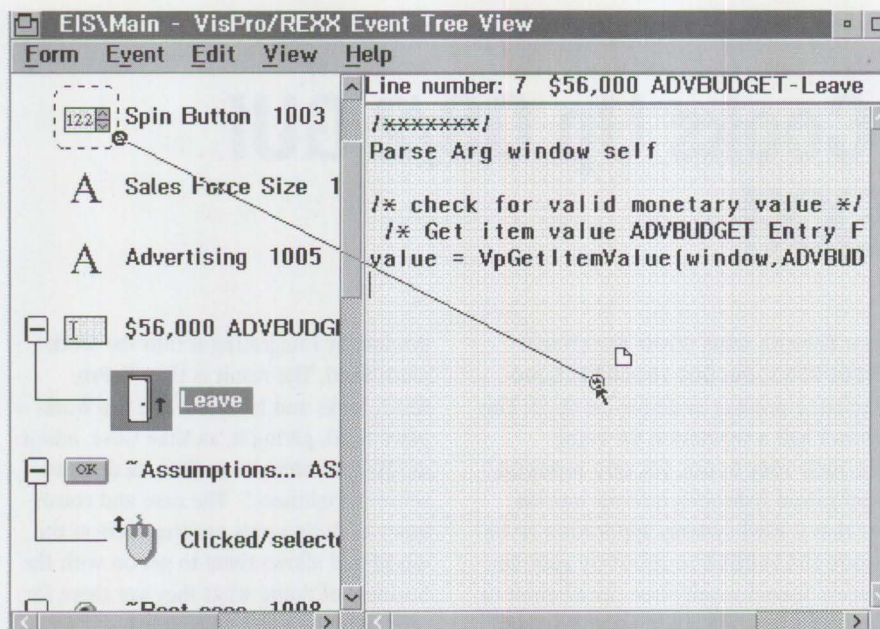


Figure 1. Drag-and-Drop Programming in the Event Tree View

The List view displays all the information about the application in an easy-to-read tabular format. This information includes object IDs, tips, OS/2 help IDs, and descriptive names. The programmer can change any of this information using direct editing, which consists of simply selecting the old text and typing the new text over the old. The order of the objects in the list allows the programmer to easily see the cursor tabbing order of the application's objects. To change the cursor tabbing order, the programmer simply drags the objects to their desired location.

To program object events, VisPro/REXX developers use the Event tree view, which divides the window into a left and a right side. The left side visually displays the objects and indicates any events associated with them in a tree format, with objects appearing as "roots" and event icons appearing as "leaves" on the tree. The right side contains the code for the object events. To program, developers simply drag an object on the left side to the right side (Figure 1). A window pops up, presenting all the possible programming choices for the object. These choices are presented in clear English, and, as with the entire product, the programmer can easily get a detailed description of the choice by pressing F1 for a help window. Upon making the appropriate selection, programmers see the code automatically appear on the right side. Programmers are free to modify the code, drag in REXX

code from Workplace Shell file objects, or even type their own code. With this feature, called "drag-and-drop programming," the programmer's hand rarely touches the keyboard at all. This greatly reduces effort, learning, and errors on the part of the VisPro/REXX programmer.

The Settings view for a form allows programmers to customize properties for the form. Menus, timers, window styles, title bar text, and window icons can all be programmed from this view. The Settings view for an object allows the programmer to quickly and easily change the object's ID, font, color, size or position, and other characteristics (for example, whether or not the object has a border). To get the Settings view for an object, the user double-clicks the object and notebook pages appear, displaying the attributes of the object.

More Objects

Babcock says the new System Object Model (SOM)-based objects HockWare is currently shipping should really help his team, especially the formatted entry field, which verifies that the correct information is typed. The formatted entry field ships as part of the Data Entry Object Pack add-on, which also includes a spreadsheet, split bar, clock, and calendar object. These objects fit right into the base VisPro/REXX product. For example, if a VisPro/REXX application currently has an entry field in which the user types

a date, the VisPro/REXX programmer might choose to plug in a calendar object instead so that the user just selects the current date from a calendar. HockWare's second object pack is the Image Processing Object Pack, which includes a pattern selector, color selector, and image editor/display object.

Babcock is also excited about VisPro/REXX's SOM extensibility toolkit, which shipped last January with version 2.0. He says he is so busy trying to keep up with the company's database needs that he's only played with it a little. But he's eager to see some of the objects his team or other customers put together. He says he would like to see HockWare or a third party develop a communications object pack, offering objects for individuals using Communications Manager or other communication links.

Shortening the Development Cycle

It's not surprising that the use of a tool like VisPro/REXX has led to new ways of handling development at Cajun. "Our approach is totally different now," says Babcock, "In the past, we'd spend months designing a product, then we'd start coding and debugging. Once the product was working, we'd show it to the end users who would usually say that wasn't what they wanted. With VisPro/REXX, we can get something on the screen in minutes. We'll work on the code for two weeks and turn it over to users for, say, a month. During this time, they actually use the product and tell us what they like or don't like. We fix what they don't like, then give it back to them for about three months. Finally, we fine-tune the application based on their comments. By this time, we've got a couple other applications in the works. The end user has more input and we're much more productive."

Even debugging applications has become minor for the team. The VisPro/REXX debugging environment incorporates the best features of a form's Event tree view. Rather than viewing a large list of apparently unrelated code modules, the programmer easily browses object event code by clicking on the appropriate icon in a tree structure. The programmer can view the contents of variables, step through code statements, and set breakpoints.

With VisPro/REXX, programmers can even debug multiple threads by setting breakpoints, halting, and resuming threads (Figure 2). And, as Babcock puts it, "What good is the ability to create multithreaded programs if you can't debug them?"

Database Design Made Easy

Babcock and his team originally used VisPro/REXX 1.0 to build front ends to their database applications, but the databases themselves were created manually. When Babcock heard VisPro/REXX 2.0 was coming out with a database feature, he expected it to have only a few new methods. What he didn't expect was a full-function database designer. With VisPro/REXX's database designer, tables are added just like objects are in the base product, presenting industry standard entity-relationship diagrams (Figure 3). This gave Babcock's department a huge productivity leap. Now his team uses VisPro/REXX to visually create databases on their PCs while simultaneously designing the user interface. Because developers have complete administrative authority over their PCs, they can quickly develop the interface and test the application on a local database without a lot of mainframe administrative overhead!

Upsizing to the Mainframe

Cajun's database was first deployed on an OS/2 server with applications accessing data from DB2/2. When it came time to deploy their applications throughout the company, giving users access to the same data, the server was no longer adequate. After evaluating performance and security considerations, Cajun decided to use its mainframe. The system administrator reconstructed the database on the mainframe using Data Definition Language automatically produced by the database designer. Then IBM's Distributed Database Connection Services/2 (DDCS/2) product was used to give users access to the data on the mainframe with no changes to the VisPro/REXX applications. Hence, Cajun's mainframe is an essential component of their client/server strategy.

Looking Ahead

In addition to using VisPro/REXX to develop databases, Babcock's team is finding the product useful for developing all sorts of applications. One of the projects the team has used VisPro/REXX for includes a document imaging application

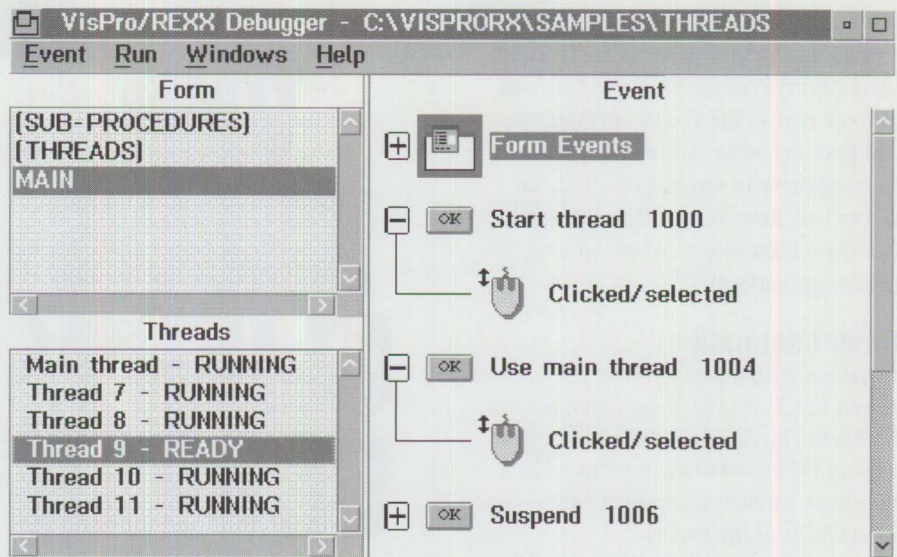


Figure 2. Debugging a Multithreaded Application with VisPro/REXX's Graphical Debugger

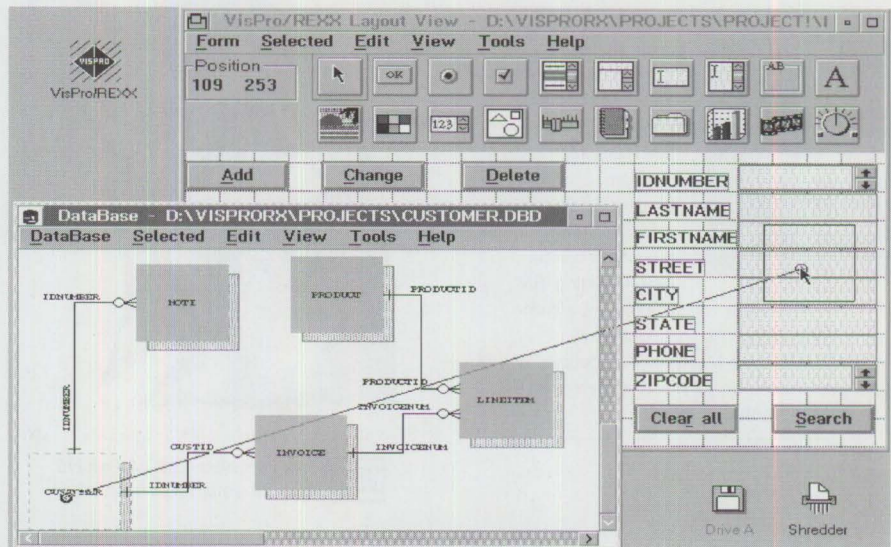


Figure 3. Drag-and-Drop Programming with VisPro/REXX's Database Designer

to reduce paperwork and improve retrievability for purchase orders and accounts receivables. They've also created an automated records information management system to classify and track documents that are moved along a life cycle from hard copy to microfiche to finally being destroyed. The team also developed an executive information system application for the company's managers who run OS/2 on ThinkPads; the project was completed in a scant two months.

Babcock's department doesn't market any of its applications since they're customized to fit Cajun's environment, but as a consultant outside of his job at Cajun, Babcock hopes to market a real estate

application he is building with VisPro/REXX. If he does, it won't be the only VisPro/REXX-built product on the market. Two OS/2 products built with VisPro/REXX are currently commercially available. Client Server Networking is currently marketing Watchit, a LAN server performance analysis and tuning tool. Multitask Consulting is marketing CS-Edit/2, a CONFIG.SYS editor for OS/2 that maintains, updates, and backs up the OS/2 CONFIG.SYS file, while shielding the user from all its complexities.

Babcock has great expectations of HockWare's upcoming offerings for other languages. VisPro/C and VisPro/C++ will offer an almost identical user interface to

VisPro/REXX. VisPro/C++ will include extensive support for IBM's User Interface Class Library. VisPro/REXX users, as well as those new to the VisPro environment, will find themselves scaling the C++ learning curve in record time. With all this to look forward to, Cajun can expect Babcock's team to serve them up even hotter applications!

Order Information

To get more information about VisPro/REXX or to place an order, call HockWare, Inc. at (919) 380-0616 or send a fax to (919) 380-0757. You can reach HockWare through CompuServe at 71333,3226 or Internet at hockware@vnet.net. HockWare's mailing address is P.O. Box 336, Cary, NC 27512-0336.



Judy Hock is an independent technical writer specializing in OS/2 and communications software. She has produced documentation for a number of companies, including IBM and HockWare, Inc. She has a Bachelor of Arts degree from the University of Windsor and a Master of Arts degree from Tulane University. She currently lives in North Carolina.

VNR KNOWS CLIENT/ SERVER

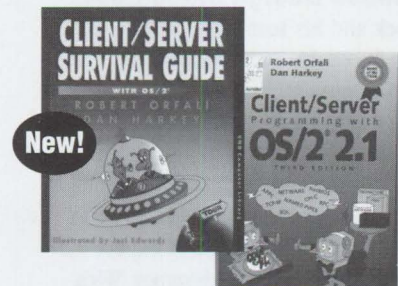


Ask for these and other VNR titles at your local bookstore.

Van Nostrand Reinhold

115 Fifth Avenue, New York, New York 10003
1-800-544-0550 73373.64 @compuserve.com

Read these two important books from bestselling authors Robert Orfali and Dan Harkey...



Client/Server Survival Guide with OS/2®

Provides a sweeping tour of client/server and distributed objects. An easy-to-follow guide that comprehensively reviews the technology, standards and over fifty commercial OS/2 client/server products. Offers 969 pages of essential information plus 400 illustrations.
\$39.95 0-442-01798-7
IBM #SR28-5494

Client/Server Programming with OS/2® 2.1, 3/E

The Second Edition won "The OS/2® Book of the Year" from OS/2® MONTHLY. Readers said individual chapters were, by themselves, *worth the price of the book*. And they're right! This new 1,142-page edition is a *must have* for all client/server and OS/2 programmers.
\$39.95 0-442-01833-9
IBM #G325-0650-02

OS/2® is a registered trademark of the IBM Corporation.

1644

Please circle #11 on reader service card.

TOMORROW'S NETWORKING TODAY . . . from IBM's PSCC (Personal Systems Competency Center)

"My technical expertise and background in OS/2 LAN Server enable me to quickly and accurately answer your questions—both simple and complex."



Saleem Akhtar



"I Help CUSTOMERS Find LAN Solutions"

Saleem holds an MS in Computer Science and has worked with OS/2 LAN Server for the last five years. He is part of our group of technical experts who support enterprise customers. If

your business environment has a mix of products (OS/2 DOS, LAN Server, NetWare, or others), then the PSCC has the expertise to help you. We provide consulting services through

Personal Systems Support Family's Consult Line on installation, migration, performance tuning, on-site assistance, application development/debug--and more.

For more information, call the PSCC at (800) 547-1283, ext. 504

Application Development by Program Integration

Consider a typical business application to manage employee records. On startup, the application connects to the company database and displays a list of employees along with key employee information, such as hire dates, departments, and salaries. You can browse the list and sort the information on any column. You can view graphs based on the information, such as a chart of department salary expenses. You can also change the information, after which any open charts will be updated to reflect the change and any correspondence required by the change will be generated. For example, if you remove employees from the list, termination notices will be generated. Now consider building such an application. Rapidly. In days. How would you do it?

Instead of building an application, you might be tempted to use a spreadsheet and word processor along with a pad of sticky notes. You write down the commands required to run the programs on the sticky notes, then you stick them to the side of your computer screen! (This is a popular technique judging by the number of computer screens adorned with sticky notes.) While this is an excellent solution for infrequently used procedures, you may require a more automatic solution if you plan to repeat the same procedure many times.

An automatic solution would reduce the training and effort required to run the procedure (and make space on the screen for a new set of sticky notes). In the past, automating such a procedure meant building a completely new program. Unfortunately, building a robust application by integrating programs such as word processors and spreadsheets was not possible. The available operating systems were unable to run multiple programs at the same time, the programs themselves lacked a way to be controlled from other programs, and powerful integrating tools

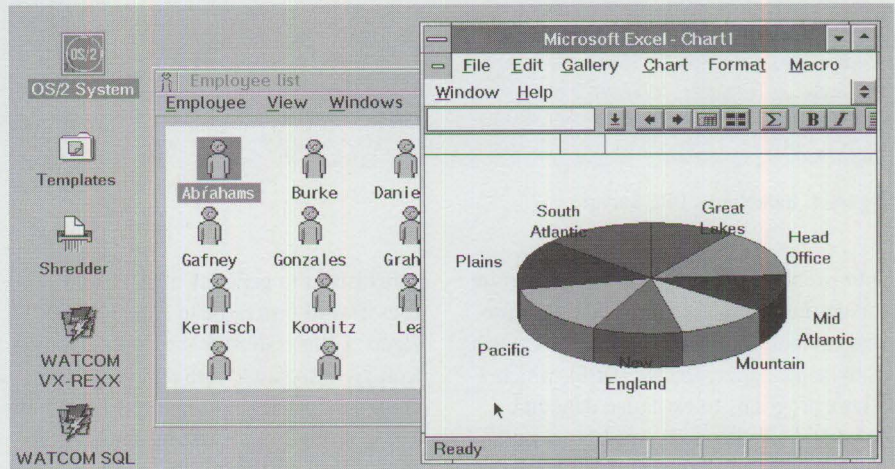


Figure 1. Employee List in Icon View

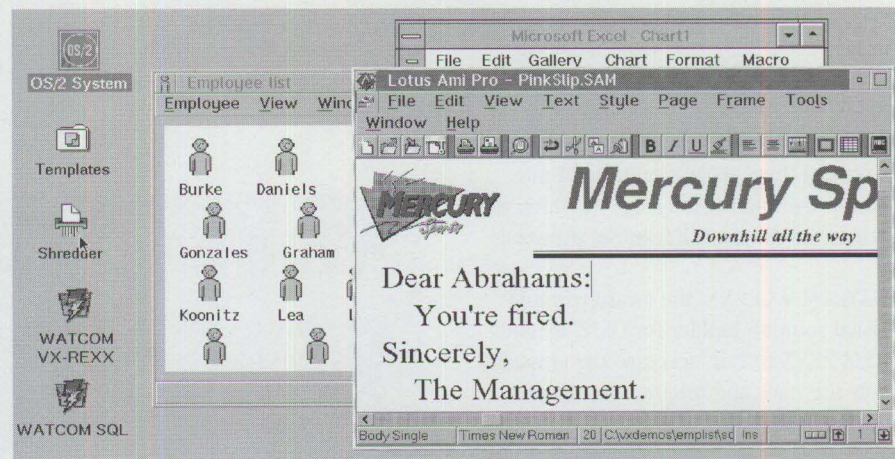


Figure 2. Automatically Generated Letters

```
call SysCreateObject "WPPProgram", "Ami Pro", "<WP_NOWHERE>".,
"OBJECTID=<_TMP_>;EXENAME=E:\AMIPROW\AMIPRO.EXE",,
"REPLACE"
call SysSetObjectData "<_TMP_>". "OPEN=DEFAULT"
```

Figure 3. Sample Code to Run Ami Pro

in which to write the integrating code were not available.

Integration Tools

However, the technology is now available that makes it practical to build programs

by integrating programs. The OS/2 operating system provides 32-bit power that lets you run many programs at the same time. OS/2 also allows you to run OS/2, DOS, and Windows applications, providing a large selection of programs to help solve

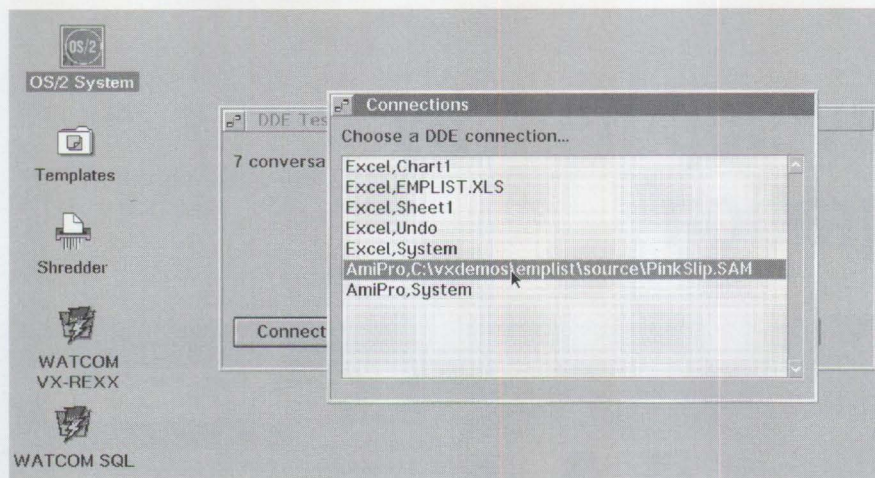


Figure 4. Selecting a DDE Server

your problem. Also important for program integration is OS/2's support for interprocess communication through features such as dynamic data exchange (DDE), which allows programs to exchange data and commands. Today most word processors, spreadsheets, and other business programs can be controlled from another program via DDE.

OS/2 also includes REXX, a powerful but easy-to-use system command language. Like DOS batch files, you can use OS/2 REXX to run programs automatically. Unlike DOS batch files, OS/2 REXX is a powerful third generation general purpose programming language in which you can create robust, full-featured programs.

WATCOM VX-REXX, the award-winning visual solution builder for OS/2, extends OS/2 REXX to fully leverage the graphical user interface and program integration features of OS/2.

Programming Techniques

The rest of this article looks at some of the techniques used to integrate programs, specifically techniques for running and controlling other programs. WATCOM provides a technical note where these techniques and others are discussed in more detail. The technical note includes the full source code for a sample employee list application similar to the application described at the beginning of this article. See the "Code Availability" section at the end of this article for information on obtaining the technical note.

The sample employee list application displays a list of employees, uses a

spreadsheet to generate a chart, and uses a word processor to create letters. Figure 1 illustrates the employee list in icon view, together with a chart of these employees' salary information. The application created this chart by running the spreadsheet and then sending the spreadsheet commands and data via DDE. Salary expenses can be adjusted by dragging employees to the OS/2 shredder. This causes a word processor to be launched and commands are sent to it that generate a letter to the terminated employee (Figure 2).

Running Programs

A program can be launched from your application in two ways. The first way uses REXX's native ability to directly execute OS/2 commands as if they were entered on a command line. You can use the OS/2 START command directly in your application to run another program in a separate process. In many cases, this is a satisfactory solution; however, this technique provides you with little control over the session used for the new process. Session control is especially important for DOS and Windows applications since you sometimes need to specify special DOS or WIN-OS/2 settings for these programs to work.

The second method lets you to control the program's session settings. To control your program using the second method, use the SysCreateObject REXX utility function to temporarily create a program reference object (class "WPPProgram") containing the required session settings. Then open the default view of that object. This is equivalent to double clicking on the program icon on the Workplace Shell. As an example, the code in Figure 3 might be used to run the program "Ami Pro for Windows."

More about VX-REXX

WATCOM VX-REXX is a robust multi-purpose development tool for creating applications that leverage the capabilities of OS/2 2.x and exploit OS/2's CUA '91 graphical user interface. VX-REXX combines a project management facility, visual designer, and an interactive source-level debugger to deliver a very approachable and highly productive visual development environment.

With the visual designer, you can create an attractive graphical interface using VX-REXX's full suite of CUA '91 interface objects, quickly customize their properties, and easily attach REXX procedures to the objects.

Drag-and-drop programming lets you write event routines without typing

any code. You can build, test, and debug your application without leaving the development environment.

Then you can package your application as an .EXE file or PM macro for royalty-free redistribution.

WATCOM VX-REXX was recently awarded *PC Magazine's* 1994 Editors' Choice for Cross-Application Macro Languages, *Software Development's* Productivity Award 1993, and *OS/2 Professional's* 1993 Best New Development Tool.

VX-REXX also comes with a 60-day money back guarantee. For more information about WATCOM VX-REXX, call (800) 265-4555.

The online REXX information supplied with OS/2 provides more information on the SysCreateObject and SysSetObjectData functions. More information on the strings used for session settings is available in *The REXX Reference Summary Handbook* by Dick Goran (available from C.F.S. Nevada, Inc. [702] 732-9616).

Once a program launches, you will also want to control its initial window size and position. As many programs do not provide this function, VX-REXX includes methods to find and control windows. For example, if the variable hWnd held the handle for the Ami Pro window, the following one-line command could be used to set its size and position:

```
call VRSet hWnd, "Left", left,
    "Top", top, "Height", height,
    "Width", width
```

Controlling Programs

A running program must receive commands and data to perform the necessary services for your application. One technique used to control a program's services is to send virtual keystrokes to the running program. This technique is equivalent to using the keyboard while working in the program. For example, the Alt+F4 keystroke sequence can be used to exit most OS/2 applications. The VX-REXX SendKeyString method can be used to send keystrokes to a specific window or to the window that currently has the focus. For example, the following one-line command sends the Alt+F4 keystrokes to the current focus window:

```
call VRMethod "Application",
    "SendKeyString", "",
    "{Alt}{F4}"
```

WATCOM VX-REXX includes a sample application called "Window Controller" that helps you explore the technique of sending keystrokes to OS/2 Presentation Manager (PM) windows.

Although the concept of sending keystrokes appears simple, you may find it difficult to build a robust application using this technique. One problem is that before keystrokes can be sent to a dialog, the dialog must be loaded. In many

systems, using a fixed delay in the application (during which it is hoped the dialog will appear) handles this problem. VX-REXX helps you solve the problem more accurately by enabling your application to wait for a window with a specific caption to appear. Even so, your error handling is ultimately limited because you cannot tell within your program if the keystrokes you sent had the desired effect.

A more robust technique is to send commands and data via DDE. This method is equivalent to executing macro commands within the program. The sample employee list application uses this technique primarily. The alternative method of sending keystrokes is used only when no DDE command is available.

DDE adheres to a client/server architecture. WATCOM VX-REXX provides a DDE client object, while the program that you wish to command acts as the server. The DDE client object in VX-REXX includes methods for listing all DDE servers and for connecting to a specific server. Once connected, you can send commands to the server and send and receive data from the server. Sending commands to the server rather than sending keystrokes allows you to track errors, since you will get a return value indicating if a DDE command failed.

The nature and format of DDE commands depends on the server. For example, in Lotus 1-2-3 for OS/2, all macro commands are available as DDE commands. If the DDE client DDE_1 connects to a 1-2-3 spreadsheet, the following one-line command can be used to chart the currently selected data:

```
ok = VRMethod( "DDE_1",
    "Execute", "/GV" )
```

To draw the salary chart, the application makes several DDE calls.

VX-REXX includes a sample program, DDE Explorer, with which you can connect to any OS/2 or Windows DDE server to test sending DDE commands. Figure 4 shows the process of selecting a DDE server in the DDE Explorer.

Code Availability

On CompuServe, WATCOM provides more information on writing VX-REXX applications to launch and control other programs. The VXT06.TXT file in the WATCOM VX-REXX section (section 16) of the Powersoft forum ([go watcom](http://go.watcom)) contains documentation and a sample program similar to the employee list program described in this article. The program can be configured to read the employee list either from the sample database included with DATABASE 2 OS/2 (DB2/2) or from a text file included with the program. The supported spreadsheet programs are Lotus 1-2-3 for OS/2 and Microsoft Excel for Windows. The supported word processors are Lotus Ami Pro for OS/2, Lotus Ami Pro for Windows, DeScribe for OS/2, and Microsoft Word for Windows.



Rob Veitch is a technical product manager at Watcom in Waterloo, Ontario, Canada. He has been involved in the development of a wide range of software including embedded systems,

interpreters, and local area networks. Since 1990 he has been working in visual development environments and is the lead architect of WATCOM VX-REXX. He holds a BAsC degree in Electrical Engineering from the University of Waterloo and is a member of the Association of Professional Engineers of Ontario.



Grace Welch is a marketing communications specialist for Watcom, where she works closely with the C, C++, FORTRAN 77, WATCOM SQL, and VX-REXX product lines for

OS/2. She joined Watcom in 1992 after receiving a BA degree from the University of Waterloo.

IBM REXX for NetWare

IBM recently announced its REXX language for Novell's NetWare operating system. For those who know both REXX and NetWare, this is a natural fit. Since NetWare lacks an easy-to-use interpreted language for creating small tools and utilities, REXX is ideal.

IBM REXX for NetWare is a port of OS/2 REXX. It has the same level of systems application architecture (SAA) compliance and nearly all of the same built-in functions as OS/2 REXX. The REXX application programming interfaces (APIs) for Exit handling, SubCommand environments, Macrospace, and External Function Registration were mapped one for one into the NetWare version. The APIs for Variable Pool, Halt, and Tracing were also ported from OS/2 REXX to REXX for NetWare. This means that IBM's REXX for NetWare is not only a highly versatile and rich language, but it is also easily customized and can be readily extended to meet many needs.

Rich in Function

Several NetWare Loadable Modules (NLMs) are shipped with the product; however, the three most significant are REXXLIB.NLM, REXX.NLM, and REXXUTIL.NLM.

- REXXLIB.NLM is essentially the interpreter. It also contains the REXX APIs and manages the global data shared by all executing REXX programs. It is automatically loaded the first time a REXX program is executed and stays resident until manually unloaded.

- REXX.NLM allows a user to load and run REXX programs from the system console as if each REXX program were an individual NLM.
- REXXUTIL.NLM contains several functions specific to NetWare that enhance the language's usefulness.

To improve initial startup performance, both REXXLIB.NLM and REXXUTIL.NLM can be loaded from the AUTOEXEC.NCF file at server startup time.

Once loaded, REXXLIB.NLM consumes approximately 300 KB of server memory. Each executing REXX program also consumes storage. The amount consumed depends upon REXX program size, REXX variable usage, internal REXX queue usage, etc. There are no limitations to the amount of storage that REXX can consume, and it is up to the REXX programmer to understand the resource limitations of a particular server and the demands a REXX program makes on these resources.

Easy to Learn

In addition to the above mentioned NLMs, a number of C and REXX program examples are shipped with IBM REXX for NetWare; most notable of these is REXXTRY.REX. It provides a method by which both novice and experienced REXX programmers can experiment with REXX without developing complete programs. REXXTRY.REX also provides a pseudo-shell from which REXX statements are issued from the console as if from within

a REXX program, then the results are evaluated. While this program is included as an example, it clearly should not be overlooked as a learning tool and a development aid.

The list of simple REXX utilities includes such programs as copying files or issuing a series of console commands. REXX simplifies performing functions at specific times, such as loading NLMs or enabling/disabling logins, creating simple user interfaces, and prompting for input. REXX can even be extended or coupled with other applications by using the supplied APIs. Products that take advantage of REXX and its capabilities already exist.

Try and Buy

The bottom line is that IBM REXX for NetWare adds value and function to the NetWare operating system, and the 60-day evaluation period makes it worth looking into. REXX for NetWare is available from IBM's As-Is product group. Call (800) IBM-CALL (426-2255) for product information or to order REXX for NetWare (product number 5764-075).

References

- *IBM REXX for NetWare Reference* (SH24-5286)
- *The REXX Language* (ZB35-5100)

Rob Maiolini is a Senior Associate Programmer in IBM's REXX development group. He is currently responsible for REXX development for NetWare and AS/400 and has worked on VM operating system development in the past.

TOMORROW'S NETWORKING TODAY . . . from IBM's PSCC (Personal Systems Competency Center)

"I don't specialize in a product line; I specialize in a 'technology' line."



Randall Autry



"I Help CUSTOMERS DEVELOP"

A specialist in communications and application development with the PSCC, Randall spends his time helping customers with medium to large scale OS/2 applications involving NetBIOS,

TCP/IP, APPC, or LAN Server. Randall uses his expertise to support enterprise customers through Personal Systems Support Family services such as Customer Application Assis-

tance. If your business environment is mixed, then the PSCC has the expertise to help you. When you need help with application development, you'll know where to turn.

For more information, call the PSCC at (800) 547-1283, ext. 504

GammaTech REXX SuperSet/2—Give Your REXX Programs the Power of C

OS/2 REXX is a powerful tool for producing applications, macros, and prototypes. The language is made flexible and extendible by providing application programming interfaces (APIs) that allow REXX to interface with the developer's application. These interfaces are usually developed using a language such as C and thus require the developer to be proficient in that language.

One viable solution to this problem is GammaTech REXX SuperSet/2. SuperSet/2 is a set of external library functions for REXX that the developer uses to interface OS/2 REXX to a variety of products including IBM LAN Server, transmission control protocol/internet protocol (TCP/IP), NetBIOS, and others. It also can tap into many of OS/2's low level facilities such as processes, threads, semaphores, named pipes, VIO commands, and others.

GammaTech REXX SuperSet/2 has over 300 REXX external functions in seven dynamic link libraries (DLLs). The libraries are divided by product and interface type so that the developer needs to load only functions actually used. All of the DLLs are full 32-bit implementations.

How Can SuperSet/2 Help the Developer?

GammaTech REXX SuperSet/2 gives the developer a set of functions that eases REXX's integration into development projects. The function libraries create a tighter integration between REXX, OS/2, and other products, allowing REXX to take on a more functional and powerful role in a product. And because REXX is an interpreted language, the development cycle can be reduced since there is no need to compile the code in order to test the program.

The SuperSet/2 functions also improve the performance of REXX programs because many of the slower REXX facilities such as file input/output are

automated. This also simplifies a REXX program, making it easier to maintain.

Each DLL function has been designed to provide functionality that the developer would create if given the time and resources. SuperSet/2 gives OS/2 REXX the kind of functionality and interface capabilities usually reserved for languages such as C. To further help developers, each SuperSet/2 package contains a 650-page manual, an online reference that fully documents each function with examples, and over 25 sample REXX programs, which utilize most of the functions.

What Kinds of Functions Are Available?

The GammaTech REXX SuperSet/2 provides a wide range of functions that perform simple and complex operations for the REXX developer. For example, SuperSet/2 includes interfaces to many of OS/2's low level capabilities:

- File and system functions including EXECIO (24 functions)

- Semaphores (20 functions)
- Named pipes (8 functions)
- VIO commands (22 functions)
- Process and thread manipulation (5 functions)
- Extensive math library (21 functions)
- Date manipulation (2 functions)
- REXX macrospace manipulation (7 functions)

SuperSet/2 also provides functions that integrate REXX into other products such as IBM LAN Server, TCP/IP, NetBIOS, User Profile Management, and Communications Manager/2 emulator high level language application programming interface (EHLAPI).

The IBM LAN Server functions fall into the following general categories:

- Protection (27 functions)
- Server administration (18 functions)
- Service and configuration (6 functions)

```

/* register all the GRXLSNET functions */
call RxFuncAdd 'GrxLSNetLoadFuncs', 'GRXLSNET', 'GRXLSNETLOADFUNCS'
call GrxLSNetLoadFuncs

/* Enumerate all the local LAN Server services */
level = 2 /* this determines the level of detail returned */
rc = GrxNetServiceEnum('', level, 'srvcs.')
if rc = 0 then do
  say 'Services enumerated successfully.'
  say 'The list of service names and information follow:'
  do i = 1 to srvcs.0
    say 'NAME =' srvcs.i._name
    if level = 1 | level = 2 then do
      say 'STATUS =' srvcs.i._status
      say 'CODE =' srvcs.i._code
      say 'PID =' srvcs.i._pid
    end
    if level = 2 then say 'TEXT =' srvcs.i._text
  end
end
else say 'Services enumeration error. RC was' rc
return 0

```

Figure 1. Program to Show Currently Running LAN Server Services


```

/* register our TSO subcommand environment DLL */
address CMD 'rxsubcom query TSO grxhost'
if rc = 30 then do
  address CMD 'rxsubcom register TSO grxhost GrxHost'
  if rc <> 0 then do
    say 'Unable to register our subcommand environment DLL.'
    say 'The return code from RXSUBCOM was' rc
    exit rc
  end
end

say 'Reading the C:\CONFIG.SYS file into the FILE stem variable.'
mycmd = 'EXECIO * DISKR c:\config.sys ( STEM file. FINIS'
address TSO mycmd
say 'The return code from the subcommand was' rc
if rc <> 0 then return 0

say 'Writing file TEMP.TMP from the stem variable.'
mycmd = 'EXECIO 'file.0' DISKW temp.tmp ( STEM file. FINIS'
address TSO mycmd
say 'The return code from the subcommand was' rc
if rc <> 0 then return 0

say 'Reading the C:\CONFIG.SYS file into the REXX queue.'
mycmd = 'EXECIO * DISKR c:\config.sys ( FINIS'
address TSO mycmd
say 'The return code from the subcommand was' rc
if rc <> 0 then return 0

say 'Writing file TEMP1.TMP from the REXX queue.'
mycmd = 'EXECIO 'queued()' DISKW temp1.tmp ( FINIS'
address TSO mycmd
say 'The return code from the subcommand was' rc

return 0

```

Figure 2. Porting REXX Programs from TSO and CMS to OS/2

■ Workstation (20 functions)

The TCP/IP functions fall into the following general categories:

■ File Transfer Protocol (FTP) (18 functions)

■ Sockets (21 functions)

The NetBIOS interface includes 20 functions that can be used to create a complete NetBIOS application using

REXX. These include Send, Receive, Add name, Hangup, Reset, and many others.

The Communications Manager/2 EHLAPI functions provide a complete interface for controlling 3270 and 5250 sessions.

All of the functions can be used in any OS/2 session including full-screen, windowed and Presentation Manager programs.

What Kind of Programs Can Be Created Using SuperSet/2?

Figure 1 shows the power of GammaTech REXX SuperSet/2. The program enumerates all of the currently running LAN Server services at the local server/workstation, places the information into a REXX stem (or compound) variable, and then displays that information.

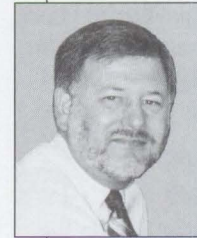
Figure 2 shows how to port REXX programs from the TSO and CMS environments to OS/2 using SuperSet/2. This has not been possible in the past because many of the programs utilize the EXECIO facility, which is not supported in OS/2. Since SuperSet/2 supports this facility, porting these programs is much easier.

Conclusion

GammaTech REXX SuperSet/2 provides complete integration solutions between REXX and other OS/2 products. SuperSet/2 will reduce and simplify your REXX programs and provide solutions to problems that could only be achieved in the past using a language such as C.

For more information on GammaTech REXX SuperSet/2, contact:

SoftTouch Systems, Inc.
Workstation Division
1300 S. Meridian
Oklahoma City, OK 73108-1751
Ph. (405) 947-8080
Fax (405) 632-6537



David Ashley currently works for a large transportation company as a Senior Programmer/Analyst. He also consults part time and is the author of the GammaTech REXX SuperSet/2 product. He has a BS in Architecture from the University of Texas at Arlington.

TOMORROW'S NETWORKING TODAY . . . from IBM's PSCC (Personal Systems Competency Center)

"A positive attitude and well-honed skills make a winning support team."



Kevin Alexander



"I Do NETWORKS"

Kevin's 20 years' experience in the computer industry makes him a natural—a natural technical lead for Q&A support of Novell NetWare from IBM, that is. Kevin also has experi-

ence in design of LANs, WANs, and encrypted data voice lines. He lends his expertise to the PSCC through the Personal Systems Support Family Consult Line Offering. Consult with

Kevin, and other PSCC technical specialists like him, on topics such as systems integration and management, LAN planning, performance, communications, and capacity planning.

For more information, call the PSCC at (800) 547-1283, ext. 504

BranchCard: A Viable Option to Stand-Alone Hubs

Is your token-ring suffering from faults and beacons? Are your network devices trying to run at the wrong speed? Do you need to connect your branch offices' computing activities to your main location? And, even if you accomplish all this, will the system be manageable?

BranchCard from LanOptics Inc. is the answer to all these questions, plus it is cost effective and easy to use! BranchCard is an intelligent hub built on a standard computer bus, either ISA (industry standard architecture) or Micro Channel architecture.

For more effective network management, you can use an intelligent hub. An *intelligent hub* is a device that allows you to connect various types of cable media and to incorporate electronic circuitry or modules for many different functions, such as signal retiming/regeneration and monitoring transmissions.

Hub card technology is not new; in fact it has been around since late 1991. Until now, though, it has not really caught on in the local area networking arena for a variety of reasons:

- Unwillingness to put hub technology in the file server
- Lack of effective management capabilities
- Lack of inter-networking products that work with hub cards
- Limited number of vendors offering hub technology

However, given the current state of the economy and reduced information systems (IS) budgets, LanOptics' BranchCard is now a viable option.

The Answers

The most prominent problems in a token-ring network are beacons or other token-ring faults. *Beacons* are frames or messages repeatedly transmitted by a station

after detecting a line break or outage. Transmission of beacon messages stops when the fault is bypassed or eliminated. The **Fault Watch** feature, built into the hardware of the BranchCard, protects your network from going down in the event of a beaconing workstation.

Another common problem in today's networks results from devices trying to run on the token ring at the wrong speed. Many token-ring network interface card (NIC) vendors ship their NICs factory-set to 16 Mbps token-ring operation (some still ship at 4 Mbps). If your installation personnel miss setting a NIC to the correct speed when pre-configuring a user's workstation, then add that workstation to the network, the LAN will go down.

Speed Watch is another BranchCard built-in feature that constantly checks each port for proper operating speed. If a new workstation tries to enter the ring at the wrong speed, Speed Watch ensures that port is never allowed on the ring, thereby preventing that workstation from causing a network failure. Speed Watch and Fault Watch constantly check all ports (enabled or disabled) to verify whether or not they can be used. Once the error has been corrected, the port is enabled.

These two features alone will save many hours of potential down-time plus reduce troubleshooting time. If a network management station is monitoring the network, it will receive an alert warning the operations staff of a problem. This allows the operations staff to take a proactive management stance, taking care of the problem before it can cause an outage. Specific problems at remote sites can also be solved without having to dispatch remote support personnel, thus reducing user down-time.

Today's diversified business environment often requires that many small branch offices be connected to a central computing site. Modern client/server technology

has provided many options: stand-alone routers or bridges, gateway boards in the file server or as a stand-alone computer, and even router or modem boards in the file server.

You may want to consider a hub card that works in parallel with some type of wide-area connectivity board, such as a router board. This gives you remote connectivity functions while still maintaining that small physical space and keeping all the computing connectivity in a single computer.

The BranchCard is fully compatible with IBM's RouteXpander, an OS/2-based multi-protocol router, plus it offers an optional Cisco 2500 series-compatible router card. Using the BranchCard in combination with these routing packages offers not only branch office connectivity but also offers wide-area network/routing technology with a wide range of protocol support. The entire system can be managed from the central site.

Network Management Options

You will want hub card technology to be manageable, usually through a simple network management protocol (SNMP) management console somewhere within the network. You can manage hub cards using a UNIX-based management console such as NetView/6000 from IBM, SunNet Manager from SunConnect, HP OpenView from Hewlett-Packard, or even DOS/Windows-based consoles such as NetWare Management System from Novell or HP OpenView/Windows.

If you use the BranchCard on a NetWare file server platform, Novell's Hub Management Interface (HMI) specification support allows you to manage the BranchCard using NetWare's HubCon NetWare Loadable Module (NLM). LanOptics' StarView/DOS (for HP OpenView/Windows) and StarView/6000 (for NetView/6000) management applications can also manage the BranchCard.

The BranchCard solution fits well in a systems network architecture (SNA) network, usually with NetView managing the host computer communications side of the system. The BranchCard is an excellent fit for this environment because it provides a bi-directional NetView management agent. You get the same control and monitoring capability that is available in an SNMP management console with the standard Network Management Vector Table (NMVT) NetView command set on a NetView console.

Solutions You Can Afford

The BranchCard offers a unique solution for the small office needing an affordable local area network (LAN) hub. Since the BranchCard uses a standard ISA or Micro Channel slot, a logical choice for installation is in the file server. Although some hub cards can be installed in another computer on the network, some type of re-routing agent is needed to manage the hub card. When the BranchCard is installed in the file server, there is usually a management agent that also runs on that network operating system (NOS), such as IBM's OS/2 or Novell's NetWare. When the BranchCard is installed in a computer that is not the file server, it only needs to run the management agent, which is available in OS/2, DOS, and NetWare formats.

The BranchCard offers up to eight ports for token-ring or Ethernet connectivity. This puts the BranchCard at a lower price-per-port than most hub products, from stackable to enterprise hubs. Since many branch offices have fewer than 16 devices to be connected, you can gain a better economy of scale using the BranchCard.

By installing the BranchCard in the file server, you have one less external device

Scenario

The Situation: You have a branch office with six 80386, OS/2-based PCs, a file server running LAN Manager, 16 Mbps token-ring topology, and a 56 Kbps leased-line circuit to your company's central computing site serving an SNA network connection. You also have Communications Manager/2 loaded on the file server, providing 3270 gateway and terminal emulation services for your branch office users.

You want to connect the six PC workstations and the file server on the token ring, provide local (branch office) management, and manage the hub back at the central site.

The Solution: Install a LanOptics Token-Ring BranchCard in the file

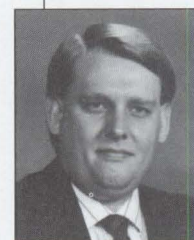
server, OS/2 and NetView management agents at the central site to manage the BranchCard, and LanOptics NetView Interface Agent on the file server to communicate with the NetView console. With these components, you will have a cost-effective network solution to provide local branch office management as well as central site management.

For more information about LanOptics products, call or write:

LanOptics Inc.
13748 Neutron Road
Dallas, TX 75244
Voice (214) 392-0647
Fax (214) 385-0723

to worry about (space, power, etc.), allowing for easier pre-configuration before shipping the system to the remote office for installation. And because the BranchCard allows you to have multiple card networks (maximum of four cards), you can easily expand the eight ports to 32.

Whether you are trying to connect a few workstations to form a token-ring or Ethernet LAN, or you are connecting that small LAN back to a central site for host communications, the BranchCard is your answer. Add the ability to manage that remote site from the central site on any network management platform, and you have the best solution to fit the overall project requirements.



Jeffrey L. Carrell, a Senior Systems Engineer for LanOptics Inc., is responsible for designing enterprise and branch office networking solutions for Fortune 500 companies.

Jeff has also worked for American Airlines, designing a network management system for over 21,000 LANs, and for Novell as a technical instructor and systems engineer. He is also a frequent industry speaker and has published numerous articles and technical reviews.

Jeff can be reached at LanOptics Inc., at (214) 392-0647 or on the Internet: jcarrell@metronet.com.

Clarification of a Figure from Last Issue's "Architecture Soup"

An illustration appearing in last issue's "Architecture Soup" article gave some readers an incorrect impression. Figure 7 was intended to point out that OS/2 is a 32-bit

multitasking operating system that is currently available, mature, and robust—while NT, Chicago, and UNIX are shown as potential future factors.

A Hands-On Primer for REXX

This article is a tutorial about using REXX, the interpretive language that gives the OS/2 user more control over the OS/2 environment and its applications. The tutorial covers extensions, data types, operations on the data, assignments and console displays, getting and using input, using functions, repetitive operations, libraries of extensions, writing REXX programs, and graphical REXX environments.

What is the big deal about the REXX language, and why should you learn it?

REXX gives you a very powerful tool for writing applications as well as controlling applications such as Communications Manager/2, DATABASE 2 OS/2 (DB2/2), and the Workplace Shell. REXX is also the language used for configuration, installation, and distribution (CID) installation using OS/2. A solid knowledge of REXX gives you the ability to gain more complete control over the OS/2 environment and many of its applications.

REXX is simple to learn and tinker with because it is an interpretive language. This means you can try things out in REXX and immediately see what they do. If you are not happy with the results, you can change the source code and try the program again. This fast, iterative process helps you create prototype applications in a fraction of the time required for development environments such as C and COBOL.

REXX Caveats

There are limitations to the REXX language, however. REXX programs do not run as quickly as programs produced with compiled languages. Compiled languages such as C and COBOL are faster because an interpretive layer of code is not sitting between the programs and the CPU. Compiled languages produce optimized machine code that can be directly executed by the CPU.

Philip Lieberman
Lieberman and Associates
Beverly Hills, California

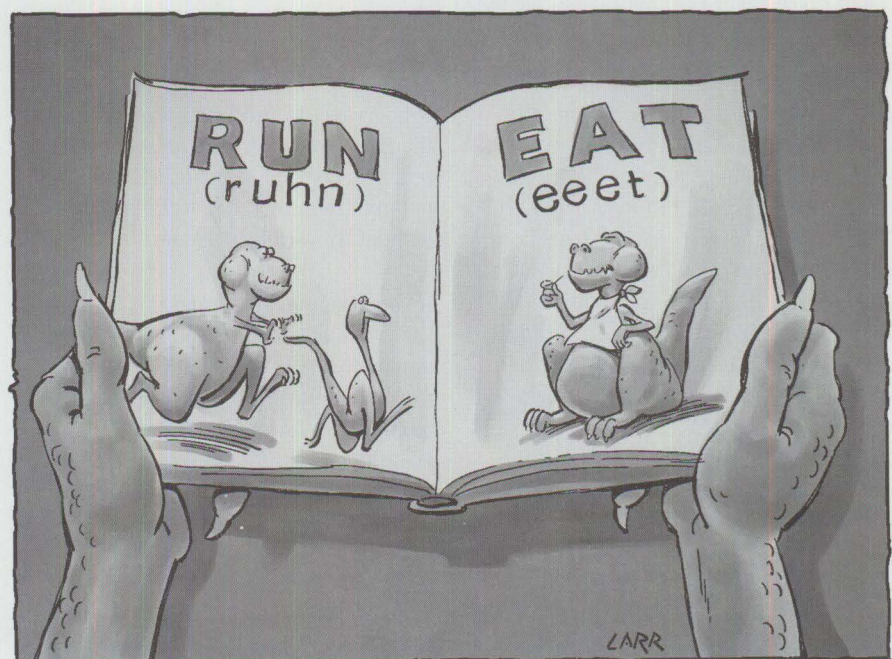
Another limitation of REXX is that it represents all types of data as strings. Numbers, booleans, and (of course) strings are represented as strings of characters. The REXX

language has all of the standard math functions, as well as program control elements (loops, if/then, do/while, and more).

REXX Extensions

Even though REXX has some speed and data-type limitations, these limitations can be overcome by writing extensions to the REXX language in a compiled language such as C. An extension is a function that can be called by REXX but normally lives in a dynamic link library (DLL). The functions in the DLL are usually written in the C language for speed and flexibility. When REXX calls a function in an extension, the function in the library converts the passed REXX parameters from strings to native data types for C. The library function then operates using the converted C native data types. Finally, the function converts the results back to a string or strings so that REXX can use the result.

By judiciously using REXX extensions, you can keep REXX's flexibility and use compiled code only for those sections of the program that need speed or access to




```

<F3> Recall the last command / restore the remaining keyboard buffer

<Del> Delete forward

<Ins> Turn on/off character insertion mode (to add characters)

-> <- Arrow keys control cursor position

```

Figure 1. REXX Editing Keys

Arithmetic operators (used on numeric strings):

```

+ add
- subtract
* multiply
/ divide
% integer divide (return only the whole number part)
// remainder (return only the fractional part)
** power (raise the number to a whole-number power)

```

String operators (used on character strings):

```

(space) concatenate with one space character
|| concatenate without any spaces
(abuttal) concatenate without any spaces

```

Note: Attempting to perform addition within a literal string such as 'Fred'+123' will cause an execution error. If you want to put two strings together, you can put them next to each other; or if you don't want any spaces, you can use the concatenation operator, such as in 'Fred||123', to get 'Fred123'.

Figure 2. Arithmetic Operators and String Operators

resources not native to REXX. A lot of extension libraries exist for things such as floating-point math, advanced program-to-program communication (APPC), transmission control protocol/internet protocol (TCP/IP), Communications Manager/2, DB2/2, and even the OS/2 Workplace Shell. Many of the extension libraries are free and available either with the product or from bulletin boards that specialize in OS/2. If you want to write your own extensions to the REXX language, the OS/2 2.1 toolkit contains sample code that demonstrates the interface and compilation process.

Full 32-Bit Implementation

Full 32-bit implementation is another powerful REXX feature. This means that

REXX applications are effectively not memory-constrained. Because REXX uses the CPU's 32-bit instruction set and registers, it is fast and compact. REXX's 32-bit nature also means that it can be extended without concern for memory.

If you decide to go beyond this tutorial and use some of the graphical REXX environments mentioned earlier, you will find full multithreading as well as access to OS/2's interprocess communication facilities. Many of these packages have already incorporated interfaces to DB/2 and multimedia.

The REXX Tutorial

The following tutorial requires that you run OS/2 2.1 and have REXX already

loaded. If REXX is not loaded, you can add it by running the Selective Install utility in the System Setup folder. (This assumes that you have access to the original OS/2 disks, CD-ROM, or network drive that contains the disk images.) If you are not sure REXX is installed in your computer, just try the tutorial—if it doesn't work, REXX is not installed.

Getting the Most from the Tutorial

You will see a check box next to each step. It is a good idea to check off each step as you complete it. This keeps you from missing steps. Since you will do a lot of typing in the following steps, you may want to use the editing keys in Figure 1 to make your experience easier.

Getting Started with REXXTRY

- 1. Open and maximize an OS/2 windowed session. You will work in this session.
- 2. Start the interactive REXX interpreter by typing:
REXXTRY <Enter>
- 3. You should see the following prompt:
REXXTRY lets you interactively try REXX statements. Each string is executed when you hit Enter. Enter "call tell" for a description of the features.

Simple Assignments and Display on Console

- 4. To send information to the console, use the SAY verb. Type:
SAY 1111*1111 <Enter>
- 5. REXX should respond with:
1234321
- 6. Next, you can assign values to variables and have REXX multiply those variables. Type:
VALUE1=1111 <Enter>
VALUE2=2222 <Enter>
SAY VALUE1+VALUE2 <Enter>
- 7. REXX should respond with:
3333
- 8. REXX is adept at handling both numeric and alphanumeric string expressions. Add a string to your previous SAY statement (strings are usually enclosed in single quotes) by typing:


```
SAY 'Two Values ='
VALUE1+VALUE2 <Enter>
```

9. REXX should respond with:

```
Two Values = 3333
```

10. You can even put a string into a variable and use it in the SAY statement. Type:

```
TWOVAL='Two Values =' <Enter>
SAY TWOVAL VALUE1+VALUE2 <Enter>
```

11. REXX should respond (again) with:

```
Two Values = 3333
```

Some simple rules of variables and operators:

- All variables are stored as strings.
- Character-string literals (e.g., 'Fred's house') should normally be enclosed in single quotes. If you need to use the single quote (') within a string, you can enclose the string with ditto marks (" ") instead (e.g., "Fred's house").
- Variable symbols themselves must be selected from the English alphabet (A-Z, a-z), numbers from 0-9, and from the . ! ? characters and the underscore. Symbol names may be up to 50 characters long.

The REXX arithmetic operators and string operators are listed in Figure 2.

Getting Input and Making Decisions With It

12. To get user input, use the PULL verb. Type:

```
SAY 'What is your age?';PULL
AGE;SAY 'You are' AGE 'years
young!'
```

13. REXX will pause with the message:

```
What is your age?
```

14. Type in your age and press Enter.

15. REXX will then respond with the message:

```
You are XX years young!
```

Note: In this last example, you used the PULL verb to get a variable's value. Also notice that you can put more than one operation on a line by using the semicolon as an operator.

16. To make decisions, you can use comparisons in your program to

Normal Comparison Operators:

(padding strings to identical length when needed)

=	equals
\=	not equal
>	greater than
<	less than
<>, ><	not equal
>=, \<	greater than or equal to (not less than)
<=, \>	less than or equal to (not greater than)

String Comparison Operators:

(exact character-for-character comparisons)

=	equals
\=	not equal
>>	greater than
<<	less than
>>=, \<<	greater than or equal to (not less than)
<<=, \>>	less than or equal to (not greater than)

Figure 3. Normal and String Comparison Operators

change program flow. Type the following to see if the age you just entered was greater than 21:

```
IF AGE > 21 then SAY 'You are
over 21' <Enter>
IF AGE < 21 THEN say 'You are
less than 21' <Enter>
```

17. Depending on the age you previously entered, one of the above statements will appear on the screen.

The normal comparison operators and string comparison operators in REXX are listed in Figure 3.

Repetitive Operations (Loops)

18. The simplest expression for looping is the DO loop. Type the following to repeat "Am I repeating myself?" ten times:

```
DO 10;SAY 'Am I repeating
myself?';END <Enter>
```

In a DO loop, everything between the DO and END is repeated according to the expression immediately after the DO. You can also have the DO loop increment (or decrement) a variable each time through the loop.

19. Type:

```
DO i=1 to 10; SAY "I'm now at:"
i; END <Enter>
```

20. You can even count down. Type

the following line (which is split for easier reading):

```
DO i=10 TO 1 BY -1;SAY
"Countdown:" i;END;
SAY "*Blastoff*" <Enter>
```

Using Functions

The REXX language has a rich variety of built-in functions (subroutines) for handling strings and input/output (I/O). The syntax of using functions can be a little confusing at times because of the way parentheses are used to enclose parameters for the functions.

21. If a function returns a result in a variable, use parentheses for the argument(s). Type:

```
RESULT = REVERSE('NATURES')
<Enter>
SAY RESULT <Enter>
```

22. The REVERSE function reverses all of the letters in the string. Type the following equivalent expression, where the function returns the string directly to the SAY verb:

```
SAY REVERSE('NATURES') <Enter>
```

But some functions do not return a result, or you may wish to simply throw away the result (like the screen-clearing example). In those cases, you call the function with a CALL verb and eliminate the parentheses.


```

/* REXX Program: TRY.COMD */
CALL RxFuncAdd 'SysLoadFuncs','RexxUtil','SysLoadFuncs'
CALL SysLoadFuncs

/* Load stem variable 'fileinfo' with directory */
CALL SysFileTree 'C:\*.*',fileinfo
DO i=1 to fileinfo.0
SAY fileinfo.i
END

```

Figure 4. REXX Program TRY.COMD

- 23. Type the following to use the CALL verb:

```
CALL LINEOUT 'CON','Sending
I/O to Console' <Enter>
```

Note: There are no parentheses enclosing the parameters because you are doing a subroutine call instead of a function call.

Note: When you use the SAY verb, you are really issuing a call to LINEOUT with the output going to the console (CON).

Arrays: Compound Symbols in REXX

In languages other than REXX, multi-dimensional variables are normally indicated by pairs of parentheses or square brackets to indicate dimension. For example, in the C language the variable matrix would be represented as:

```
matrix[x][y][z]
```

or in the BASIC language it would be represented as:

```
matrix(x,y,z)
```

But in REXX it is represented as:

```
matrix.x.y.z
```

Using the terminology of compound variables in REXX, the name *matrix* is the *stem* part of the variable. The *x.y.z* part of *matrix* is known as the *tail* of the variable. The zero entry in a particular dimension of stem variable indicates how many elements are in that dimension. For example, if you pass the single string variable name *files* to a directory-listing function, the function returns with the *files.0* entry indicating how many files are in the just-filled *files* variable, starting with *files.1* and ending with *files.(files.0)*.

Using REXXUTIL

The REXX language is designed to be completely independent of the platform

under which it runs. REXX runs in DOS, OS/2, OS/400, and various mainframe operating systems. However, each environment provides special features unique to the platform under examination. To support extensions of the REXX language, REXX allows libraries of extensions to be loaded.

Loading the REXXUTIL Library

If you start to use a function in the REXXUTIL library before the library has been loaded, the attempt will fail. However, on some systems, the library may be autoloading at bootup, so the following step may succeed! This will be the case if your system is running OS/2 2.0 plus the ServicePak.

- 24. Try out the screen-clear function in the REXXUTIL library. Type:

```
CALL SYSCLS <Enter>
```

- 25. REXX should respond with:

```
Oooooops! Try again. Routine
not found
```

```
rc=43 .....
REXXTRY.COMD on OS/2
```

The second line containing *rc=43* is a return code that indicates the success or failure of the last REXX statement. In the case shown, it means that no function called *SYSCLS* was known to REXX.

To add a library, you must first load it, then initialize it (make it ready to run). This method of adding extensions to REXX is the same for all extensions, except for the names of the added libraries and the initialization point names.

- 26. To add the REXXUTIL library, type:

```
CALL RxFuncAdd 'SysLoadFuncs',
'RexxUtil','SysLoadFuncs'
```

```
<Enter>
```

```
CALL SysLoadFuncs <Enter>
```

- 27. If you typed each of the lines correctly, no error codes will be returned. If you got an error code, just retype the command.

If you are using the OS/2 2.0 ServicePak, these two commands may cause an error.

You are probably wondering whether these functions must be typed exactly in upper- and lower-case as shown above. The answer in this example is no. For loading functions, the case of parameters is ignored. We used an upper/lower convention to make the strings easier to use.

- 28. Try out that screen-clear function again. Type:

```
CALL SYSCLS <Enter>
```

- 29. REXX should respond by clearing the screen.

Writing REXX Programs

At this point, you are probably becoming burned out typing these long strings, so let's move on to writing REXX files or programs.

REXX files are ordinary text files with a *.CMD* extension on the file name. As you already know, OS/2 batch files also have the *.CMD* extension. To differentiate between normal OS/2 command files and REXX files, REXX files always have a C-language-style comment line starting with */** and ending with **/* as their first line.

- 30. Exit the REXXTRY utility by typing:

```
EXIT <Enter>
```

- 31. Start the editor to create TRY.COMD by typing:

```
START E TRY.COMD <Enter>
```

- 32. Try typing in a program that combines stem variables and a function in REXXUTIL. The program in Figure 4 gets a complete list of files and subdirectories, starting at the path *C:*.**. The list of files is stored in the variable *fileinfo*. The stem variable is then printed out.

- 33. Once you have typed the program, use the FILE | Save command and save the file as a PLAIN TEXT document. Don't close the editor yet.

34. Switch back to the OS/2 window session, and start the recently written REXX program by typing:

TRY <Enter>

Note: If there are any errors in the program, just return to the editor, make changes, save the program, and go back to the OS/2 session to try it again.

Graphical REXX Environments

In this quick tutorial on REXX, you have seen many of the features of the language. REXX is very simple and easy to learn, yet is extraordinarily powerful as a result of the available extension libraries. You can use the built-in REXX shipped with OS/2 for controlling a bewildering variety of things, but much more is possible if you move on to the graphical REXX environment provided by third-party vendors.

I recommend that you try out such graphical environments as WATCOM VX-REXX, HockWare VisPro/REXX, and GPF Systems' GpfRexx. All of these environments use the base of knowledge that you have obtained through this tutorial, and they give you the ability to put together sophisticated graphical interfaces in a fraction of the time needed to do so in C or COBOL. Best of all, these packages produce code without run-time charges.

Acknowledgment

Many thanks to Rick Adamson of Learning Tree International for copyright permission to use parts of a REXX exercise from my four-day course titled *OS/2 2.x: A Comprehensive Hands-On Workshop*. For more information about this and other courses, contact Learning Tree International at (800) THE-TREE (843-8733).



Philip Lieberman is a well-known author, lecturer, and consultant in real-time systems and OS/2. He is the author of the Learning Tree course *OS/2 2.x: A Comprehensive*

Hands-On Workshop. Mr. Lieberman owns Lieberman and Associates, which produces the LAN Intensive Care Utilities (ICU) for IBM LAN Server software product. He is also a magician and a member of the Hollywood Magic Castle and the International Brotherhood of Magicians.

Lieberman and Associates
221 N. Robertson Blvd., Suite C
Beverly Hills, CA 90211
Voice (800) 829-6263 or (310) 550-8575
Fax (310) 550-1152
BBS (310) 550-5980
CompuServe 76426,363
IBMLink: DEV2203
PRODIGY: PKFJ82A

Correction

In the "DCE: An Application Primer" article in the July/August issue of *Personal Systems*, the phone number for the Lieberman and Associates Bulletin Board System (BBS) was incorrect. Please note the correct number for the BBS: (310) 550-5980.

Enhancements
in
progress

LAD/2 LAB

"No interruptions please.
These enhancements are a
very delicate business."
Professor 'LADdie'
MacDougal



Avoid the diskette shuffle while saving both time and money—

Let LAD/2 automatically
install and distribute
your operating systems
and applications over
multiple workstations.

LAD/2 NOW SUPPORTS

NetView DM/2	Comm. Mgr.
OS/2 2.0 / 2.1	DB2/2 OS/2
LAN Server	DOS/Windows
CID enabled and non-CID enabled OS/2, DOS, and Windows programs	

Just call 800 547-1283, ext.693



to find out what
LADdie is up to.

(International customers please call 817-961-5232)

Do the frugal thing—call 1 800 547-1283, ext. 693.

Visual REXX Development Environments

Visual REXX development environments provide a fast, inexpensive way of developing OS/2 applications. Combined with a very short learning curve and the ability to build graphical user interfaces with simple drag-and-drop actions, these environments are quickly becoming a popular method of creating conceptual prototypes as well as line-of-business applications. This article provides a first look at several visual REXX environments for OS/2.

Learning how to write a graphical user interface (GUI) program is not easy. If you have ever written, or attempted to write, Presentation Manager (PM) applications in C or COBOL, you will no doubt concur. The vast number of library calls, data structures, messages, and the event-driven architecture has caused more than one programmer to throw in the towel. Fortunately, there is another way to write GUI programs: visual REXX environments.

The three environments we will explore in this article are:

- WATCOM International's VX-REXX
- HockWare's VisPro/REXX
- Gpf Systems' GpfRexx

Philip Lieberman
Lieberman and Associates
Beverly Hills, California

Each of these environments is interpretive, giving you the chance to try different functions and settings immediately, and eliminating the laborious compile, link, and bind stages of compiled languages like C and COBOL. Each visual REXX package has its own integrated development

environment that provides an editor, debugger, and EXE generator. Also, each visual REXX package has built-in database and multimedia libraries.

The best thing about these packages is that you can learn to program real applications in just a few days, rather than in the six months it takes to learn conventional PM programming. Since all of these environments are built around 32-bit OS/2, you won't run into a brick wall when you reach the 80% or 90% completion point due to lack of memory or operating-system resources. These environments also allow you to extend the REXX environment by calling external dynamic link libraries (DLLs) that can be purchased or written internally.

A side benefit of programming in any of these visual REXX environments is simply the educational value. When you begin to write a simple "hello world" program in C, you are confronted with an overwhelming amount of overhead code, as well as a bewildering number of setup, compile, link, and bind options; if one of the options is wrong, the program will usually not run. The

warning message produced by a problem in a "hello world" program, as well as the tools, is frequently non-existent or complex in nature.

Using visual REXX products, the "hello world" program is trivial to produce—a single line of code or less. ("Hello world" can be a codeless program by setting the text or caption property of an object to "hello world"—therefore, no code is needed to produce the most trivial program example in computerdom!)

When you are ready to exploit the power of OS/2 Presentation Manager in a visual REXX environment, you need to learn only a minimal amount about a control or object in order to become productive. Naming and operating objects in these environments is very close to what is done in conventional C, except that the gory details are hidden behind the scenes. If you forget to set an object parameter, a default is almost always provided. When there is a problem in the program, a built-in debugger points out the problem clearly and immediately; you can usually make the change with the editor, then restart the program to immediately see the results.

By providing the PM interface and objects in small, bit-sized chunks, you gain immediate confidence in using PM. Since the paradigms used in visual REXX are identical to those used by compiled languages, the transition to the full-blown C interface is not completely overwhelming. In essence, using a visual REXX environment introduces you to the internals of PM gently and only as needed.

If your ultimate goal is to use a compiled language for your application, you will find it extremely useful to first learn one of these visual REXX packages to get

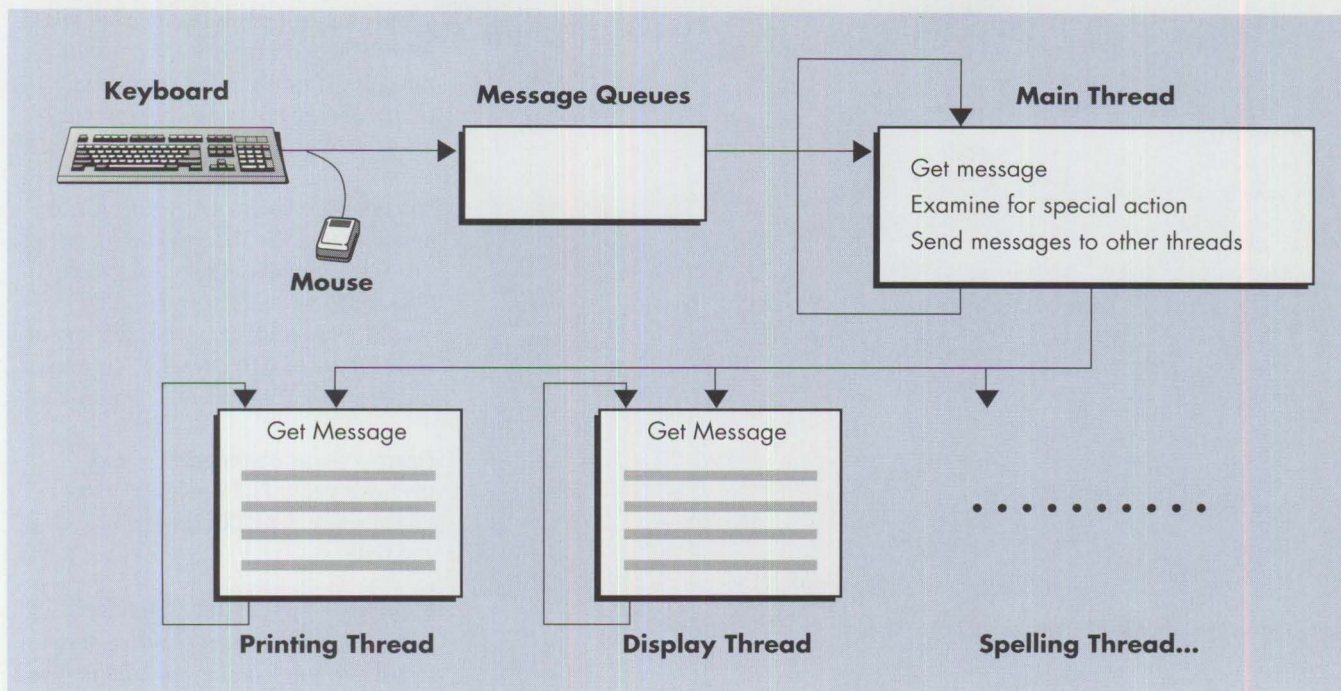


Figure 1. Conventional Programmer's C and COBOL Message Model

familiar with the messaging mechanism and object characteristics of PM programming. This experience will cut down your learning curve in attacking PM, no matter what the compiled language.

The Event Model: The Driving Force of Applications

The core of all PM programming is event programming using messages. I like to explain it as a wild animal show. In this show, different animals do tricks for different treats. For example, you might give sunflower seeds to a parrot to have it sing a song. Another type of food, such as a fish, would be given to a dolphin to have it jump through a hoop. Finally, you might want to give a programmer a salary bonus and a new sports car to have him or her finish the documentation on a program. The key behind each of these scenarios is that there is a unique incentive for each entity to get them to perform an action.

The food that drives PM applications is the *message*. Messages are created both internally and externally within the PM environment. *External* messages are generated for each mouse action and each keyboard action. *Internal* messages are generated by timers, DDE, and clipboard operations. Every time a message is generated, we call it an *event*.

In conventional compiled programs, you must write a loop that continuously checks for new messages. When a message arrives for that application, the program must examine the message and immediately handle it or pass it on to a default message handler. This model is depicted in Figure 1.

Because OS/2 has multithreading, the application does not have to continuously check for messages; when there are no messages for the application (the application queue is empty), the operating system automatically blocks the application. When a message arrives (constituting an event), the application is awakened (its thread is unblocked) and the message is processed.

In visual REXX environments, the message mechanics are hidden behind the scenes. Programmers can write REXX procedures to intercept these messages, but this is an exceptional case. The visual REXX environment keeps track of which messages go with which REXX procedures. If you have not written a REXX procedure for a particular message, the REXX environment provides a blank default procedure. As a result of this underlying architecture, you can create an application interface and drop in the code to handle different objects as you wish.

Message Handling in Conventional Compiled Languages

For readers who are already PM programmers, I acknowledge that compiled programs can also be created to handle this blank or partially implemented state. The fundamental difference here is that the REXX environment provides more default procedures for things like resizing, repainting, scaling, font metrics, printing, and a lot of other stuff that can sometimes be real headaches in conventional compiled languages.

In object-oriented languages like C++ and Smalltalk, default behaviors, similar to visual REXX environments, are provided with readily available *class libraries*. Class libraries are a wonderful way of getting around the limitations of conventional message handling. The visual REXX environments deal directly with classes, methods, and properties. Fortunately, the implementation and usage in REXX are much simpler and easier to handle.

Programming Process in Visual REXX Environments

Programming in a visual REXX environment is a snap. When you start up the environment, you are presented with a blank application panel and a palette of tools and potential controls to add to your application. The process of writing

CUA '91 objects include:

Windows
Pushbuttons
Radio buttons
Check boxes
Descriptive text
Entry field
Multiline entry field
Lists
Spin buttons
Groups
Notebooks
Containers
Sliders
Value sets
Pointers
Pop-up menus

Extra objects include:

Database access
Data graphing
DDE
Timers
Multimedia

Figure 2. Visual REXX CUA '91 Objects and Extra Objects

an application consists of the following steps:

1. Drag each control from the palette to the application panel.
2. Set the properties of the object (object name, color, size, fonts, and more).
3. Write the event (message) handler code for each object.
4. Make the application into an executable (EXE) file.
5. Ship it (of course, you never write code with bugs, so there is no debugging stage).

If you don't like the appearance or operation of your application, these environments allow you to change object properties as well as go behind the scenes to change the code that handles the objects. All the operation for development is carried out in a single seamless environment, with all the development tools available at one time.

Palettes—Objects and Tools

Each visual REXX environment touts its product as having more objects available for design than its competitors. These objects break down into the objects that are part of IBM's CUA '91 standard and extra objects (listed in Figure 2). Another

differentiating feature between environments is support for REXX code multi-threading (for better concurrent operation). Some environments also provide access to low-level PM functions and DLLs.

You can place objects during the initial design process (by drag and drop), or the REXX application can selectively place objects and make them visible while it runs. Besides placement, you have continuous control of virtually all object properties while the application is running.

Figure 3 shows object palettes from the three visual REXX products mentioned above: VX-REXX, VisPro/REXX, and GpfRexx.

Methods—Application Behavior

There are many different sources of code to achieve your working application. The developer is responsible for creating the REXX code to do something when buttons are pressed and keyboard actions are taken. The code to do this is sometimes called the *method*. Figure 4 lists some methods built into visual REXX.

There are also general routines that are designed to be shared by all routines and are therefore called *common libraries*. The general or common routines are written as separate pure REXX files that are globally callable procedures.

Most of the intrinsic (built-in) methods are designed to deal with the CUA '91 objects and basic language requirements

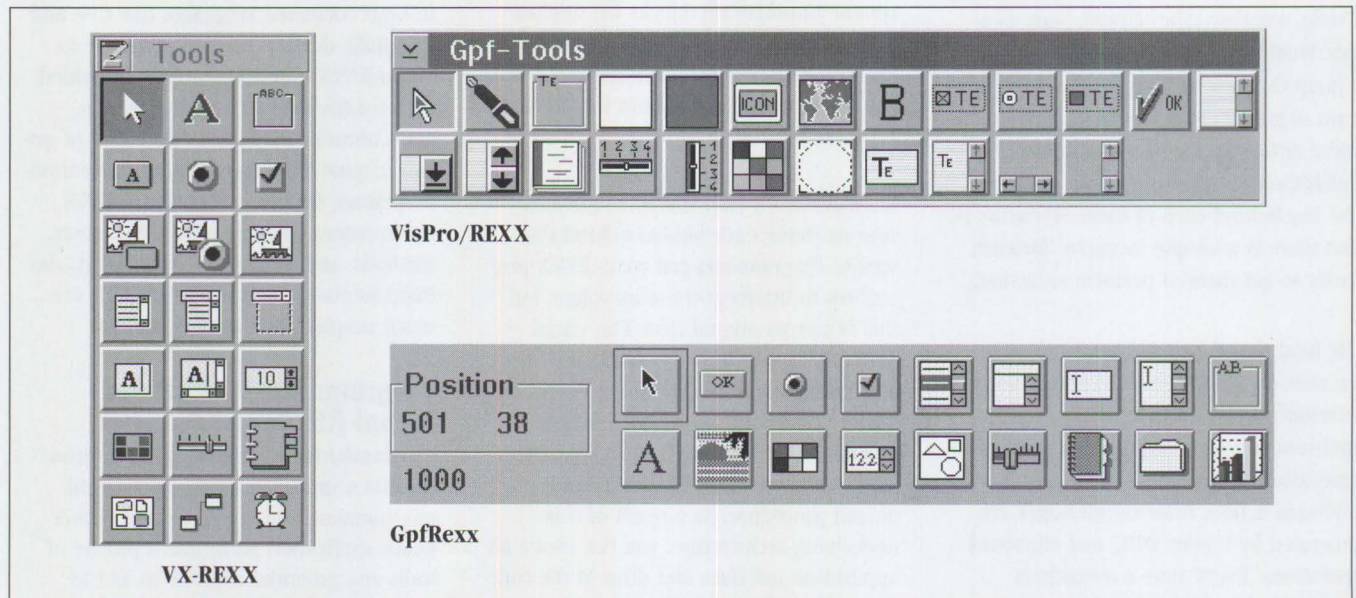


Figure 3. Visual REXX Object Palettes

(files, strings, math, etc.). You are still responsible for writing the control logic that makes the application do what you intend. Fortunately, even writing the basic REXX code has been made easier with a pop-up menu that has the most common REXX coding sequences ready for paste-in with a click of the mouse.

For example, in VX-REXX, to change the code behind a button object, you would right-click on the button to get its object menu. Then, select the event cascade sub-menu. Finally, select the click event.

Figure 5 shows the menu selection.

Once the proper event is selected, the code window will pop up (see Figure 6). At this point, you enter the REXX code to handle this type of click event. If you are not sure which REXX sequence to use, a right-click in the code window provides an object window with an insert code . . . option, as shown in Figure 7.

By selecting the insert code . . . option, you get many different sources of code to insert:

- All object names and all methods (or actions) to perform on them
- Canned procedures for handling databases, graphics, and the like
- Basic REXX constructs as well as library calls

The beauty of this type of implementation is that you have to memorize neither object names nor the exact syntax of the entire REXX language and its extensions—they are all a click or two away.

In VisPro/REXX, all objects are graphically represented next to a code window for the object under revision. With this architecture, you can just drag and drop the object to be referenced into the code window (see Figure 8). Once the object is dragged in, all the methods that can be performed on or with the object are available. VisPro/REXX also provides an extensive library of code sequences for selectable paste-in.

Projects and Forms

Each of the REXX environments has a different model of how the entire application is constructed.

Built-in methods:

- | | |
|---|-------------------------------------|
| Insert/delete items in list boxes (ahead or behind an item, at list start or end) | Control periodic timers |
| Sort items in list boxes (ascending, descending) | Control PM windows |
| Look up items in list boxes | Add/delete/update files/directories |
| Add/delete/select notebook pages | Handle accelerator keys |
| Line, curve, bitmap drawing | Handle help files |
| Get/set object properties | Multithreading |
| PM message handling | Clipboard |
| | String handling |
| | Drag and drop |

Figure 4. Some Methods Built Into Visual REXX

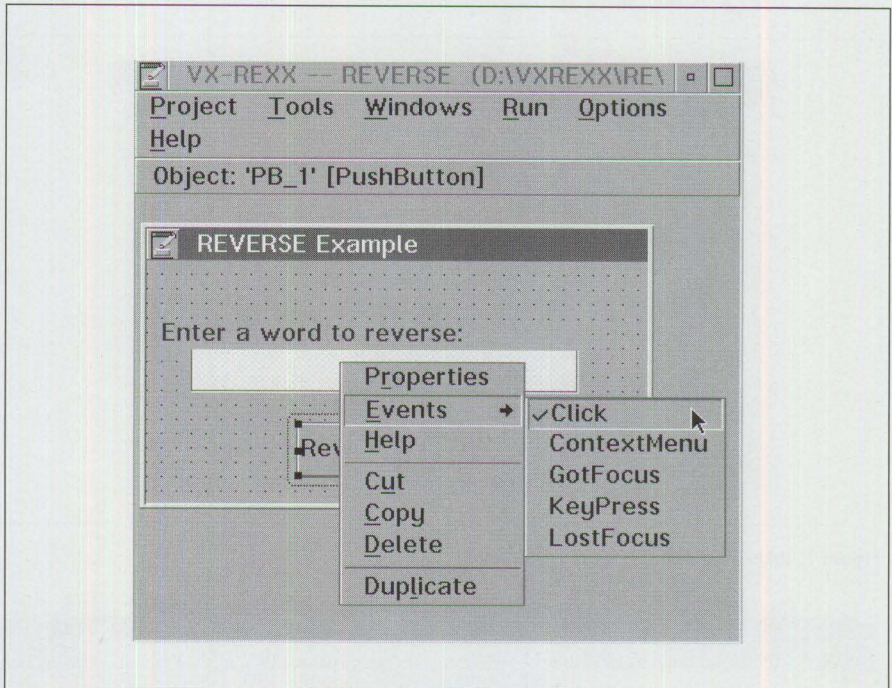


Figure 5. Cascaded Menu in VX-REXX

In VX-REXX, you create a *project* file to which you add *code* and *window* files. Each window file has its own event queue and event processing loop (only one can run at a time). The primary active window can send messages to other windows in the system, including other VX-REXX window files. Multiple threads can also be utilized for concurrent window file execution.

In VisPro/REXX, the fundamental object is a *form*. Multiple forms can be created using templates. There is normally only a single main form with other secondary forms linked to it.

In GpfRexx, an Interface Builder application creates a *.ORC file that contains definitions of windows and controls

within those windows. The Interface Builder also connects the REXX code to the events generated by the controls designed by the Interface Builder.

Properties—Object Appearance

Object *properties* control how an object appears and how it acts. When an object is created, it has a default set of object properties. You can change the properties during the design process or modify them at run time. The initial or default properties are set via different methods, depending on which environment you are using. In GpfRexx, objects are set via the Interface Builder. Both VisPro/REXX and WATCOM VX-REXX set their object properties via a Settings notebook, as shown in Figure 9.

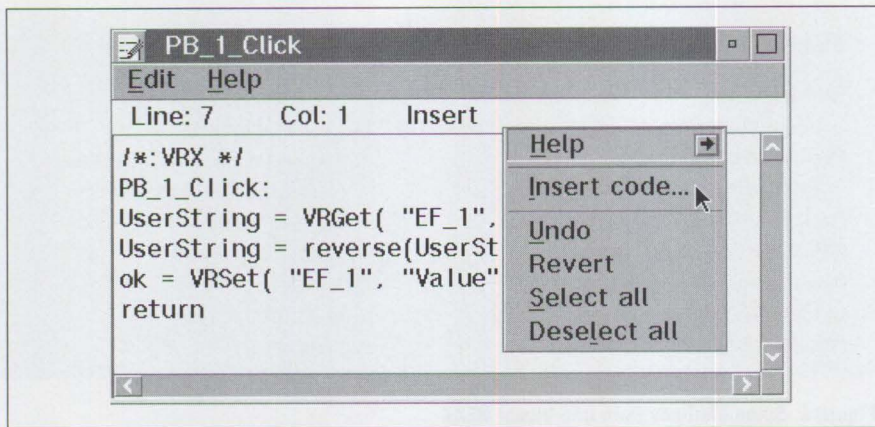


Figure 6. Code Window Object Pop-Up Menu

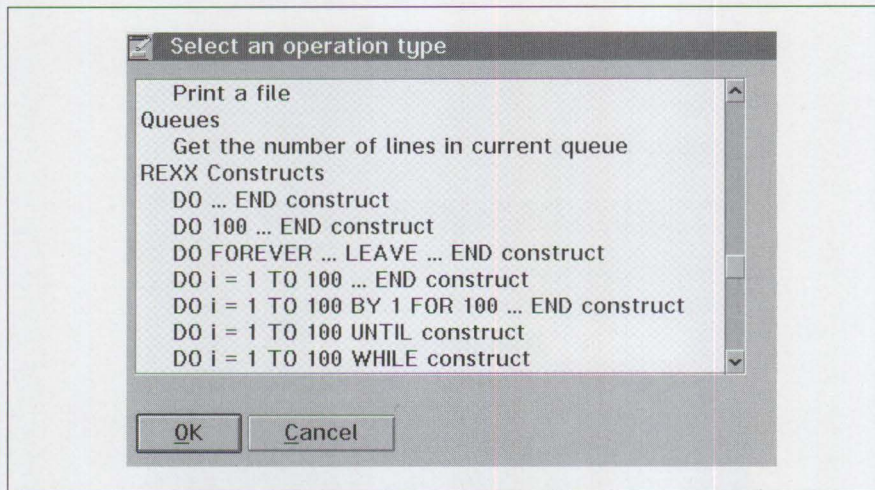


Figure 7. Insert Code Dialog

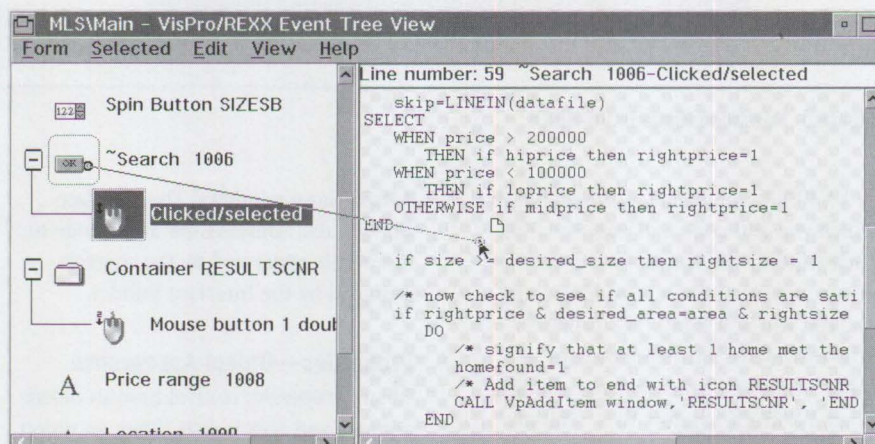


Figure 8. VisPro/REXX Drag-and-Drop Programming in the Event Tree View

The real fun begins when you change object properties at run time. This is where you fill in dialog boxes, list boxes, entry fields, containers, bitmaps, and more. Let's create a simple visual REXX program that asks for a word and then reverses it. In this example, we use the following objects:

- *Descriptive text* object, to provide instructions (Object Name=DT_1)
- *Entry field* object, to take in the new word and display the reversed word (Object Name=EF_1)
- *Pushbutton* object, to perform transformation of word (Object Name=PB_1)

The descriptive text object would contain the caption Enter a word to reverse:. The entry field would have an initial value property of all blanks. Finally, the pushbutton would have a caption property of Reverse Now!. Figure 10 contains the code for the pushbutton object.

Voila! That is an entire working GUI program for OS/2. Figure 11 shows the working program.

VisPro/REXX—Database Designer

One of the most amazing pieces of software design is the database designer found in the VisPro/REXX development environment. This part of the package is capable of reverse-engineering existing DB/2 files by importing an existing database design from an existing database. The built-in database designer can automatically create structured query language (SQL) code directly from imported database definitions using automatically created graphical entity relationship (E/R) diagrams—allowing the creation of drag-and-drop database front-ends.

The database designer also allows for the creation of *ad hoc* queries. The SQL statement window interactively shows you all tables and allows you to select information interactively. An SQL syntax diagram is created automatically.

Tools are provided to create databases and to control each table's attributes. The tools also allow you to draw the relationship between different tables. You can also set up referential integrity with the tools provided.

Which Visual REXX Product is Best?

The question of which visual REXX product is best requires a tough decision. All three products offer all the CUA '91 controls and access to both database and multimedia. In my opinion, the best manuals and samples are provided by WATCOM VX-REXX, but HockWare's VisPro/REXX is a close second. As for performance, VisPro/REXX seems to start and execute faster than the VX-REXX product. I also like the layout of the Settings notebooks in VisPro/REXX better than in VX-REXX. The GpfrExx product provides the seductive capability of converting your REXX applications (designed using their Interface Builder) to C for the ultimate in performance.

In general, I think most developers would be very happy with any of the three choices. For those least experienced with REXX and GUIs, the WATCOM product is my choice. For those with more experience, either the WATCOM or HockWare solutions would be a good choice. If you ultimately plan to target your application from REXX to C, you should consider the family of Gpf products.

One parting thought: If you are about to begin an OS/2 project using a GUI front end, you must try these environments! They offer the quickest way to build an application, as well as an excellent way to learn what goes on beneath the hood of the OS/2 Presentation Manager. At the very least, I think one of these products will become your prototyping tool of choice for OS/2. Give them a try!

For more information about products in this article:

VX-REXX:
 WATCOM International Corp.
 415 Phillip Street
 Waterloo, Ontario, Canada N2L 3X2
 Phone: (800) 265-4555 or
 (519) 886-3700
 Fax: (519) 747-4971

VisPro/REXX:
 Hockware, Inc.
 P.O. Box 336
 Cary, NC 27512-0336
 Phone: (919) 380-0616
 Fax: (919) 380-0757

GpfRexx:
 Gpf Systems, Inc.
 30 Falls Road
 Moodus, CT 06469-0414
 Phone: (203) 873-3300
 Fax: (203) 873-3302



Philip Lieberman is a well-known author, lecturer, and consultant in real-time systems and OS/2. He is the author of the Learning Tree course *OS/2 2.x: A Comprehensive*

Hands-On Workshop. Mr. Lieberman owns Lieberman and Associates, which produces the LAN ICU for IBM LAN Server software product. He is also a magician and a member of the Hollywood Magic Castle and the International Brotherhood of Magicians.

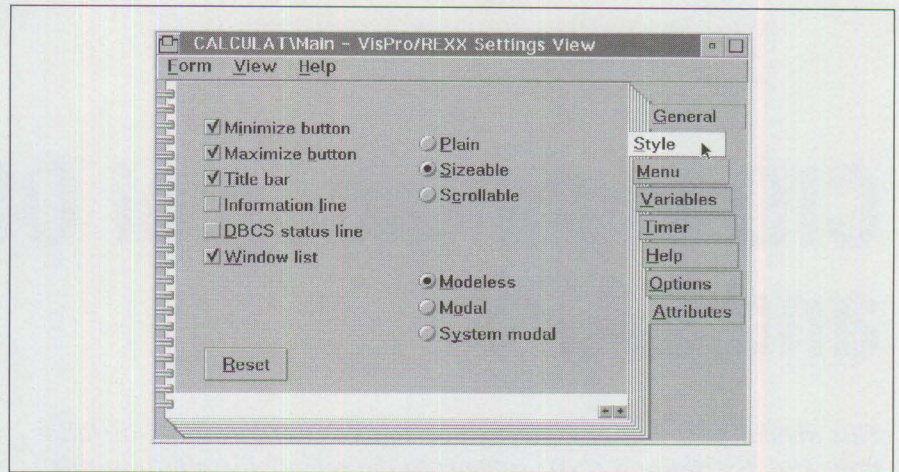


Figure 9. Style Tab from a Form Settings Notebook in VisPro/REXX

```

/*:VRX */
PB_1_Click:
UserString = VRGet( "EF_1", "Value" )
UserString = reverse(UserString)
ok = VRSet( "EF_1", "Value", UserString )
return
  
```

This pushbutton object program contains these statements and parameters:

<code>/*:VRX */</code>	The first line must be a comment, but it can be changed at will.
<code>PB_1_Click:</code>	The procedure label name for the first pushbutton is <code>PB_1</code> (the supplied default). The added <code>_Click</code> is the event type.
<code>VRGet</code>	This is the VX-REXX property Get verb.
<code>"EF_1"</code>	This is entry field 1 (there is only one).
<code>"Value"</code>	This is the property to retrieve.
<code>UserString</code>	This REXX variable receives the result of <code>VRGet</code> .
<code>reverse</code>	This standard REXX verb reverses all of the letters in a string.
<code>ok</code>	The success or failure flag on the property Set operation.
<code>VRSet</code>	This is the VX-REXX property Set verb, which puts back the reversed string.
<code>return</code>	The end-of-event handler for pushbutton.

Figure 10. Pushbutton Object Code in VX-REXX

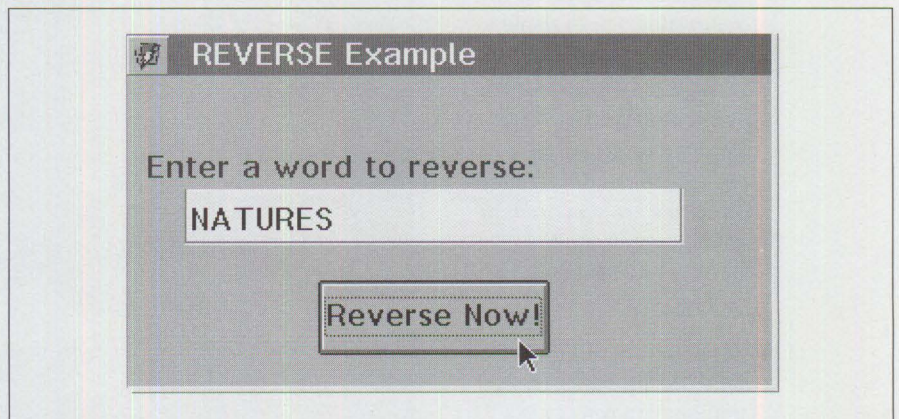


Figure 11. A Working OS/2 PM Application in Five Minutes

CID Installation of OS/2 2.11 and LAPS

This article, part 2 of 2, explains how to add configuration, installation, and distribution (CID) support to the remote installation code server discussed in part 1, which appeared in the May/June 1994 issue of this magazine. While part 1 covered a redirected installation of OS/2 2.x, part 2 explains a CID installation of OS/2 2.11 and LAN Adapter and Protocol Support (LAPS).

The acronym CID stands for configuration, installation, and distribution. This article explains the CID setup for OS/2, including LAPS. Why? Because LAPS is a necessary component for CID installation of the OS/2 platform (the base operating system, LAN Server, Communications Manager/2, etc.).

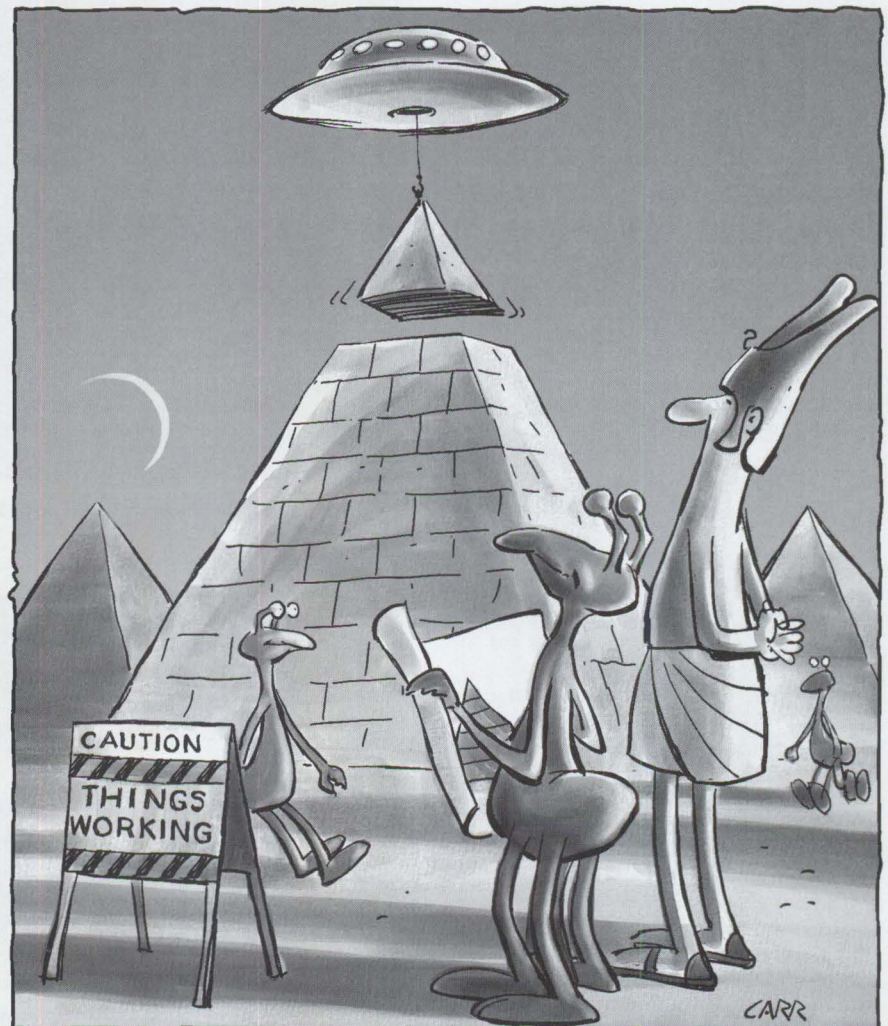
supported via a response file. All customization of the client's workstation can be configured in the response file, which is stored at a central location such as a code server. The code server houses the OS/2 diskette images, installation programs (e.g., SEINST.EXE), response files (SAMPLE.RSP), installation command files (LCU command file), and any logs created during the installation.

Erik Mintz
IBM Corporation
Boca Raton, Florida

Three Objectives of CID

CID strives for three objectives: eliminate end-user involvement, consolidate configuration, and automate distribution. Let's take a closer look at each of these objectives.

- **Eliminate end-user involvement.** Initial or replacement installation of OS/2 is carried out via redirection. For example, redirection takes place when a client's machine connects to the code server via a redirected drive letter (e.g., X:, Y:, or Z:). Different installation scenarios are supported by using a response file such as the one provided with OS/2: SAMPLE.RSP.
- **Consolidate configuration.** Here, the product's configuration parameters are



■ **Automate distribution.** Here, the CID installation is driven via the LAN CID Utility (LCU). Instead of an end user performing a diskette-based installation, the LCU REXX command files drive the installation. The LAN CID Utility uses a set of REXX procedures to install multiple CID-enabled products.

IBM offers three tools for building a CID environment:

- **CASSETUP**, provided with the Network Transport Service/2 (NTS/2) Utilities diskette.
- **SRVSETUP**, provided with the IBM International Technical Support Organization's (ITSO's) Redbook *Automated Installation for CID-Enabled OS/2 V2.x* (GG24-3783).
- **CIDSETUP**, developed by Tom Lambert of the IBM Personal Systems Competency Center in Roanoke, Texas. To acquire CIDSETUP, call (800) 547-1283. (See Lambert's article "Easy Setup of CID Code Servers" in the January/February 1994 issue of *Personal Systems*.)

None of these tools, however, gives an administrator or end user an "under-the-covers" view of a CID installation setup for OS/2. This two-part series of articles provides that "under-the-covers" look. Part 1 showed how to build your remote installation server to do a redirected installation of OS/2 2.x. In part 2, you will learn how to add the CID support to your server for OS/2 2.11 and LAPS.

The hardware requirements in part 2 remain the same as in part 1 for the client and server. The only software change in part 2 is to install the new version of LAPS, service level WR07045, on your code server and then create an image of the new version (see Step 1 below). To order LAPS service level WR07045, call (800) 992-4777.

In part 1, we described using a naming convention to store five different versions of OS/2 2.x on your OS/2 Remote Installation Server. Part 2 adheres to that naming convention (see Figure 1), adding XR06300, which is new since part 1 was published. However, part 2 only describes a CID installation setup for OS/2 2.11 and LAPS.

Naming Convention	OS/2 Version
XR06100	OS/2 2.0 Corrective Service Facility (CSF), ServicePak II (XR06100)
XR06200	OS/2 2.1 Customer Service Diskette (CSD), ServicePak I (XR06200)
OS2V21D	OS/2 2.11, dark blue diskettes
OS2V21G	OS/2 for Windows, green diskettes
OS2V21S	OS/2 2.1, salmon diskettes
XR06300	OS/2 for Windows, ServicePak I (XR06300)

Figure 1. Naming Convention for OS/2 Versions

d:\cid\client
d:\cid\client\os2v21d
d:\cid\compile
d:\cid\dll
d:\cid\dll\os2v21d
d:\cid\img\lcu
d:\cid\log
d:\cid\log\lcu
d:\cid\log\os2v21d
d:\cid\log\srvifs
d:\cid\rsp\laps

Figure 2. Directory Structure for Drive D: on Code Server

1. Type: `unpack2 d:\cid\img\os2v21d\disk_2\bund1e d:\cid\exe\os2v21d /n:setboot.exe (all one line)`
2. Type: `unpack2 d:\cid\img\os2v21d\disk_8\rexx d:\cid\dll\os2v21d`
3. Type: `unpack2 d:\cid\img\os2v21d\disk_11\required d:\cid\exe\os2v21d /n:xcopy.exe (all one line)`

Figure 3. Unpack Instructions for OS2V21D

Building Your Code Server

The procedure below describes how to add CID support to your existing OS/2 Remote Installation Server. The step-by-step procedure guides the administrator or end user through the setup for a CID installation of OS/2 2.11 and LAPS. The client's name in this example is John Doe.

1. To create an image of LAPS service level WR07045 on your code server, insert the NTS/2 LAPS diskette into drive A: and type:


```
a:\lapsdisk a: d:\cid\img\laps
```
2. On drive D: create the directories shown in Figure 2.
3. In this step, you add the LAN CID Utility to your code server. You need two different service levels of the

NTS/2 Utilities disk: WR07000 and WR07045. The latest service level, WR07045, does not contain the input files needed for the CASPREP utility. To obtain those input files, perform steps 3a and 3b.

- 3a. Insert the NTS/2 Utilities Disk from service level WR07000 into drive A: and type:

```
copy a:\applets\cas*.*
d:\cid\compile
```

- 3b. Open an OS/2 window. Type `service /q` (this parameter will shut down the SRVIFS services).

Insert the NTS/2 Utilities Disk from service level WR07045 into drive A: and type:

```
copy a:\lcu d:\cid\img\lcu
```


1. Use an ASCII editor to edit `d:\cid\rsp\os2v21d\default.rsp`
2. Add/change the following parameters under the appropriate section within the response file:

```
exitonerror=1
rebootrequired=0
userexit=del %targetdrive%\*.add
```

3. Save the file.
4. Type: `copy d:\cid\rsp\os2v21d\default.rsp d:\cid\rsp\os2v21d\johndoe.rsp <this is all one line>`
5. Now edit `d:\cid\rsp\os2v21d\johndoe.rsp` and change the following keywords to reflect the values below:

```
basefilesystem=2
formatpartition=1
```

These two parameters cause the target partition to be formatted using the FAT file system.

Figure 4. Mandatory Response-File Changes for a CID Installation

1. Insert Disk 11 into drive A:
2. Type: `unpack2 a:\required d:\cid\rsp\os2v21d /n:sample.rsp`
3. Type: `copy d:\cid\rsp\os2v21d\sample.rsp d:\cid\rsp\os2v21d\default.rsp <this is all one line>`

Figure 5. Unpack Instructions for SAMPLE.RSP

1. Type: `cd\cid\img\laps`
2. Type: `lapsrsp c:\ibmcom\protocol.ini d:\cid\rsp\laps\lapsrsp.rsp /t:c: /i:product <this is all one line>`

Figure 6. Creating a LAPS Response File

1. Type: `cd:\cid\compile`
2. Type: `casprep casadv.fil d:\cid\client\os2v21d\default.cmd`
3. Type: `copy d:\cid\client\os2v21d\default.cmd d:\cid\client\os2v21d\johndoe.cmd <this is all one line>`

Figure 7. Compiling Your LCU Command File

- 3c. Type:


```
copy a:\srvifs
d:\cid\img\srvifs
copy d:\cid\img\srvifs\
xi*. * d:\server
copy d:\cid\img\srvifs
\service.exe d:\server
```
4. This step unpacks some files necessary to set up your code server.

Figure 3 gives the unpack instructions for OS2V21D.

5. Using an ASCII editor, edit `d:\server\service.ini` (the SRVIFS configuration file) to add the statements below:

```
alias=readonly,single,cid,d:
\cid <no space after the :>
```

```
alias=readwrite,single,log,d:
\cid\log <no space after the :>
```

6. Shut down your machine and reboot.
7. In this step, customize your client's OS/2 installation options by editing the response file that comes with OS/2 2.11. From part 1, `DEFAULT.RSP` should reside in the `d:\cid\rsp\os2v21d` directory. Figure 4 explains the mandatory changes to the response file for a CID installation. If you do not have `DEFAULT.RSP` for OS/2 2.11, see Figure 5 for the unpack instructions for `SAMPLE.RSP`.

For consistency, I always try to keep the client's response file name the same as the LCU command file name. The reason will become clearer in step 9 when we look at the LCU command file for the client, John Doe.

8. You can use the `LAPSRSP` utility to create a response file for a CID installation of LAPS. Parameters for `LAPSRSP.EXE` can be found in two resources: *NTS/2 LAN Adapter and Protocol Support* (S96F-8489), section 4-2, and ITSO's Redbook *Automated Installation for CID-Enabled Extended Services, LAN Server V3.0 and Network Transport Services/2* (GG24-3781), page 41. Follow Figure 6 to create a LAPS response file.

Note: You are creating a LAPS response file based on your code server's `PROTOCOL.INI`. Any protocol support in your code server's `PROTOCOL.INI` will be installed on your client's workstation.

9. This step describes how to compile and edit your LCU command file. Figure 7 provides an example for building a customized LCU command file. This method, along with `JOHNDOE.RSP`, will install OS/2 2.11 and LAPS on your client's C: drive.

Appendix A in *NTS/2 Redirected Installation and Configuration Guide* (S96F-8488) describes the `CASPREP` utility, keywords, and input files that serve as the basis for the example in Figure 7.

10. After completing the steps in Figure 7 that compile your LCU command file `JOHNDOE.COMD`, use an ASCII line

editor and refer to Figures 8 and 9 to edit JOHNDOE.CMD.

Figures 8 and 9 describe, by line number, the individual changes that must be made to the JOHNDOE.CMD file. Follow the instructions in double parentheses. First, complete the changes in Figure 8, then make the changes in Figure 9.

This step creates your CID LAN Transport (LT) client boot diskettes. Now follow the steps in Figure 10.

Installation Procedures Available via Fax

The installation information in parts 1 and 2 of this article has been summarized in a series of documents available via fax. There are documents for both redirected and CID installation of all the products listed in Figure 1. To obtain any of these documents via fax:

1. Call (900) 225-5522.
2. Open a Program Management Report (PMR) for the Installation team.
3. Request a particular document(s) using its fax name(s) in Figure 11. IBM will then fax the document to you.

Or

Send a letter to the OS/2 Service and Support Center at the address below requesting a document from the list of fax name(s) in Figure 11.

OS/2 Service and Support
C/O CID Team
1000 NW 51 St.
Boca Raton, FL 33431

References

This article was compiled from information in several ITSO Redbooks:

- *OS/2 V2.0 and V2.1 Remote Installation and Maintenance* (GG24-3780)
- *NTS/2 Redirected Installation and Configuration Guide* (S96F-8488)
- *NTS/2 LAN Adapter and Protocol Support Configuration Guide* (S96F-8489)
- *Automated Installation for CID-Enabled OS/2 2.x* (GG24-3783)

```
line 25. d1="x:" ((delete the remainder of the line))
line 30. ((change v200 to os2v21d))
line 33. ((change v200 to os2v21d))
line 34. rsp_dir = d2
line 35. log_dir = d3
line 41. server = "\\imagesrv" ((add this entire line))
line 51. /* "SRVATTCH" d2 "srvifs_ alias1" */ ((delete the remainder of the line))
line 52. /* "SRVATTCH" d3 "srvifs_server2" */
line 57. x.1.name = "OS/2 2.11"
line 59. ((change os2v20 to os2v21d))
line 60. ((change os2v20 to os2v21d))
line 61. ((change os2v20 to os2v21d))
line 62. ((change default.rsp to johndoe.rsp))
line 137. ((add \os2v21d to /cmd:"d1\client"))
        e.g. /cmd:"d1\client\os2v21d
line 151. add two blank line
line 152. ((For ease, copy lines 118 - 125, paste three separate times starting at line 152,
        each time adding a blank line to separate the queues. Edit the three queues added
        to reflect the changes below. Or, you can manually add these statements starting
        at line 152. BE VERY CAREFUL OF THE SYNTAX!))

x.thinifs1 = 12
x.12.name = "SRVIFS Requester1"
x.12.statevar = ""
x.12.instprog = img_dir\srvifs\thinifs /S:"img_dir"\srvifs /t:"ifs_dir" /tu:"bootdrive"\,
                /l1:"log_dir"\srvifs\client".log",
                /req:"client" /srv:"server"\cid /d:"d1

x.12.rspdir = ""
x.12.default = ""

x.thinifs2 = 13
x.13.name = "SRVIFS Requester2"
x.13.statevar = ""
x.13.instprog = img_dir\srvifs\thinifs /S:"d1\img\srvifs /t:"ifs_dir" /tu:"bootdrive"\,
                /l1:"log_dir"\srvifs\client".log",
                /req:"client" /srv:"server"\rsp /d:"d2

x.13.rspdir = ""
x.13.default = ""

x.thinifs3 = 14
x.14.name = "SRVIFS Requester3"
x.14.statevar = ""
x.14.instprog = img_dir\srvifs\thinifs /S:"d1\img\srvifs /t:"ifs_dir" /tu:"bootdrive"\,
                /l1:"log_dir"\srvifs\client".log",
                /req:"client" /srv:"server"\log /d:"d3

x.14.rspdir = ""
x.14.default = ""

NUM_INSTALL_PROGS = 14
```

Figure 8. Editing the LCU Command File, Part 1 of 2

```
((After changing/adding lines from Figure 8., the next set of
statements you need to edit to JOHNDOE.CMD should occur around line 188
beginning with the "Do Forever" statement. Edit this section to reflect
the changes below.))

Do Forever
Select
  when OVERALL_STATE = 0 then do
    if RunInstall(x.SEINST) == BAD_RC then exit
    if RunInstall(x.LAPS) == BAD_RC then exit
    if RunInstall(x.THINIFS1) == BAD_RC then exit
    if RunInstall(x.THINIFS2) == BAD_RC then exit
    if RunInstall(x.THINIFS3) == BAD_RC then exit
    if RunInstall(x.CASINSTL) == BAD_RC then exit
    Call CheckBoot
  end
  when OVERALL_STATE = 1 then do
    if RunInstall(x.IFSDDEL) == BAD_RC then exit
    if RunInstall(x.CASDELET) == BAD_RC then exit
    Call Reboot
  end
end
exit
```

Figure 9. Editing the LCU Command File, Part 2 of 2

- *Automated Installation for CID-Enabled Extended Services, LAN Server V3.0 and Network Transport Services/2* (GG24-3781)

The NTS/2 references are provided with IBM's OS/2 LAN Server 3.0.

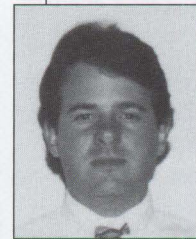
Acknowledgment

The author thanks the CID team from the OS/2 Service and Support organization—Ahmad Ali, Steve Cunningham, and Jack Leung—for helping to create this article.

1. Type: cd\cid\exe\os2v21d
2. Type: sedisk /s:d:\cid\img\os2v21d /t:a:
(SEDISK prompts for two formatted disks; the first is the installation disk, and the second is Disk 1 of your LT boot diskettes).
3. Type: cd\cid\img\laps
4. Type: thinlaps d:\cid\img\laps a: ibmtok.nif
(It is very important to leave a space between a: and ibmtok.nif! If you are using a Token-Ring Bus Master Adapter, use ibmtrbm.nif in place of ibmtok.nif).
5. Type: cd\cid\img\srvinfos
6. Type: thininfos /t:a: /s:d:\cid\img\srvinfos /tu:a: /req:johndoe
/d:x: /srv:\imagesrv\cid /w <this is all one line>
7. Type: thininfos /t:a: /s:d:\cid\img\srvinfos /tu:a: /req:johndoe
/d:y: /srv:\imagesrv\rsp /w <this is all one line>
8. Type: thininfos /t:a: /s:d:\cid\img\srvinfos /tu:a: /req:johndoe
/d:z: /srv:\imagesrv\log /w <this is all one line>
9. Type: cd\cid\img\lcu
10. Type: casinst1 /tu:a: /cmd:x:\client\os2v21d /pa:x:\img\lcu
/pl:x:\dll\os2v21d;x:\img\lcu
/ll:z:\lcu\srvinfos_req.log /d <this is all one line>
11. Edit Diskette 1's CONFIG.SYS file by adding the following statement at the end:

set sourcepath=x:\img\os2v21d

Figure 10. Creating CID LAN Transport Client Boot Diskettes



Erik Mintz is an associate programmer in IBM's OS/2 Service and Support group in Boca Raton, Florida. He is currently a CID analyst. He joined IBM in 1990 in

environmental engineering and later worked in multimedia development. Erik has a BS degree in hydrogeology from Florida Atlantic University and is completing his MS degree in civil engineering there.

Redirected Installation Documents:

Fax Name	Product
OSIN610R	XR06100 (OS/2 2.0 ServicePak II)
OSIN21BR	OS/2 2.1 (Blue Version)
OSIN21SR	OS/2 2.1 (Salmon Version)
OSIN620R	XR06200 (OS/2 2.10 ServicePak I)
OSIN211R	OS/2 2.11
OSINWINR	OS/2 for Windows
OSIN630R	XR06300 (OS/2 for Windows ServicePak I)

CID Installation Documents:

Fax Name	Product
OSIN21BC	OS/2 2.1 (Blue Version)
OSIN21SC	OS/2 2.1 (Salmon Version)
OSIN620C	XR06200 (OS/2 2.10 ServicePak I)
OSIN211C	OS/2 2.11
OSINWINC	OS/2 for Windows
OSIN630C	XR06300 (OS/2 for Windows ServicePak I)

Figure 11. Fax Documents for Redirected and CID Installations

Upgrading from Microsoft LAN Manager to IBM LAN Server 3.0

With the release of IBM's LAN Server 3.0 Advanced in October 1992 and the later release of related products such as LAN Server for Multimedia and LAN Distance, many customers who use Microsoft's LAN Manager are upgrading to IBM LAN Server 3.0. LAN Server 3.0 provides superior performance, better quality, superior OS/2 support, and better domain and LAN management function than Microsoft LAN Manager provides (LANQuest Labs, November 1992 and National Software Testing Laboratories). This article details the steps necessary to convert Microsoft LAN Manager servers to IBM LAN Server 3.0.

Generally speaking, since IBM LAN Server was developed from Microsoft LAN Manager, it is fairly easy to upgrade from LAN Manager to LAN Server 3.0. This migration is easy because LAN Server's BACKACC utility can back up the user definitions and access control lists on a LAN Manager server, and because LAN Server Advanced can read data on LAN Manager High-Performance File System (HPFS) partitions.

Steve French and Gary Hunt
IBM LAN Development
Austin, Texas

LAN Server 3.0 not only includes advanced features not present in LAN Manager, but also has much-improved performance. A large percentage of LAN Server was rewritten for LAN Server 3.0, but compatibility with LAN Manager and earlier versions of LAN Server was intentionally preserved

whenever possible. This compatibility eases the upgrade to LAN Server 3.0.

This article focuses on the most difficult migration scenario—the migration of LAN Manager with HPFS partitions to LAN Server 3.0 Advanced.

Four Migration Steps

There are four steps in migrating from LAN Manager to LAN Server:

1. Backing up the server
2. Preparing the server for an OS/2 operating system upgrade
3. Upgrading the OS/2 operating system and LAN Server
4. Restoring the server configuration and data

The major operations in these four steps are standard operations. For example, if you have a server, you need to have a backup procedure even if you are not planning an upgrade.

Generally speaking, the upgrade can be accomplished with standard LAN Server tools (BACKACC and RESTACC) plus the standard installation programs in OS/2, LAN Adapter and Protocol Support (LAPS), and LAN Server 3.0.

Step 1: Backing Up the Server

The first step in any upgrade or migration scenario is to make a reliable backup of the entire system. In subsequent steps, we discuss a migration scenario that does not require reformatting disk space and therefore does not require restoring the data from the backup. However, sometimes things go wrong, so even if you plan to use the techniques that do not require reformatting, you should still do a backup to guard against the remote possibility of an unrecoverable migration error.

You also need to back up your user definitions and access control lists (ACLs) with the BACKACC utility before you begin the migration. Make sure that you run BACKACC against all of the HPFS partitions on your LAN Manager server. The application programming interfaces (APIs) that the BACKACC utility uses to perform its work are the same in LAN Manager as in LAN Server, so you can use the LAN Server BACKACC command to back up your LAN Manager configuration.

Note that you can also use the LAN Manager BACKACC and RESTACC commands, but you cannot use the LAN Manager BACKACC followed by the LAN Server RESTACC. Although the set of backup file names (NETACC.BAK and ACLBAKx.ACL) that the two versions of BACKACC produce are the same, the formats of the backup files are different, and LAN Server's RESTACC cannot read

backup files created by LAN Manager's BACKACC.

Also note that if you use the default drive and path for storing the backup files, the LAN Server BACKACC command stores the backup files in the X:\LANMAN\ACCOUNTS subdirectory. The BACKACC command determines its default drive and path by looking at the LANROOT (the path where the IBMLAN.INI file resides); the LANROOT is the parameter following the /I: in the IFS= line in the CONFIG.SYS file.

In addition to backing up your user definitions, access control lists, and data, you should record printer and queue definitions from Print Manager, and you should save the LANMAN.INI and PROTOCOL.INI files from your LAN Manager server for reference in subsequent steps. If you are running LAN Manager fault tolerance, you should unmirror and deactivate any currently mirrored drives on which you plan to install OS/2, LAPS, or LAN Server.

Step 2: Preparing the Server for an Operating System Upgrade

LAN Manager 2.x servers run on top of OS/2 1.3, and LAN Server 3.0 runs on top of OS/2 2.x. Therefore, before you migrate to LAN Server 3.0, you must upgrade your OS/2 installation.

The term *system partition* denotes a partition in which both OS/2 and LAN Server are installed. If you plan to install LAN Server in the same partition as OS/2, then you must reformat the target system partition for both the OS/2 upgrade and the LAN Server installation.

However, LAN Server can be installed on a different partition from OS/2. If you intend to install LAN Server on a different partition even though LAN Manager is currently installed on the same partition as OS/2, you must reformat both target partitions to avoid the server preparation.

Situations When Preparing Your Server Is Not Necessary

There are several situations in which you do not need to prepare your server for an upgrade of the operating system. These situations are:

- The system partition will be reformatted, and you are migrating to LAN Server Advanced.
- The system partition of the LAN Manager server is formatted as FAT, and

you are migrating to LAN Server Advanced.

- All partitions are formatted as FAT, and you are migrating to either LAN Server Entry or LAN Server Advanced.

Situations When Preparing Your Server Is Necessary

In some cases, you will have to prepare your server for the OS/2 upgrade. They include:

- Migrating from LAN Manager with any HPFS partitions to LAN Server Entry.
- Migrating from LAN Manager with an HPFS system partition to LAN Server Advanced (without reformatting the system partition).

... take advantage of LAN Server's unique public application support. . .

Access Control Lists

If the system partition of your LAN Manager server is formatted as HPFS, its file system structures contain access control lists created by HPFS386. Microsoft LAN Manager's and IBM LAN Server Advanced's HPFS386 file systems can understand ACLs. However, the HPFS file system driver that comes with OS/2 does not understand how to process HPFS386 ACLs. Therefore, if you attempt to upgrade OS/2 when ACLs are in the file system structures, you will get an error when the OS/2 installation attempts to replace files in the subdirectories protected by ACLs.

During our test migration, we attempted to install OS/2 2.1 when an ACL protected C:\ (our system partition). The OS/2 installation gave us an error, informing us that it could not install the OS/2 loader and kernel because they were locked by an ACL.

To successfully upgrade OS/2 without reformatting your system partition, the ACLs need to be removed from the file system. (Keep in mind that we backed up the ACLs when we backed up the server in the previous step, so at this point it is OK to delete the ACLs.)

If you are upgrading LAN Server 2.0 Advanced to LAN Server 3.0 Advanced,

or if you are upgrading OS/2 2.0 to OS/2 2.1 under LAN Server 3.0, you should use the PREPACL utility to remove the ACLs from the file system. The LAN Server PREPACL utility does not work under LAN Manager; however, LAN Manager provides the same capability in the DETACH function of its SETUP utility. Therefore, if your LAN Manager system partition is formatted as HPFS, you need to run the DETACH function of the LAN Manager SETUP utility to prepare your server for the OS/2 upgrade.

Keep in mind that the LAN Manager DETACH function removes (and discards) the ACLs in all of your server partitions. If you run the LAN Manager DETACH function, be sure to run RESTACC against all of your server partitions during the restoration step.

If you are migrating from LAN Manager to LAN Server Entry, you need to remove the ACLs from all HPFS partition(s), because LAN Server Entry uses the OS/2 HPFS file system to read HPFS data, but the OS/2 HPFS file system does not understand how to read ACLs. You also need to run the LAN Manager DETACH function in this case, even if your system partition is FAT or if you plan to reformat your system partition.

Step 3: Upgrading the Operating System and LAN Server

Once the above steps have been performed, follow the standard procedures to install OS/2 2.1, LAPS, and LAN Server 3.0. No special steps or considerations are needed during this phase of the migration.

If you had planned to reformat your OS/2 partition, you should do so when asked during the installation of OS/2. If you are installing LAN Server on a separate partition, and if you also plan to reformat the LAN Server system partition, you should perform the reformat after installing OS/2 and before installing LAN Server. Make sure you keep your backup files on a different partition from the ones being reformatted.

Most parameters associated with LAPS and LAN Server are defaulted at installation time. However, you must specify some parameters during the LAPS and LAN Server installation processes:

machine name, domain name, server role, and the number of NetBIOS resources configured in LAPS and allocated to LAN Server.

You can use the `PROTOCOL.INI` and `LANMAN.INI` files from the LAN Manager server as a reference to help configure your new LAN Server server the same way that the LAN Manager server was configured. The LAPS and LAN Server installation programs take your answers to the questions asked by the installation program and properly add them to your new `PROTOCOL.INI` and `IBMLAN.INI` files.

Step 4: Restoring the Server Configuration and Data

The first thing in this final step is to restore your server data. You should restore your data before restoring your ACLs, because `RESTACC` discards ACLs for subdirectories and files that it does not find. If you did not reformat your server data partitions, you do not have to restore from your backup.

Restoring NET.ACC

At this point, you have a `NET.ACC` file that is basically empty. To migrate your user definitions from LAN Manager to LAN Server, you will either replace the `NET.ACC` file created by LAN Server installation with the `NETACC.BAK` file you created when you ran `BACKACC` on the LAN Manager server, or you will replicate your user definitions from the Domain Controller of the domain you choose to join.

Using the procedures described below will allow you to create—without additional steps—domains consisting of migrated LAN Manager servers and newly installed LAN Server servers. If you are adding existing LAN Server servers to one of these domains, you may have to reset the server user ID password at the existing LAN Server servers using the procedures described in the LAN Server publications.

Migrating a Stand-Alone LAN Manager Server to a LAN Server Domain Controller

The strategy for migrating a stand-alone LAN Manager server to a LAN Server Domain Controller is to use the `NETACC.BAK` file from the original LAN Manager server and to add the user

definitions necessary to bring it up as a domain controller in a domain.

Use the following steps to perform this procedure:

1. Copy `NETACC.BAK` over the newly installed `NET.ACC` in `\IBMLAN\ACCOUNTS`.

You should use the `NETACC.BAK` file from the server that most closely matches the user definitions that you want to have in this new domain (which may not be the `NETACC.BAK` file that was backed up from this stand-alone server). The procedure below will work regardless of which server the `NETACC.BAK` file came from, with the exception that the `SERVERS` group may already exist and may not have to be created.

2. Perform a local User Profile Manager (UPM) logon with a valid administrator account from the LAN Manager server.
3. Perform the following steps using UPM:
 - a. Add a User ID via UPM equal to the name of the local server.
 - b. Add a group named `SERVERS`.
 - c. Add the User ID equal to the name of the local server to the new `SERVERS` group.
4. Open an OS/2 window or full screen, and enter


```
NET ACCOUNTS /ROLE:PRIMARY
```

You can now successfully start the server.

Next, add more server definitions to this domain by using the LAN Server Full-Screen Interface (FSI) from any OS/2 client. The FSI automatically creates the necessary user IDs at the domain controller, but the passwords are set to null.

If you are adding a migrated LAN Manager server as a new additional server to this domain, you should use the `NET.ACC` file that the LAN Server installation provided on the new additional server, and let LAN Server's automatic user replication mechanism (provided by the Netlogon service) restore user definitions to the additional server. This procedure enables you to avoid having to create user IDs or to reset passwords on the additional server. Keep in mind that user definitions that exist in the `NETACC.BAK` file of

the additional server (formerly LAN Manager) do not exist in the `NET.ACC` file on the domain controller and need to be added at the domain controller.

Migrating a Domain LAN Manager Server to a LAN Server Domain Controller

To perform this type of migration, start by copying the `NETACC.BAK` file over the `NET.ACC` file created by LAN Server installation in `\IBMLAN\ACCOUNTS`. If you are migrating a Backup or Member, you need to use the `NET ACCOUNTS` command to change the role of the `NET.ACC` file from Backup or Member to Primary. After these steps, the server will start successfully.

You are now running with a `NET.ACC` file that has user definitions for additional servers in the previous LAN Manager domain. LAN Server does not allow an additional server definition to be created for a user ID that already exists. As a result, if you are adding to this domain additional servers whose server names already exist as user definitions, those user definitions must be deleted.

Migrating a LAN Manager Server to a LAN Server Additional Server

The easiest way to bring up a newly migrated LAN Manager server (Primary, Backup, Member, or Stand-Alone) as a LAN Server additional server is to use the `NET.ACC` file created by LAN Server installation, and let `NET.ACC` replication automatically bring down the necessary user definitions from the domain controller. Keep in mind that if there are user definitions that existed on the LAN Manager server but do not exist at the domain controller, they will need to be added. With this procedure, the new additional server will start successfully, and user definition replication will occur automatically with no additional steps required. If the server does not start because the `NETLOGON` service cannot start, you will have to set the password of the additional server user ID at the domain controller to null.

Replication of the user definitions to this new additional server begins as soon as the server is started. You should let the user definitions replicate completely before restoring the ACLs to this server. Full `NET.ACC` replication usually takes less than 10 minutes, but for large `NET.ACCs` (more than 250 user definitions), it may take longer.

Restoring ACLs

The next step in migrating your server is to restore the server's access control lists. To restore ACLs, you simply run the RESTACC command for each ACLBAKx.ACL file that was created during the backup phase of the migration. Some recent backup products, such as Sytos from Sytron, back up the file system ACLs along with the data. In this case, you do not have to run RESTACC to restore your ACLs; however, if you are not sure, it is best to restore ACLs using RESTACC.

Finishing the Upgrade

The final step in migrating your server is to finish configuring your server and to create the same printer definitions that existed on the LAN Manager server. Unless you made specific configuration changes to your LAN Manager server, you will operate efficiently with the configuration that the LAN Server installation program sets up as a default. However, if you made changes from the default on your LAN Manager server, you can use your backup copy of LANMAN.INI and PROTOCOL.INI and make similar changes to your new LAN Server server.

Finally, remember to create printer objects to correspond to the OS/2 1.3 printer and queue definitions that you recorded before the upgrade started. Also, if you are going to restore a fault tolerance configuration, run the FTSETUP utility to remirror drives as desired.

Other Considerations

If you have shares that are automatically issued at server startup time in SRVAUTO.PRO, you will need to create "share at server startup" aliases for these resources after LAN Server is installed. There is no conversion utility for changing SRVAUTO.PRO to LAN Server alias definitions.

Remote initial program load (RIPL) configurations will have to be converted by hand to the LAN Server format, since there are no conversion tools.

You will want to create aliases for your resources to take advantage of LAN Server's directory function.

We highly recommend that, after conversion, you take advantage of LAN Server's unique public application support, which enables users to automatically get a personalized menu of network-based or local applications at logon time.

Client Interoperability Considerations

Many customers will prefer to migrate in stages—servers first, then client machines. It is possible to do this, especially for simple file and print sharing, but there are some restrictions.

Client machines that are set up for domain logon should log on to a domain controller of the same type. While in many cases it is possible to log on from LAN Manager requesters to IBM LAN Server servers (and vice versa), IBM's unique features such as public applications, logon assignments, and home directories may not work as expected. OS/2 Presentation Manager printing between LAN Manager requesters and OS/2 LAN Server servers may have problems. Network administration through the IBM full-screen interface on IBM requesters will not usually work to Microsoft LAN Manager servers. Frequently, cross-product network administration from the command line will work.

IBM LAN Server 3.0 supports a more advanced server message block (SMB) dialect than Microsoft LAN Manager, but IBM servers and requesters can negotiate a lower-level dialect when communicating with Microsoft LAN Manager machines. For this reason, IBM and Microsoft servers can interoperate for file sharing and most print sharing.

IBM requesters can also interoperate with Microsoft Windows for Workgroups servers, IBM PC LAN Program (PCLP) servers, and Microsoft MS-NET servers, but these three servers cannot handle 32-bit OS/2 application requests since they are more primitive, DOS-based servers. Windows for Workgroups, PCLP, and MS-NET requesters can interoperate with IBM servers for file and print sharing.

IBM servers and requesters can also interoperate with Microsoft Windows NT as well as various X/Open-compliant

UNIX servers. Since X/Open has standardized the SMB protocol for PC-to-UNIX interworking, other companies such as DEC have been able to produce compatible, SMB-based network requesters and servers based on their X/Open specifications.

We in IBM LAN Server development continue to try to improve interoperability with other SMB-based networks such as those from Microsoft. A few interoperability enhancements are available in LAN Server Corrective ServicePak IP07045. The LAN Server 3.00.1 manufacturing refresh, released March 31, 1994, also includes these interoperability enhancements.

Steve French is a Staff Programmer in LAN Systems within the IBM Personal Software Products division in Austin, Texas. Since joining IBM in 1989, he has worked on LAN Server as a designer and (currently) as the development lead for the LAN Requester. He has published many articles and has patent applications pending for his work on LAN Server. Steve has a BA degree in Computer Science and a Master's degree in Electrical Engineering, both from Rice University. His Internet userid is `stevef@ausvm1.vnet.ibm.com`.



Gary Hunt is an Advisory Programmer in LAN Systems, IBM Personal Software Products, Austin, Texas. He is currently the design lead for LAN Server future

releases. Gary has worked on all LAN Server releases in various roles as development manager, technical planner, and designer. He joined IBM in 1975 and has an MS degree in Electrical Engineering from the University of Texas. His Internet userid is `gthunt@ausvm1.vnet.ibm.com`.

Stretching Your LAN with LAN Distance

Trying to maintain user productivity when users connect remotely into your LAN can be challenging. This article discusses IBM's LAN Distance technology and explains how LAN Distance provides remote access to local area networks.

The need for remote access to local area network (LAN) resources has never been greater. More and more companies are putting their work force on the road or in their homes.

With IBM LAN Distance, mobile users and telecommuters have the LAN access they need to do their jobs. LAN Distance makes it appear as though you grabbed your LAN wires and stretched them to wherever you are!

Edward Duhe'
IBM Corporation
Roanoke, Texas

LAN Distance enables you to be as productive on the road as you are in the office by giving you simultaneous access from a remote location to all the resources on your LAN. LAN Distance allows remote OS/2 or DOS/Windows

workstations to connect to a token-ring or Ethernet LAN through dial-up or leased-line connections. Once your remote workstation is connected, it becomes just another device on the LAN, and you can run your LAN-based applications unmodified.

LAN Distance Components

LAN Distance has two components: LAN Distance Remote and LAN Distance Connection Server.

LAN Distance Remote

A LAN Distance Remote is a stand-alone workstation with significant capabilities of its own. A LAN Distance Remote comes with two clients, one for OS/2 and the other for DOS/Windows. Each LAN Distance Remote client can dial out to another LAN Distance Remote or to a LAN Distance Connection Server. When two LAN Distance Remote workstations are connected, they form a virtual LAN, as shown in Figure 1.

The virtual LAN provides a "LAN" connection between the two workstations across a communications link or wide area network (WAN). With this "virtual LAN," client/server applications can run as if they were on a physical LAN.

No network hardware such as a token-ring or Ethernet network adapter is required in the LAN Distance Remote workstation.

When a LAN Distance Remote connects with a LAN Distance Connection Server (Figure 2), the remote can be given access to all the resources on the LAN such as LAN servers, print servers, database

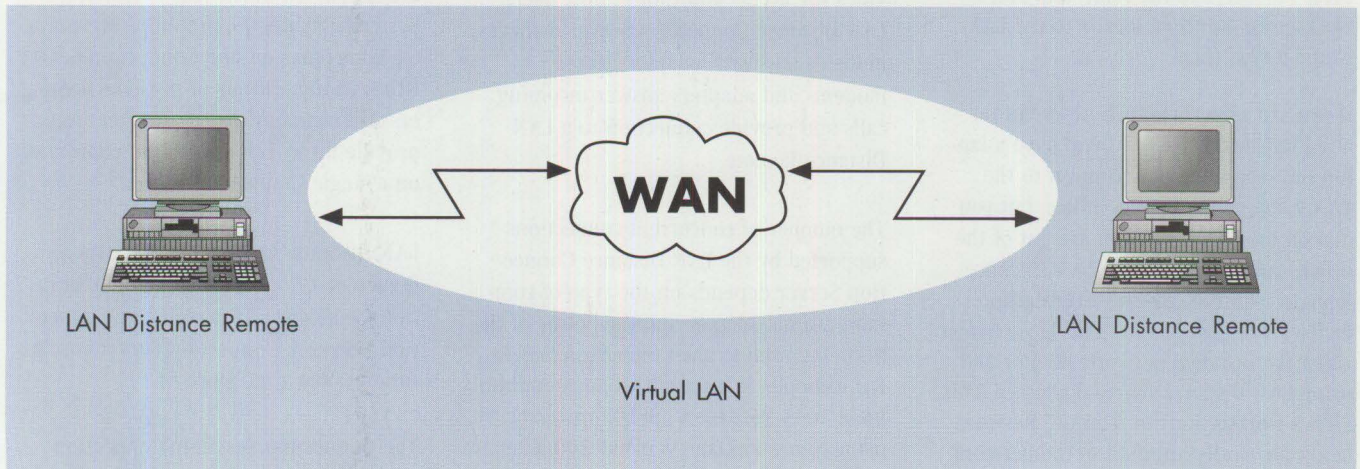


Figure 1. Remote to Remote (LAN over WAN)

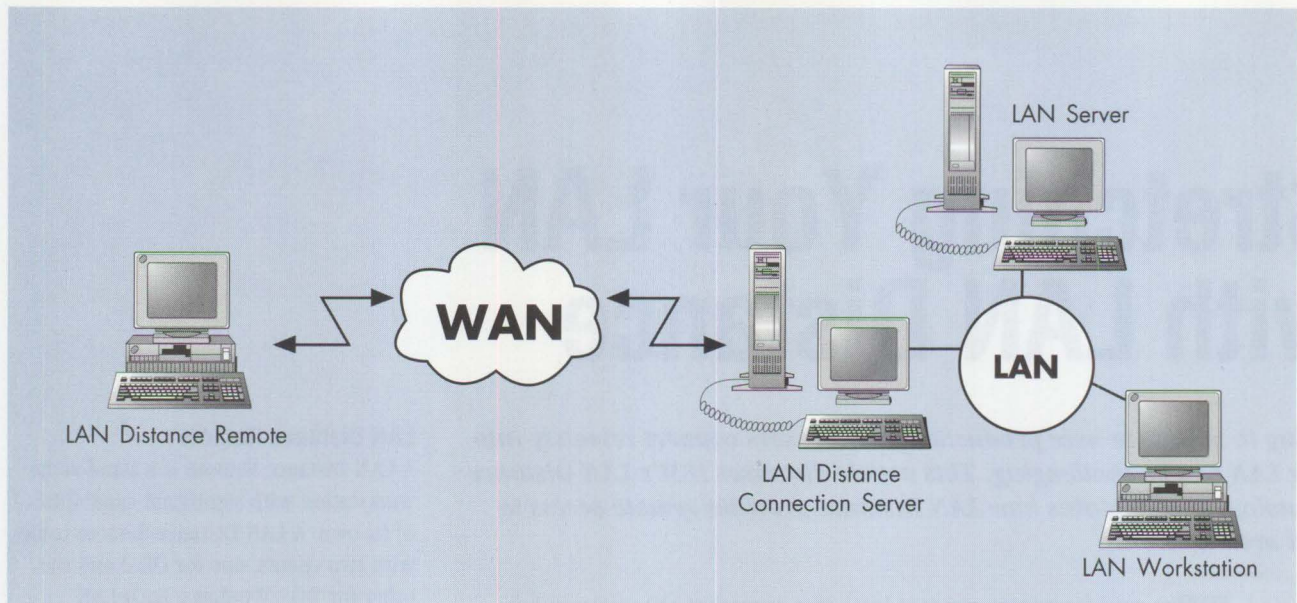


Figure 2. Remote to Connection Server

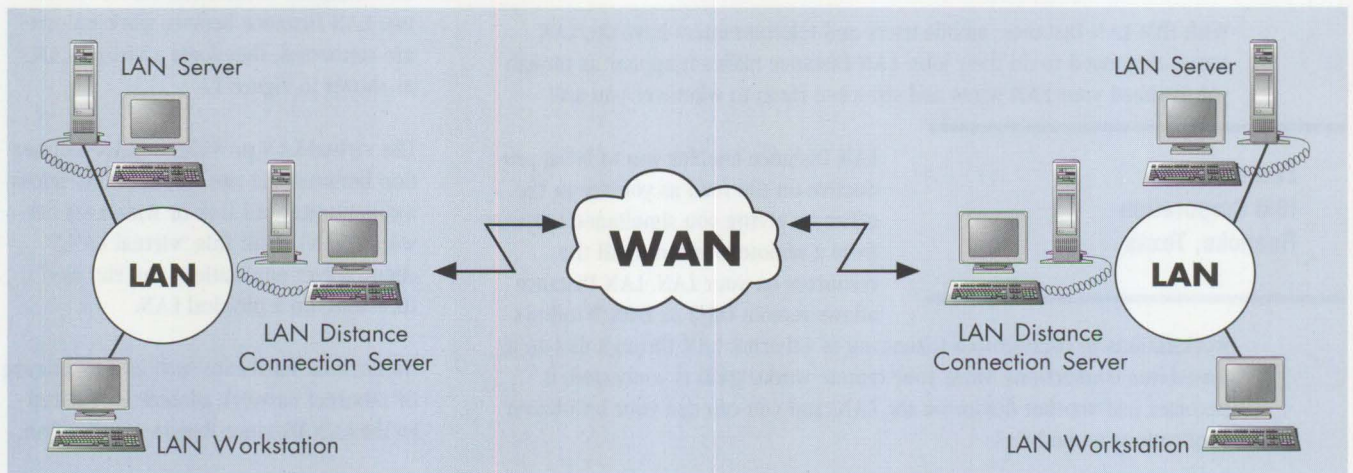


Figure 3. Connection Server to Connection Server

servers, host gateways, and other workstations. The LAN Distance Remote accesses these resources in the same way that a workstation attached locally to the LAN would access them.

If you are a user who is both remote and local—for example, if you have a laptop that you physically connect to the LAN when you are in the office, but you dial up the LAN when you are out of the office—you can utilize the LAN Distance Remote “shuttle” feature. This feature enables workstations to “shuttle” between using the installed network adapter and using LAN Distance’s virtual adapter. This allows workstations to “shuttle” between being physically attached to the network via the installed network adapter and being remotely attached via LAN Distance.

LAN Distance Connection Server

The LAN Distance Connection Server provides the access point for remote users. LAN Distance Connection Server manages modems and WAN adapters; these modems and adapters answer incoming calls and provide connection to a LAN Distance Remote.

The number of concurrent connections supported by the LAN Distance Connection Server depends on the type of communications adapter and the power of the PC being used as the Connection Server. For example, in an asynchronous environment, a 33 MHz Intel 386DX processor using four Star Gate Avanstar 200 adapter cards provides up to 64 concurrent communications sessions. For integrated

services digital network (ISDN) connections, three IBM Primary Rate Adapters are supported in the IBM Personal System/2 Model 95, providing sixty-nine 64 Kbps ports on one Connection Server. Other configurations of personal computers and communications adapters can provide up to 128 concurrent connections on a single Connection Server.

LAN Distance Version 1.1 supports a Remote-to-Connection Server configuration (see Figure 2) as well as a Connection Server-to-Connection Server configuration (shown in Figure 3).

The Connection Server-to-Connection Server environment provides for casual bridging of two LANs. If you need an

occasional bridge between two LANs, this configuration provides an attractive solution. Keep in mind, however, that it does not replace a leased line bridge that handles heavy use.

Whenever you allow the outside world into your LAN, you must ensure its security. The Connection Server provides security by requiring identification and authentication of users as they access your LAN. LAN Distance Connection Server has very comprehensive security mechanisms that are based on the Message Authentication Code Standard ANSI X9.9. Figure 4 lists the security features in LAN Distance.

Remote Node Technology

Using *remote node technology*, LAN Distance provides transparent access to client/server applications across a remote link or a WAN without any change to the application.

In remote node technology, a remote workstation is connected to a LAN through a communications link, but the remote workstation appears to be attached locally (via physical wiring) to the LAN. Since the LAN treats the remote workstation as just another node, your application runs the same at the remote workstation as it would on a locally attached workstation.

The strength of remote node technology lies in its support for multiple LAN applications running simultaneously over a single dial-up remote link. For example, you might have an OS/2 multitasking workstation that is dialed into a LAN and running 3270 emulation and Lotus Notes, or you might have a DOS/Windows workstation that is logged onto a LAN server running transmission control protocol/internet protocol (TCP/IP) in the background and cc:Mail in the foreground. Remote node technology, therefore, provides the same capabilities that you would have on a workstation that is physically attached to a LAN.

In a remote node environment, applications are physically running at the remote workstation; that is, the applications are loaded into the remote workstation's memory and are processed by the remote workstation's CPU. This contrasts with a remote-control environment in which

Passphrase. LAN Distance uses passphrases instead of passwords. A passphrase is case-sensitive and can be 4 to 32 characters long, including blanks.

User Types. There are three user types: user, system administrator, and security administrator. An administrator manages the modems and ports, changes the configuration, and accesses audit and tracking statistics. To add and change user information or to enable or disable security, you must have security administrator authority.

Callback. LAN Distance can call back a phone number associated with the userid, or the user can specify the callback number when dialing the Connection Server.

Workstation Identification. A logical adapter network address can be associated with a particular user. This means the user has to dial in from a computer that has the specified adapter address.

Policy Options. The security administrator can define policy options that include minimum password length, duplicate passphrase check, maximum and minimum passphrase age, and maximum number of logon attempts.

Logon Time Intervals. Logon time intervals allow the administrator to limit the days of the week and the times of day when a user can log on to the Connection Server.

Encryption. The passphrase is encrypted using data encryption standard (DES), a private key encryption standard, which creates a random session key. This key is used by the client and the server to authenticate all service requests between the LAN Distance Remote and the Connection Server.

Figure 4. LAN Distance Security Features

applications run on the host PC that is physically attached to the LAN. In this environment, the remote user dials into the host PC and takes over control of the host PC.

When considering a remote node environment, think about how the LAN applications will be run. Due to the relatively slow speed of the communications link, loading an application from a LAN drive could take quite some time. It is best, therefore, to physically load the applications onto the remote workstation and to access the data over the remote link.

LAN Distance provides a network-layer link for communicating with the LAN; therefore, when planning your environment, remember to include your network software as well as LAN Distance. For example, consider a LAN Distance connection into an OS/2 LAN Server token-ring network. In this situation, you run LAN Distance and either OS/2 LAN Requester or DOS LAN Requester on the remote workstation. LAN Distance makes the

connection to your LAN, and then you log on to the OS/2 LAN Server to access your LAN data files.

Another example is a Lotus Notes environment. Here, the remote workstation has LAN Distance and the Notes client software installed. The remote workstation then accesses the Notes databases that reside on the LAN.

Sample Remote Node Environment

Here is an example of a configuration I use. My remote workstation has the following software installed:

- OS/2 2.1
- LAN Distance Remote 1.1
- Communications Manager/2 (for 3270 emulation)
- TCP/IP for OS/2 (Telnet and file transfer protocol [FTP])
- Lotus Notes Client
- OS/2 LAN Requester

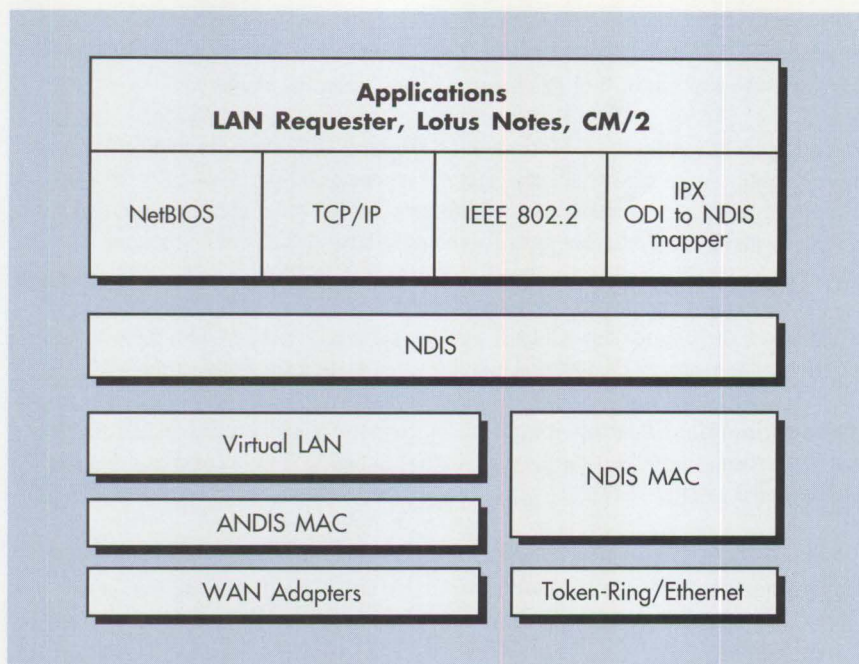


Figure 5. LAN Distance Architecture

With this configuration, I can dial into the LAN and have a 3270 session to the host, access LAN data files and printers, run Telnet sessions on the Internet, and access my Lotus Notes database—all at the same time!

Wide Area Network

A wide area network is a network that provides communications beyond the local area network. In a LAN Distance environment, a WAN can be provided by one of several types of communications links. Under OS/2, LAN Distance supports asynchronous, synchronous, ISDN, and X.25 connections. Under Windows, LAN Distance supports only asynchronous connections.

The most commonly used LAN Distance connection is the asynchronous dial-up connection. Using today's 14.4 Kbps modems with V.42bis data compression, you can achieve throughputs of up to 57.6 Kbps. With the growing availability of ISDN service, you can reach speeds of 64 Kbps. Also, with adapters such as IBM's Wide Area Connector, you can reach synchronous links up to 2.048 Mbps with a leased line. X.25 connections through a Packet Assembler/Disassembler (PAD) can also be used to access a LAN Distance Connection Server.

Protocols

LAN Distance provides transparent access for LAN applications that work with network driver interface specification (NDIS) or that use an NDIS-compliant protocol. NDIS is an industry standard architecture for device drivers that allows a workstation to support multiple network adapters and protocols.

LAN Distance supplies several NDIS-compliant protocols. For OS/2, NetBIOS and Institute of Electrical and Electronics Engineers (IEEE) 802.2 protocols are supplied. For Windows, NetBIOS is supplied while other protocol support, such as IEEE 802.2, requires IBM's LAN Support Program. NetWare support for both OS/2 and Windows is provided by an open data-link interface (ODI)-to-NDIS mapper. This mapper maps the Novell ODI calls to the standard NDIS interface. Although not supplied with LAN Distance, TCP/IP is supported for OS/2 and Windows through NDIS-compliant drivers.

LAN Distance is actually based on an extended or advanced version of NDIS 2.0.1. This advanced NDIS, or ANDIS, enables the support of communication adapters other than the normal token-ring or Ethernet adapters. To support these adapters, ANDIS provides a layer that, in effect, creates a virtual LAN adapter. The

upper-level NDIS protocols see a LAN adapter, which is then mapped to WAN hardware. It is this virtual LAN adapter that provides the transparency for the protocol stacks and applications that use them. (See Figure 5.)

LAN Distance Connections Server Bridge

In a Remote-to-Connection Server environment (shown in Figure 2) where the LAN Distance Remote is attached to the LAN Distance Connection Server, there are two separate LAN segments. One LAN segment is the local LAN, to which the Connection Server is physically attached. The second LAN segment is the WAN connection between the LAN Distance Remote and the LAN Distance Connection Server.

The Connection Server provides a bridge between the two LAN segments. Because these two LAN segments transmit data at very different speeds—16 Mbps versus 9600 bps, for example—care must be taken not to overload the slow-speed segment. Only LAN traffic that is relevant to the remote workstation should flow across the Connection Server bridge, onto the WAN segment, and to the LAN Distance Remote.

You can avoid overloading the slow-speed segment through filtering. LAN Distance has filters that limit the data on the WAN by analyzing the frames on the network. Frames are accepted or rejected based on user-defined criteria. The LAN Distance Connection Server Bridge supports the following types of filter criteria:

- Source addresses
- Source address range
- Service Access Point (SAP)
- NetBIOS names
- Bit-mask destination address

An alternative to setting up a specific filter is to use LAN Distance Bridge's automatic filtering capabilities. Enabling the automatic filtering allows the LAN Distance Bridge to "learn" what requests are being made by the remote workstation. Based on these requests, the LAN Distance Bridge uses protocol-specific criteria and forwards only the frames that are relevant to the remote workstation.

If you are a token-ring user, you will have to consider that a bridge hop is added to your LAN configuration. A *bridge hop* means network traffic must cross a bridge or "hop" from one LAN segment to another. When you use LAN Distance to access a multi-segmented token-ring LAN, consider the number of bridge hops and segment numbers. Failure to properly configure the bridge hops and segment numbers can prohibit the remote workstation from accessing all the needed resources.

Usage Examples

The following four scenarios illustrate how these concepts are put to practical use:

- **Mobile User Access.** The most popular use for LAN Distance is to provide mobile workers with simultaneous access to the resources needed to do their jobs. For example, a sales force has laptop computers that they take on their customer calls. From these laptops, they can dial up the office LAN and check their e-mail, access a LAN-based database to retrieve customer information, and log on to the mainframe to gather the latest product information. This can all be done with a single connection per user.
- **Remote Systems Management.** LAN Distance can be very helpful in an environment where you must manage remote LANs. Using LAN Distance with products such as System Performance Monitor/2 and LAN NetView, you can

administer these LANs from your own site. You can also administer your OS/2 LAN Server, NetWare server, or Banyan Vines server over the remote connection.

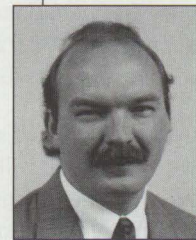
- **Technical Support.** Remote-control utilities are commonly used to provide support for users on a LAN. Using LAN Distance with a LAN-enabled remote-control product can enhance your technical support environment. LAN Distance gives you a single connection mechanism that you can use to "take over" a user's workstation. You may also need to copy files or make changes to a file server to correct the user's problem, but because you are connected with LAN Distance, you can accomplish all these tasks without disconnecting.
- **Distributed Meetings.** Using LAN Distance with a product such as IBM's Person-to-Person, you can conduct meetings remotely. Person-to-Person is a conferencing solution that displays information on any LAN-attached workstation. Using Person-to-Person, you can display charts or draw on a shared "chalkboard." Because LAN Distance stretches the LAN, this type of conferencing can be done from a place that is remote to any location.

LAN Distance: Connectivity and Productivity

Working on the road or at home is becoming commonplace in today's business world. Why shouldn't you enjoy the

same level of connectivity from a remote location that you have at your office? With LAN Distance's implementation of remote node technology, you can connect to your LAN and maintain your productivity from anywhere.

For an evaluation copy of LAN Distance, call (800) 3IBMOS2 (342-6672) and request the LAN Distance 60-day evaluation CD-ROM. This full production-level version of LAN Distance (with publications for the Remote and Connection Server) locks 60 days after it is installed.



Edward Duhe is a Marketing Support Representative in IBM's Personal Systems Competency Center in Roanoke, Texas. He provides technical support for LAN Distance and OS/2

LAN Server. Edward joined IBM in 1987 as a Systems Engineer in South Louisiana. He has a BS degree in Business Administration from Louisiana State University. Direct your questions and comments about this article to Duhe' at eduhe@vnet.ibm.com or to psts@vnet.ibm.com.

For Personal Systems services call (800) 547-1283.

Are You Moving?

Please let us know when you change your address. Just fill in your new address at right, and either

- 1) fax it to **(817) 961-7218**, or
- 2) mail it to **Personal Systems
IBM Corporation
Internal Zip 40-B3-04
One East Kirkwood Blvd.
Roanoke, Texas 76299-0015**

Name _____

Company _____

Address _____

City _____ State _____ Zip _____

Phone (____) _____

(Please include old mailing label if possible.)

DB2/2—More Than Ever Before!

DB2/2, IBM's relational database engine for the OS/2 platform, has become a much stronger offering with the introduction of many enhancements and the availability of a number of products that improve its value. This article looks at the recent enhancements and at one of those complementary products.

IBM has had a relational database management product on the OS/2 platform since 1988, when OS/2 Extended Edition Database Manager was announced. This product has steadily become stronger, evolving from Extended Edition Database Manager, to Extended Services Database Manager, to DATABASE 2 for OS/2 (DB2/2). More and more businesses in every industry use one of these products to manage mission-critical data.

This year, the capability grows again with more enhancements to the product and with new products that will be available to make DB2/2 an increasingly useful tool in environments ranging from departmental or small business networks to enterprise-wide distributed data networks.

Nancy Miller
IBM Corporation
Roanoke, Texas

DB2/2 is a member of IBM's relational DB2 family of products now offered on a diverse set of IBM and non-IBM platforms from PCs to super computers. In conjunction with middleware for distributed databases, access-enabling

tools for data replication and decision support, tools for systems management and administration, and tools for application development, the DB2/2 database engine turns organizational data into an informational asset.

What's New in DB2/2 V1.2

DB2/2 V1.2, introduced in June 1994, is a full-function relational database management system for OS/2. When used in conjunction with Distributed Database Connection Services/2 2.2 (DDCS/2), it provides client access to IBM's host relational databases: DB2 for MVS, DB2/VSE and VM, and DB2/400.

New Packaging Options

The packaging for DB2/2 V1.2 Client/Server is now based on the maximum number of clients that will be accessing the server at any given time. The entry configuration supports up to four concurrent users.

For more concurrent users, 10-user and 50-user packs are available. One or more of the user connection packs can be ordered as required. DB2/2 V1.2 Client/Server is a prerequisite for the user packs.

For example, if you want 12 concurrent users, you would purchase DB2/2 V1.2 Client/Server (for four users), plus a 10-user pack. If you are planning for 60 concurrent users, you would purchase DB2/2 V1.2 Client/Server, plus a 50-user pack, plus a 10-user pack.

A new DB2 Enterprise Connection Package is also available. It combines the DB2/2 V1.2 Client/Server and DDCS/2 V2.2 Multi-User Gateway products into a single package.

Enhanced Client-Enabling Support

DB2 Client Application Enabler/DOS (CAE/DOS) V1.2 and DB2 Client Application Enabler/2 (CAE/2) V1.2 are included with DB2/2 V1.2 Client/Server. They replace the Distributed Client Enabler feature that came with DB2/2 V1.0. Following are some of their features:

■ **Access to all of the IBM relational databases.** A DOS, Windows, or OS/2 workstation running CAE/DOS V1.2 or CAE/2 V1.2 can directly access data stored in DB2/2, DB2/6000, or DB2 for HP-UX. In addition, when going through a DDCS/2 V2.2 gateway, CAE clients can access any of the IBM host relational databases.

■ **IPX/SPX support.** DOS, Windows, and OS/2 users on a Novell NetWare LAN can now access DB2/2 database servers and (through DDCS/2) host relational databases, using the internet packet exchange/sequenced packet exchange (IPX/SPX) communication protocol. Since the IPX/SPX protocol is already used to connect to a Novell NetWare local area network (LAN), management of IBM relational databases in the NetWare LAN environment is simplified.

Other protocols supported by CAE/DOS and CAE/2 include NetBIOS, transmission control protocol/internet protocol (TCP/IP), and advanced program-to-program communication (APPC).

Figure 1 shows protocol choices that can be made to access databases on DB2/2, DB2/6000, and DB2 for HP-UX.

■ **Virtual DOS machine and WIN-OS/2 support.** DOS and Windows applications running in the DOS or WIN-OS/2

sessions of OS/2 can now access DB2/2 V1.2 databases on the same workstation. No communication protocol or adapter card is necessary.

■ **ODBC driver.** An Open Database Connectivity (ODBC)-compliant driver is included to support Windows desktop applications developed using the ODBC specification. Numerous popular Windows database applications in the marketplace conform to the ODBC specification. These applications can now be used with CAE/DOS V1.2 to access data in any of the DB2 family of relational databases.

■ **Client-based application development.** With earlier versions of DB2/2, all application development took place on the server or on a stand-alone DB2/2 workstation. Now, database applications can be developed on CAE/2 or CAE/DOS clients using either IBM DATABASE 2 Software Developer's Kit/2 V1.2 or IBM DATABASE 2 Software Developer's Kit/DOS V1.2. These products include precompilers and tools as well as documentation and code samples.

Unlike the Distributed Client Feature for DB2/2 V1.0, the CAE/DOS V1.2 and CAE/2 V1.2 do not require a license for every workstation. The CAE may be installed on as many workstations as necessary.

Backup

Earlier versions of the product required that no users be connected to the database whenever a backup was performed. Now, a new option has been added to the DB2/2 backup application programming interface (API) so that a backup can be performed while users are still connected. When this option is invoked, the backup waits until current transaction activity completes and the database is in a quiesced state. During this waiting period, no new transactions are permitted to start. When the database is quiesced, the backup takes place. Transactions resume after the backup has completed.

OS/2 2.1 users who want to use the online backup capability will need to apply a fix to OS/2 for Authorized Program Analysis Report (APAR) PJ12396. This fix is available from IBM Service by calling (800) 992-4777.

Client	Communication Protocols Supported	DB2/2 Server	DB2/6000 Server	DB2 for HP-UX Server
DB2 CAE/DOS V1.2 (for DOS and Windows clients)	NetBIOS IPX/SPX TCP/IP APPC	Yes Yes No No	No Yes (*) Yes No	No Yes (*) Yes No
DB2 CAE/2 V1.2 (for OS/2 clients)	NetBIOS IPX/SPX TCP/IP APPC	Yes Yes No Yes	No No Yes Yes	No No Yes No

(*) Provided by the FireFox, Inc. NOV*IX for NetWare product.

Figure 1. Protocol Options with the Client Application Enabler

SQL Enhancements

Two structured query language (SQL) enhancements increase DB2/2's compatibility with the host DB2 platforms:

- The concatenation operator (CONCAT) is now supported. It may be used to link two string operands to form an expression.
- The value scalar function returns the first non-null result in a series of two or more expressions.

Support for OS/2 for SMP

DB2/2 V1.2 users will see significant performance gains if implementing on OS/2 for Symmetrical Multiprocessing (SMP) V2.11. On this platform, the database engine is sensitive to the number of processors available and tailors its latching algorithms accordingly. Users of DB2/2 V1.2 should experience up to a 40 percent performance improvement in high-contention situations, depending on workload. Read-only workloads will scale better than update workloads—up to 1.9 on a dual-processor machine.

DB2/2 V2.0: Significant New Function

DB2/2 V1.2 provides important enhancements, but DB2/2 V2.0 will bring significant new function to the product. DB2/2 V2.0 is planned to be available to a selected set of customers in October 1994.

When these customers confirm the functional and quality objectives, IBM will announce the planned general availability.

Listed below are some of the new functions and enhancements that will make DB2/2 V2.0 an increasingly useful tool.

Support for Advanced Object-Oriented and Multimedia Applications

More and more users of relational database management systems need to store data in natural forms such as sound, video, images, graphics, animation, and formatted text. DB2/2 V2.0 will enable new applications that require access to voice, video, image, and text data stored in the database. Object-oriented techniques can be used to design and develop applications.

Several new features make DB2/2 conducive to object-oriented and multimedia applications:

■ **Support for Large Objects (LOBs).** LOBs permit users to store very large binary or text objects in a database. Binary LOBs (yes, they're called BLOBs) can be used for multimedia objects such as video, images, and voice. A set of powerful built-in functions—such as search, substring, and concatenation—is supported for large objects. With user-defined types and functions, LOBs can be used to manage information with a more complex internal structure.

Large objects can be useful for extending the breadth of many business applications. For example, a bank might wish to store a customer's signature in a relational table along with other information about the customer.

■ **User-Defined Types (UDTs).** UDTs let users define new data types based on built-in types provided by DB2/2 such as integers, decimals, or character strings. The newly defined types, however, assume different semantics

STOP!

Have you used the reader service card to request fast, free information about the products and services advertised in *Personal Systems*?

NO? With the heavy traffic of new technology to choose from in the personal computer market, you need to know about all the most recent developments.

Caution.

- 1.** Use the advertiser's index to get the reader service numbers of the products and services for which you want to receive literature.
- 2.** Circle the same numbers on the reader service card and fill out the necessary information.
- 3.** Drop it in the mail (at no charge!), and we'll give your request the green light!

YES.
Smart move.

Copying or reprinting material from this magazine is strictly prohibited without the written permission of the editor. Titles and abstracts, but no other portions, of information in this publication may be copied and distributed by computer-based and other information-service systems.

IBM believes the statements contained herein are accurate as of the date of publication of this document. However, IBM hereby disclaims all warranties as to materials and workmanship, either expressed or implied, including without limitation any implied warranty of merchantability or fitness for a particular purpose. In no event will IBM be liable to you for any damages, including any lost profits, lost savings, or other incidental or consequential damage arising out of the use or inability to use any information provided through this service even if IBM has been advised of the possibility of such damages, or for any claim by any other party.

Some states do not allow the limitation or exclusion of liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This publication could contain technical inaccuracies or typographical errors. Also, illustrations contained herein may show prototype equipment. Your system configuration may differ slightly.

IBM has tested the programs contained in this publication. However, IBM does not guarantee that the programs contain no errors.

This information is not intended to be a statement of direction or an assertion of future action. IBM expressly reserves the right to change or withdraw current products that may or may not have the same characteristics or codes listed in this publication. Should IBM modify its products in a way that may affect the information contained in this publication, IBM assumes no obligation whatever to inform any user of the modifications.

Some of the information in this magazine concerns future products, or future releases of products currently commercially available. The description and discussion of IBM's future products, performance, functions, and availability are based upon IBM's current intent and are subject to change.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not imply giving license to these patents.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming, or services in your country.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever.

The articles in this publication represent the views of their authors and do not necessarily represent the views of IBM. This publication may contain articles by non-IBM authors. IBM does not endorse any non-IBM products that may be mentioned. Questions should be directed to the authors.

Publication of advertising material in this magazine does not constitute an expressed or implied recommendation or endorsement of IBM of any particular product, service, company, or technology. IBM takes no responsibility whatsoever with regard to the selection, performance, or use of any advertised products. All understanding, agreements, or warranties must take place directly between the vendor and prospective users.

than their base types, so that appropriate restrictions or behavior can be enforced. For instance, a user can define a data type of CNUMBER for customer number. This data type will be based on the system-supplied integer data type, but it cannot be multiplied, and it can be restricted within a certain range.

■ User-Defined Functions (UDFs).

UDFs, like UDTs, extend the power of the database management system. They enable users to hide the internal representation of the data from applications. Also, they enable users to define the semantics and behaviors of their UDTs. The SQL optimizer evaluates the semantics and execution cost of UDFs, just as it does built-in functions.

■ Constraints and Triggers.

Constraints can be defined to ensure that data values of a certain column fall within a specific range. For instance, salaries or employee serial numbers might be constrained within a certain range.

Triggers enable the user to instruct the database management system to automatically enforce certain business rules. The trigger is actually a piece of code that defines the rule. Then, when a certain event occurs—such as the attempt to update a column—the trigger executes automatically to enforce the rule. For example, a trigger can be written to verify that a customer's order doesn't exceed the customer's credit limit. Or, a trigger could write to other tables for audit-trail purposes.

Constraints and triggers are both designed to let the users associate more meaning with their data. These features, along with UDFs and UDTs, provide for a more active database management system. The users have the ability to build additional intelligence into the system, making it more aware of the semantics they associate with their data and therefore able to take appropriate actions.

■ Support for Recursive SQL.

The SQL optimizer has been changed to support recursive queries. The following types of queries are now possible:

- *Bills-of-material queries.* In this type of query, a user may have a parts table and may want to return subparts of parts and subparts of

the subparts, all of which are listed as parts in the same table.

- *Path expression queries.* These are used for airline reservations, where a user wants to calculate the lowest-cost plane fares on multi-hop routes.

Increased Management Capabilities for Very Large and Highly Available Databases

A number of enhancements to DB2/2 V2.0 greatly increases its ability to manage data integrity and availability as databases grow ever larger. These enhancements allow users to create and effectively manage databases that may be terabytes in size.

- **Database Tablespaces.** Database administrators will be able to partition a DB2/2 V2.0 database into parts called *tablespaces*. When users create tables, they may specify the names of the *base* tablespaces for most table data; *index* tablespaces for storing indexes; and *long* tablespaces for storing LOB data. Thus, indexes and LOBs can be kept separate from the rest of the data. An administrator may place data, indexes, and LOBs, all of which are in the same table, in separate tablespaces for performance or operational reasons.

- **Flexible Space Allocation.** An administrator can increase the size of a tablespace by adding more storage space while the database is in operation. The data in the tablespace is automatically rebalanced for optimal performance.

- **Backup and Recovery Enhancements.** Backup and recovery may be performed at the tablespace level or at the database level. Depending on how the tablespaces have been structured, this could be equivalent to a table-level backup or recovery. Tablespaces can be backed up while either online or offline. During recovery, all tablespaces other than the one being recovered can remain online.

Backup and recovery of a database or tablespace can be performed in parallel to or from multiple devices, thus reducing the time required.

- **High-Speed Load Utility.** The new load utility greatly increases the speed of doing data loads and ensures the recoverability of the data being loaded. It has been added to the product to

facilitate the bulk loading of new tables and for appending large amounts of data to existing tables. This utility can optionally build indexes and gather statistics as well as load data.

Distributed Data Access

Administrators implementing DB2/2 V2.0 with several accompanying products will find that they have a new level of flexibility in distributing data and applications across platforms. A number of enhancements to DB2/2 facilitate that flexibility:

- **Distributed Relational Database Architecture (DRDA) Application Server.** This long term simply means that data on DB2/2 may be accessed from database applications running on Multiple Virtual Storage (MVS), Virtual Machine (VM), and OS/400 platforms.

- **DRDA Stored Procedures.** The support that already exists for stored procedures in DB2/2 client/server databases is being extended to the DRDA environment. Thus, a client application may invoke a stored procedure that executes on the DRDA Application Server platform, reducing the communication costs between client and server.

- **Distributed Unit of Work (DUOW).** DUOW function, also called *two-phase commit*, provides the capability for an application to read or update tables in more than one database from within a single unit of work with full data integrity.

Performance Enhancements

Several enhancements to DB2/2 will bring significant improvements to performance in V2.0.

- **SQL Optimizer with Query Rewrite Capability.** Database Manager and DB2/2 have from the beginning had a sophisticated cost-based optimizer based on the design of the DB2 for MVS optimizer. Now the demands on the optimizer are greater than ever because of the new application domains enabled by the object-oriented extensions. Major enhancements have been made to the optimizer to accommodate the new extensions to the product.

The optimizer has been redesigned to evaluate more possibilities before choosing an access path to the data. New, more sophisticated techniques

are used to model the costs of the various alternatives.

Many front-end applications that are popular today use a point-and-click graphical interface to create *ad hoc* queries. The queries created this way can be quite complex and can use a large amount of system resources to execute. Because of the trend toward very complex database queries, the optimizer has been modified to incorporate a query rewrite phase. This phase automatically transforms an inefficient query into a faster executing query. The user benefits with the best possible performance.

- **Explain Facility.** The new Explain facility gives the user detailed information about the access plan that the optimizer has chosen to process SQL statements. The user may then use this information to make decisions about database design and to determine how that design can be changed to improve performance.

In Database Manager and DB2/2 V1.0, Explain was included as a productivity aid, or applet, and it generated a text report. The new Explain in DB2/2 V2.0 is integrated into the product as a fully supported feature; it supports dynamic as well as static SQL and offers the user more detailed information about access plans.

- **I/O Prefetch.** I/O prefetch includes several techniques that have long been available to users of DB2 for MVS to improve performance of queries that require a large amount of I/O. In such cases, it is beneficial for the database manager to read sections of the table into the buffer pool in parallel with the processing of the data. The performance improvements will be especially noticed on machines where there is a gap between the speeds of the CPU and the hard disks.

The techniques for I/O prefetch include *sequential prefetch*, which is generally performed on clustered data (data that is already more or less in the order in which it is to be accessed). The purpose of sequential prefetch is to prevent an I/O bottleneck by getting pages into the buffer pool before they are required. *List prefetch* is a similar technique that is used for non-clustered data. *Parallel I/O* is the ability to perform many I/O

operations in parallel on behalf of a single query. *Big-block read* is the ability to read several 4 KB pages in a single I/O operation. All these techniques bring significant performance benefits.

Database Management Enhancements

As databases become larger and more complex, tools to aid in management and administration become crucial. Following are some of the new tools that facilitate database systems management in DB2/2 V2.0:

- **Database Event Monitor.** Event monitors are new database objects that can be created to monitor transient events within the database. Users will be able to monitor database and summary information, connections, SQL statement information, and deadlocks. For example, a monitor could be used to record the details of deadlocks as they occur. That information could then be used to make design changes in order to avoid deadlocks in the future.

*... a new level of flexibility
in distributing data and
applications. ...*

- **Quiesce.** Database administrators need a way to ensure that all database activity has been stopped in order to perform activities such as online backup. The Quiesce command provides that capability.

- **Remote Administration.** In previous versions, many administrative tasks could be performed only at the server. Now, these tasks, such as initiating a backup, can be performed from a client. A single client workstation can perform administrative tasks for multiple remote servers.

- **History File.** Another enhancement to the database management function is the creation of a history file for each database. Operational activities, including backups, recoveries, and table loads, will be recorded in this file.

Visualizer Family: An Exciting New Front End

All the new strengths and capabilities

within DB2/2 require tools for accessing and manipulating data that will exploit those new strengths. The Visualizer product family is IBM's answer to that demand. It is a client/server toolset for query, analysis, and presentation of data from relational databases. The workstation products include Query, Charts, Procedures, Plans, Statistics, Ultimedia Query, and Development. Each of these products is discussed below.

Visualizer Query V1.0 for OS/2 is the first member of the family to be available. Visualizer Query is the core product of the family and can be used not only with DB2/2, but also with DB2/6000, DB2/400, DB2/MVS, and DB2/VSE and VM.

Visualizer Query will let the user perform queries, browse data, update databases, and generate reports. Visualizer Query has a simple graphical user interface that masks the complexities of the relational database environment. It is a decision-support tool as well as a query tool and is designed for use by business professionals such as managers, salespeople, planners, and administrators who need access to corporate, departmental, and personnel data.

Figure 2 shows an example of the point-and-click query interface. The data can also be accessed directly by using the SQL language, if the user desires.

Visualizer Charts is an optional product that can be used with Visualizer Query to present the results of a query in an attractive chart. Chart types include bar, line, surface, scatter, pie, and composite. From within Charts, users can calculate columns, calculate percentages, and select total, minimum, maximum, and average values.

Charts can be printed or exported for use by other products using dynamic data exchange (DDE), the OS/2 clipboard, and standard industry formats such as OS/2 Metafile, OS/2 Bitmap, .PCX, .TIF, and .GIF files. Figure 3 illustrates a chart generated from DB2/2 data using Visualizer Charts.

Visualizer Procedures is a tool that allows you to automate repetitive tasks. For instance, you may want to run a report every day at a certain time and

generate a chart based on that report. Procedures lets you build the sequence of tasks and link them in the order you need. The procedure you created can be run on a regular timed basis or after business hours.

A procedure is built by using a mouse to drag required icons on the desktop and to draw links between them. The links define the logical flow through the procedure. The procedure itself can then be saved as an icon.

Several other members of the Visualizer family add rich function to the base query product. *Visualizer Plans* is a tool for building multidimensional computer models suitable for solving a wide range of business and organizational problems. For instance, you could compare your business performance in terms of time and geography. *Visualizer Statistics* is a powerful tool for applying many different types of statistical analysis against your data. *Visualizer Development* allows you to easily develop custom-made applications for accessing your relational data.

One of the most exciting new tools is *Visualizer Ultimea Query*. Using Ultimea Query, you can query and display traditional data—such as numbers and text—with related images, video, audio, and documents. Thus, facts and figures can be linked to related multimedia information and all can be displayed on the desktop.

Multimedia objects may be searched by content. For instance, you can find images by color, shape, texture, and layout. This innovative new technology is called “query-by-image-content.”

More Than Ever Before

DB2/2 Versions 1.2 and 2.0 bring more capabilities to manage more data in new, more flexible ways. Using Visualizer, you have the power to access and manipulate that data. In future articles, we'll look at managing data in the distributed environment.

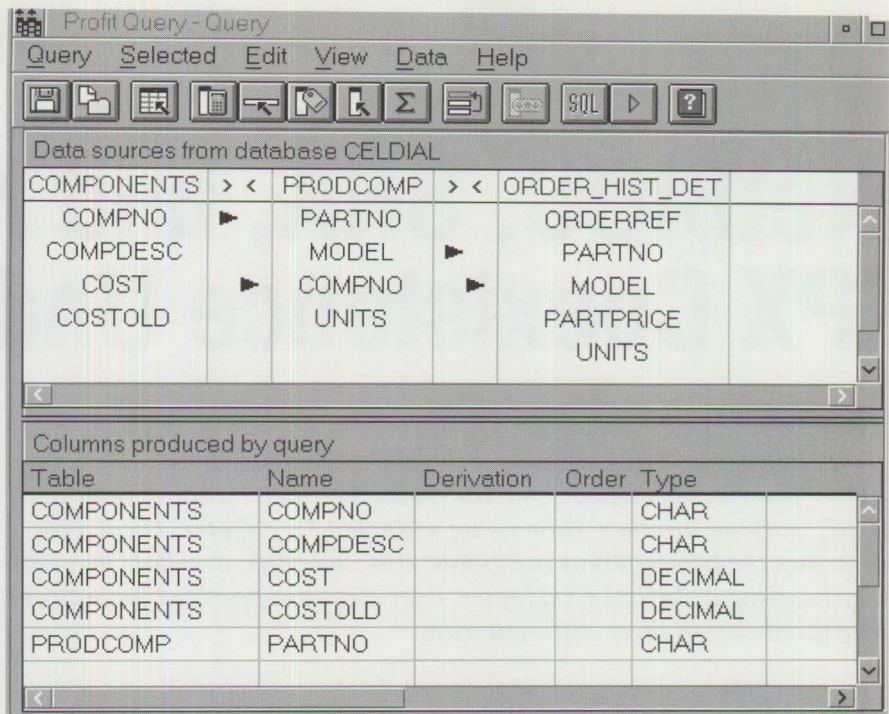


Figure 2. Visualizer Query Interface

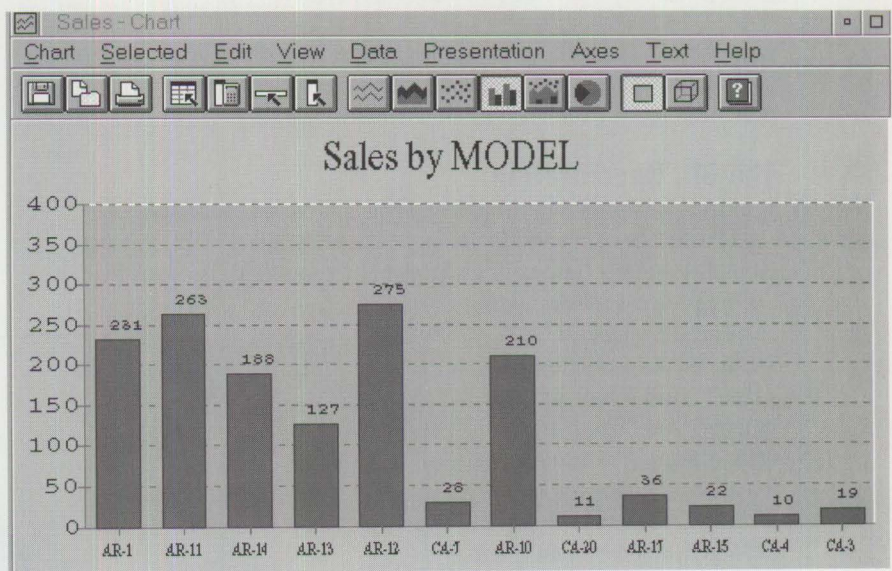


Figure 3. Visualizer Charts



Nancy Miller is an Associate Marketing Support Representative in IBM's Personal Systems Competency Center in Roanoke, Texas, providing technical support and consulting for database products on the OS/2 platform. She has supported DB2/2 and OS/2 Database Manager since 1989.

For Personal Systems services call (800) 547-1283.

NetBIOS, SNA, and NetWare IPX Coexistence Under OS/2

This article is intended to help you better understand coexistence in an OS/2 environment. Coexistence is the ability to have both the Network Driver Interface Specification (NDIS) and the Open Datalink Interface (ODI) working together while using a single LAN adapter. The article describes the two different interfaces, ODI and NDIS, and how they can coexist using a single LAN adapter. Easy steps are provided for setting up coexistence in three different ways.

Coexistence is important for running ODI-based applications (such as Novell NetWare Requester for OS/2) with NDIS-based applications (such as IBM LAN Server, LAN Requester, and Communications Manager/2 [CM/2]) when you have only a single LAN adapter.

Both the ODI and NDIS specifications have similar components: protocols, LAN adapter drivers, and protocol/LAN adapter driver managers. These specifications and their components are discussed below.

The ODI Specification

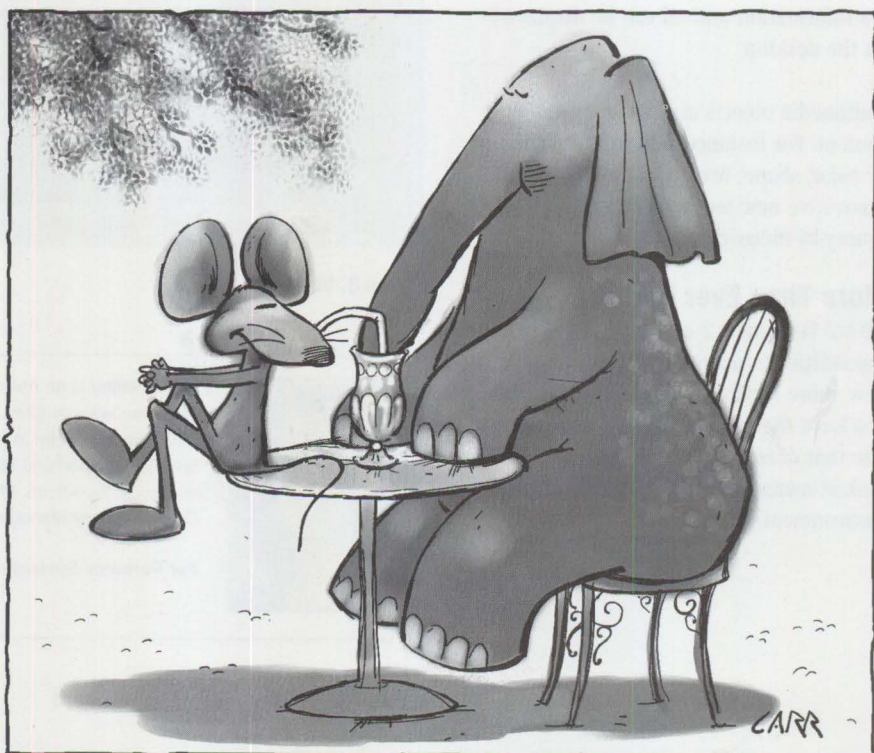
The ODI specification, developed by Apple and Novell, is the standard used for Novell's NetWare client and file server products. ODI supports multiple protocols (Internetwork Packet eXchange [IPX], transmission control protocol/internet protocol [TCP/IP], and AppleTalk), and it enables different LAN adapters (token-ring, ARCNet, Ethernet, and fiber distributed data interface [FDDI]) to use different or multiple protocols. IPX is the default protocol used with NetWare file servers and clients.

Albert Juarez
IBM Corporation
Roanoke, Texas

Figure 1 illustrates the ODI specification. The LAN adapter has a driver called an *ODI Multiple-Link Interface Driver (MLID)*. The MLID receives a packet and puts it into an Event Control Block (ECB), which is then

passed to the Link Support Layer (LSL). The LSL routes packets from the MLID layer to the proper protocol stacks. The LSL has a data segment that contains protocol stack information, binding information, and ECB information. The protocol stacks are IPX (IPX.SYS) and TCP/IP (TCPIP.SYS).

NetWare Requester for OS/2 is based on the ODI specification. When the Requester is installed (using the utility INSTALL.EXE), it typically has a set of defaults for its configuration. To change these defaults, edit the NET.CFG configuration file, which is located in the root directory of the drive on which NetWare Requester and OS/2 are installed. Specifically, for coexistence, change the following parameters:



■ **Link Support**—This heading adjusts the number and size of communication buffers for ODI. Under the Link Support heading is the statement `Buffers 14 4202` where 14 is the number of buffers and 4202 is the buffer size. These parameters adjust the number and size of communication buffers for ODI; therefore, changing these parameters will affect the packet size negotiated between the file server and the workstation.

■ **Link Driver name**—In this heading, name represents an MLID such as token. This parameter configures ODI for your LAN adapter. Under this heading, you can define the ODI-compliant protocols to be used for the adapter. The default protocol is IPX.

■ **Protocol [protocol]**—To set up coexistence with NDIS, use `Protocol ODINSUP`.

For more information about these parameters and other `NET.CFG` parameters, see the *NetWare Client for OS/2 User's Guide* or the article "NetWare Requester for OS/2, v2.01: Features and Installation" in the March/April 1994 issue of this magazine.

The NDIS Specification

Figure 2 depicts the NDIS specification, which was developed by 3Com and Microsoft, and is used by IBM products such as CM/2, LAN Server, and TCP/IP for OS/2. The LAN adapter has a driver (called an NDIS *Media Access Control* [MAC] driver), a protocol manager, and NDIS protocol stacks. The NDIS MAC driver controls the LAN adapter and interfaces with the protocols.

The NDIS MAC driver for token-ring is `IBMTOK.OS2`. The NDIS protocol stacks are NetBIOS (`NETBEUI.OS2`, `NETBIOS.OS2`), IEEE 802.2 (`LANDD.OS2`, `LANDLDD.OS2`), and TCP/IP (`INET.SYS`, `IFNDIS.SYS`).

The protocol manager is `PROTMAN.OS2`. It configures NDIS using the binding program `NETBIND.EXE`, and it receives information from the NDIS configuration file `PROTOCOL.INI`.

`PROTOCOL.INI` consists of four sections:

■ **Protocol manager [PROT_MAN]**, which defines the protocol manager

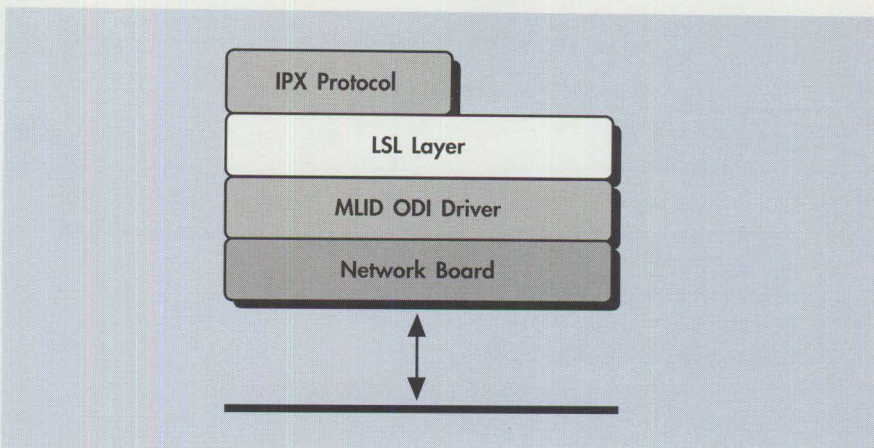


Figure 1. The ODI Specification

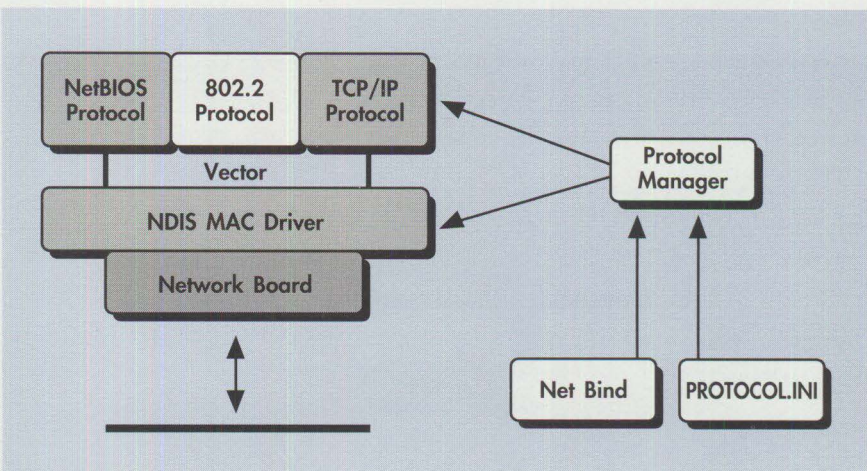


Figure 2. The NDIS Specification

- LAPS configuration information [`IBMLXCFG`], which defines protocol and MAC drivers
- Protocol driver [`NETBEUI_nif`], which defines driver name and bindings
- MAC driver [`IBMTOK_nif`], which defines the driver name and configuration

Each section has additional configuration information that can be changed. Refer to the *Network Transport Services/2 LAN Adapter and Protocol Support (LAPS) Configuration Guide* (S96F-8489) for this additional information.

IBM's CM/2, LAN Server, and TCP/IP for OS/2 products contain the LAN Adapter and Protocol Support (LAPS) program, which sets up the NDIS LAN adapter driver and the protocols needed for the applications.

Coexistence Using NetWare Drivers

The NetWare Client Kit for OS/2 (NetWare Requester for OS/2) comes with two drivers, `ODINSUP.SYS` and `LANSUP.SYS`, which set up coexistence in two different ways.

NetWare ODINSUP

NetWare's `ODINSUP.SYS` is an OS/2 ODI-based driver that enables the NDIS interface to coexist with the ODI interface (see Figure 3). The ODI interface controls the LAN adapter. `ODINSUP.SYS` looks like a protocol driver to the LSL layer, but it also looks like an NDIS MAC driver to the NDIS interface. `ODINSUP` enables the translation of packets going to and from the NDIS interface, while `ODINSUP.SYS` allows the NetWare Requester, using IPX, to work simultaneously with IBM's LAN Requester (using NetBIOS) and IBM's CM/2 (using IEEE 802.2).

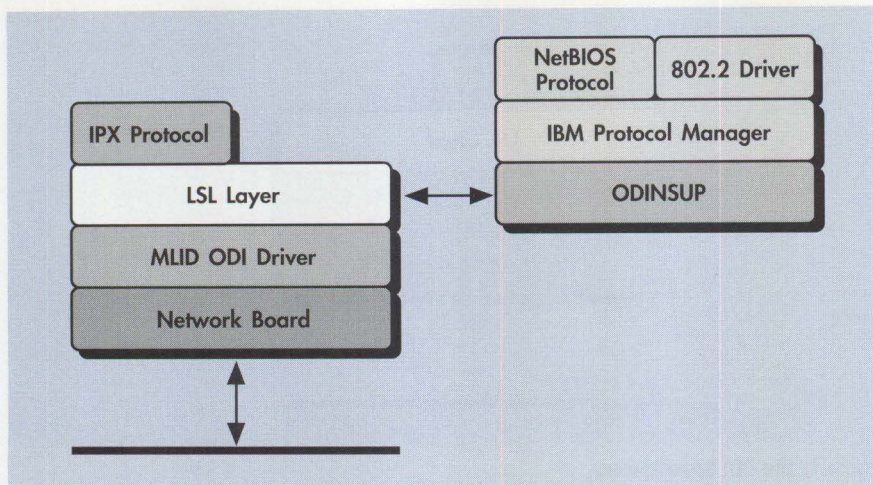


Figure 3. ODINSUP Architecture

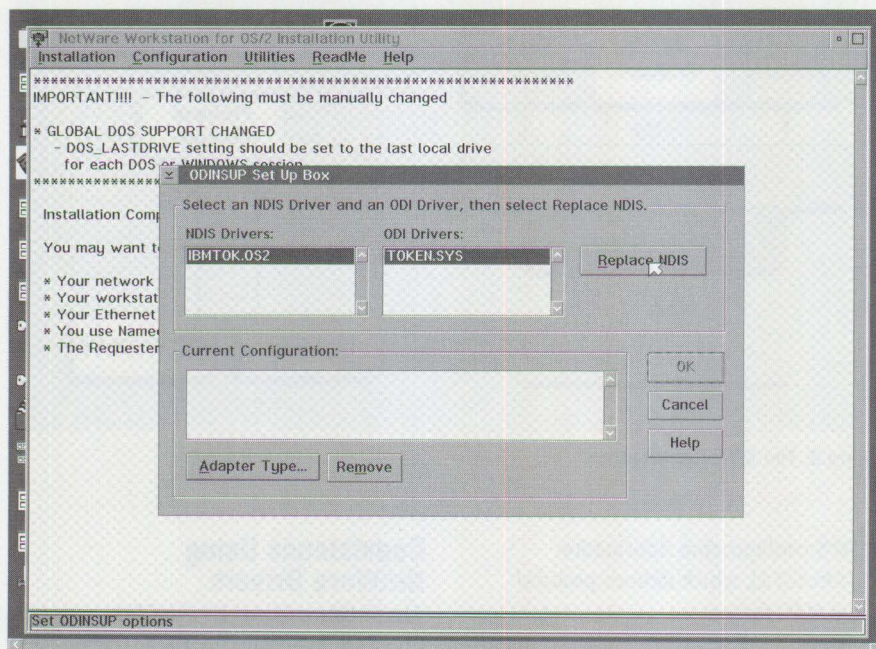


Figure 4. ODINSUP Setup

The steps for installing ODINSUP under OS/2 2.1 or 2.11 are:

1. Test the NDIS interface and its applications to verify that they work.
2. Install the NetWare Requester, choosing an MLID and ODINSUP.SYS.
3. Reboot.

Using the LAPS program, configure your workstation to work with LAN Server and/or CM/2. Make sure that the NDIS interface and NDIS applications are operational. Once all NDIS-based applications are working, continue by installing the NetWare Requester for OS/2. This installation is discussed in the *NetWare*

Requester User's Guide and in the article "NetWare Requester for OS/2, v2.01: Features and Installation" in the March/April 1994 issue of this magazine.

During NetWare Requester for OS/2 installation, be sure to choose the right MLID for your LAN adapter card. In the NetWare Requester for OS/2 main installation screen, select Utilities, then ODINSUP Setup (see Figure 4). This tells the installation program to configure for ODINSUP. After the installation finishes, exit the installation screen, then reboot the system.

These steps automatically modify the following files:

C:\IBMCOM\PROTOCOL.INI

C:\CONFIG.SYS

C:\NET.CFG

Therefore, there is no need to manually change these files. In fact, it is best not to manually change them because you may cause your NDIS applications or the NetWare Requester to fail.

Figure 5 shows the contents of the configuration files for ODINSUP.SYS. (Not all of the information in these files is provided.)

NetWare LANSUP

Another coexistence OS/2 ODI driver supplied by NetWare Requester for OS/2 is LANSUP.SYS. Unlike the ODINSUP.SYS driver, LANSUP.SYS uses the NDIS interface to control the LAN adapter and interfaces to ODI. Figure 6 illustrates the architecture of LANSUP.

LANSUP.SYS was written to the IEEE 802.2 specification. This driver was originally intended for non-NDIS configurations such as early LAN Server and Extended Services (Communications Manager) configurations that used a logical link control (LLC) subsystem (before IBM provided LAPS with the NDIS interface). When the LAPS NDIS-based interface was designed and the LLC subsystem was added, provisions were made to continue the use of LANSUP.SYS.

To install the LANSUP.SYS driver in an OS/2 2.1 or 2.11 environment, use the NetWare Requester installation program. The steps for installing the LANSUP.SYS driver are:

1. Make sure NDIS is set up.
2. Make sure that the NDIS interface and the applications that use the NDIS interface work.
3. Run the NetWare Requester for OS/2 Installation program, select the LANSUP.SYS driver, and then select Configure this Workstation to edit the NET.CFG file.
4. Reboot the system.

Figure 7 lists the configuration files for LANSUP.SYS.

CONFIG.SYS with ODINSUP

```

DEVICE=C:\NETWARE\LSL.SYS
RUN=C:\NETWARE\DDAEMON.EXE
REM - ODI-Driver Files BEGIN -
DEVICE=C:\NETWARE\TOKEN.SYS
REM - ODI-Driver Files END -
DEVICE=C:\NETWARE\ROUTE.SYS
DEVICE=C:\NETWARE\ODINSUP.SYS
DEVICE=C:\NETWARE\IPX.SYS
DEVICE=C:\NETWARE\SPX.SYS
RUN=C:\NETWARE\SPDAEMON.EXE
rem DEVICE=C:\NETWARE\NMPIPE.SYS
rem DEVICE=C:\NETWARE\NPSEVER.SYS
rem RUN=C:\NETWARE\NPDAEMON.EXE
DEVICE=C:\NETWARE\NWREQ.SYS
IFS=C:\NETWARE\NWIFS.IFS
RUN=C:\NETWARE\NWDAEMON.EXE
rem DEVICE=C:\NETWARE\NETBIOS.SYS
rem RUN=C:\NETWARE\NBDAEMON.EXE
DEVICE=C:\OS2\MDOS\LPTDD.SYS

```

Note: In CONFIG.SYS, the TOKEN.SYS MLID and the ODINSUP.SYS driver are added by NetWare Requester for OS/2. Not listed is the removal of the NDIS MAC driver (IBMTOK.OS2) for the LAN adapter; if it were present, it would conflict with the ODI MLID (TOKEN.SYS) for the LAN adapter.

NET.CFG with ODINSUP

```

link driver TOKEN
  node address 10005AAB00
  frame token-ring
  frame token-ring_snap

protocol ROUTE
  source route board 1
  source route board 2

protocol ODINSUP
  bind TOKEN

```

Note: In NET.CFG, in the link driver section, TOKEN.SYS is for the token-ring MLID. The protocol bound to TOKEN.SYS is ODINSUP.

PROTOCOL.INI with ODINSUP

```

[PROT_MAN]
  DriverName = PROTMAN$

[IBMLXCFG]
  LANDD_nif = LANDD.nif
  NETBEUI_nif = NETBEUI.nif
  IBMTOK_nif = IBMTOK.nif

[LANDD_nif]
  DriverName = LANDD$
  Bindings = TOKEN

[NETBEUI_nif]
  DriverName = NETBEUI$
  Bindings = TOKEN

[IBMTOK_nif]
  DriverName = IBMTOK$

[TOKEN]
  DriverName = TOKEN$

```

Note: In PROTOCOL.INI, all the Bindings = statements have TOKEN as the driver to bind to. Normally, LAPS uses the IBMTOK\$ driver for this statement. However, because ODINSUP setup was done last, the NetWare Requester installation program:

- Altered the PROTOCOL.INI file
- Added a MAC driver section [TOKEN] at the bottom, even though it is the name of the ODI TOKEN driver
- Added a line space at the top and at the bottom of the MAC driver section

Figure 5. ODINSUP Configuration Files

Coexistence Using IBM Driver ODI2NDI.OS2

A third way to set up coexistence is to use IBM's OS/2 NDIS driver, ODI2NDI.OS2. This driver is supplied with the LAPS program that comes with Network Transport Services/2 (NTS/2) in the latest IBM LAN Requester and CM/2 products.

Like the LANSUP.SYS driver, the ODI2NDI driver allows the ODI interface to coexist with the NDIS interface. The NDIS interface is in control of the LAN adapter, using the NDIS MAC driver (see Figure 8). However, unlike the LANSUP.SYS driver, ODI2NDI.OS2 acts as an NDIS protocol driver, and it interfaces directly with the protocol manager.

ODI2NDI is configurable through the LAPS program. Within the LAPS program, select NetWare Requester Support as one of the protocols to add to the NDIS MAC driver. The following steps will ensure proper installation of ODI2NDI coexistence in an OS/2 2.1 or 2.11 environment:

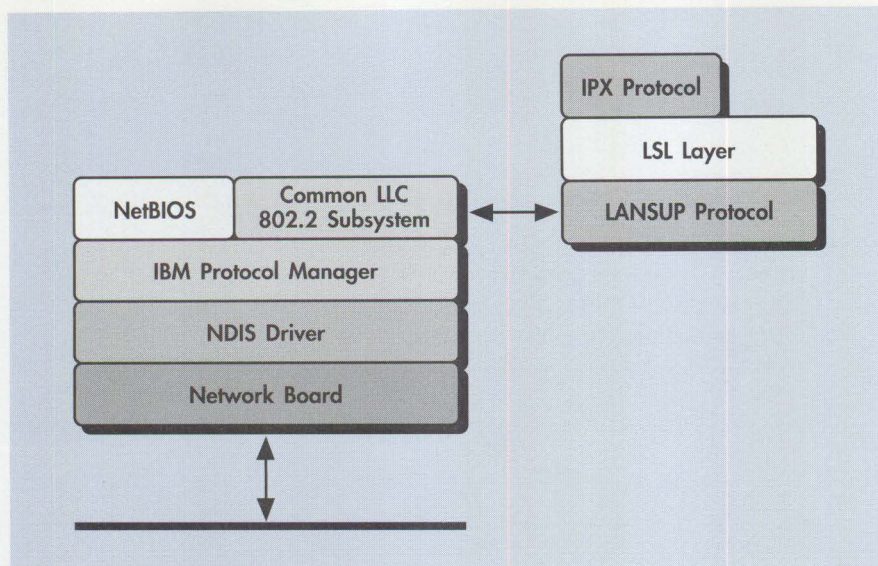


Figure 6. LANSUP Architecture

CONFIG.SYS with LANSUP

```

DEVICE=D:\NETWARE\LSL.SYS
RUN=D:\NETWARE\DDAEMON.EXE
REM - ODI-Driver Files BEGIN -
DEVICE=D:\NETWARE\LANSUP.SYS
REM - ODI-Driver Files END -
DEVICE=D:\NETWARE\ROUTE.SYS
DEVICE=D:\NETWARE\IPX.SYS
DEVICE=D:\NETWARE\SPX.SYS
RUN=D:\NETWARE\SPDAEMON.EXE
rem DEVICE=D:\NETWARE\NMPPIPE.SYS
rem DEVICE=D:\NETWARE\NPSEVER.SYS
rem RUN=D:\NETWARE\NPDAEMON.EXE
DEVICE=D:\NETWARE\NWREQ.SYS
IFS=D:\NETWARE\NWIFS.IFS
RUN=D:\NETWARE\NWDAEMON.EXE
rem DEVICE=D:\NETWARE\NETBIOS.SYS
rem RUN=D:\NETWARE\NBDAEMON.EXE
DEVICE=D:\OS2\MDOS\LPTDD.SYS

```

Note: In CONFIG.SYS, LANSUP.SYS, used in place of the MLID, was added by the NetWare Requester installation program. Not listed is the IBM-TOK.OS2 driver, which is also loaded for the NDIS protocol manager added by LAPS.

NET.CFG with LANSUP

```

link driver LANSUP
  node address 1005AA43808
  frame token-ring
  frame token-ring_snap
link support
  buffers 14 4202

```

Note: In NET.CFG, the link driver statement uses the LANSUP driver. The link support section is optional, for performance enhancements.

PROTOCOL.INI with LANSUP

Add the following to PROTOCOL.INI under the heading [IBMTOK_nif]:

```

[IBMTOK_nif]
  MAXBUFFSIZE = 4208

```

[IBMTOK_nif] is already in the PROTOCOL.INI file, but MAXBUFFSIZE is not. The maximum buffer size ensures that the workstation can utilize a 4 KB packet when using a token-ring board running at a data-transfer rate of 16 Mbps. Also, make sure the token-ring board is set up with a RAM segment size of 16 KB or higher.

Figure 7. LANSUP Configuration Files

1. Make sure the NDIS interface and applications are operational.
2. Run the NetWare Requester for OS/2 installation.
3. Do not reboot yet.
4. Run the LAPS program and select NetWare Support (ODI2NDI).
5. Make sure to enter LAA or UAA for Netware Requester Support driver.
6. Exit and reboot.

Install LAPS and configure for IBM OS/2 NetBIOS and IBM IEEE 802.2. Test your NDIS interface and applications that use the NDIS interface to make sure they all work. Next, run the NetWare Requester installation program (INSTALL.EXE). Select TOKEN.SYS as your MLID. After the installation finishes, exit the program. *Don't reboot yet!*

Now, go back into the LAPS program (C:\IBMCOM\LAPS.EXE), select Configure, select Configure LAN transports, and click on Continue. You see a screen like the one in Figure 9. Add IBM NetWare Requester Support to the IBM Token-Ring Network Adapter MAC driver. Highlight the IBM NetWare Requester Support option, and click on Edit. Fill in the Network adapter address with either UAA or LAA. Then exit LAPS; you will be prompted to save the changes, after which the program will finish. At this point, you can reboot!

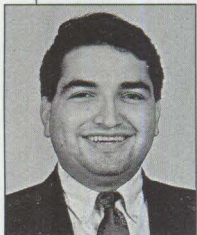
You could not reboot after installing the NetWare Requester because it was configured for ODI only, using TOKEN.SYS. By running the LAPS program and adding the NetWare Requester Support option, you change the configuration files (CONFIG.SYS and PROTOCOL.INI) to ODI2NDI.OS2. Figure 10 lists the configuration files for ODI2NDI.OS2.

Choosing Your Method of Coexistence

One of the most frequent questions I get from customers is "Which coexistence driver should I choose: ODINSUP.SYS, LANSUP.SYS, or ODI2NDI.OS2?" If the majority of your traffic is NetWare traffic, use ODINSUP.SYS. If most of your traffic comes from Systems Network Architecture (SNA) or NetBIOS, use either ODI2NDI.OS2 or LANSUP.SYS. Because LANSUP.SYS goes through the the IEEE

802.2 LLC subsystem, performance may be somewhat slower, so the better choice is ODI2NDI.OS2.

The main thing to keep in mind is to standardize on one method of coexistence and to stick with it. This standardization makes it easier to troubleshoot, especially if you have many workstations to support.



Albert Juarez is a member of the Workgroup LAN Systems Support and Services Group in IBM's Personal Systems Competency Center (PSCC) in Roanoke, Texas.

Albert's mission is to support Novell NetWare products in IBM environments, specializing in NetWare itself, OS/2 base with NetWare, and the coexistence of NetWare, IBM LAN Server, and IBM Communications Manager/2 under OS/2. He has over seven years of experience in the NetWare environment. Albert is a Novell Enterprise CNE, an IBM OS/2 Engineer, and has taught Skill Dynamics courses about NetWare. He has an Associate degree in Electronic Engineering Technology from DeVry Institute.

For Personal Systems services call (800) 547-1283.

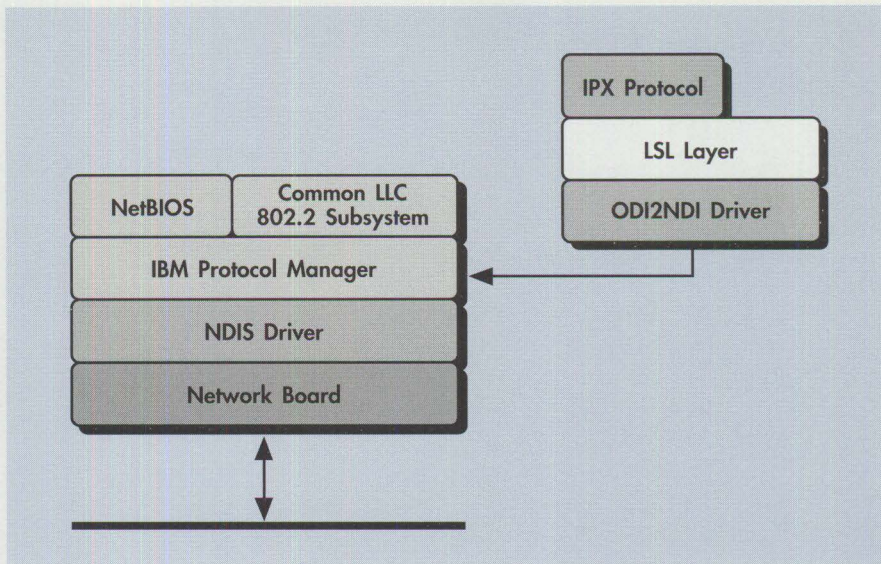


Figure 8. ODI2NDI Architecture

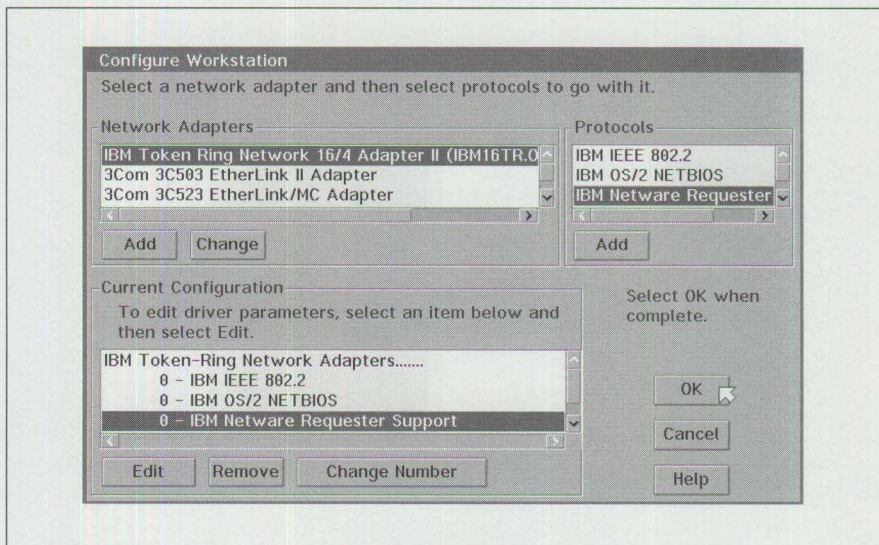


Figure 9. LAPS Configuration Screen

CONFIG.SYS with ODI2NDI

```

DEVICE=D:\NETWARE\LSL.SYS
RUN=D:\NETWARE\DDAEMON.EXE
DEVICE=D:\IBMCOM\PROTOCOL\ODI2NDI.OS2
REM - ODI-Driver Files BEGIN -
REM DEVICE=D:\NETWARE\TOKEN.SYS
REM - ODI-Driver Files END -
DEVICE=D:\NETWARE\ROUTE.SYS
DEVICE=D:\NETWARE\IPX.SYS
DEVICE=D:\NETWARE\SPX.SYS
RUN=D:\NETWARE\SPDAEMON.EXE
rem DEVICE=D:\NETWARE\NMPIPE.SYS
rem DEVICE=D:\NETWARE\NPSEVER.SYS
rem RUN=D:\NETWARE\NPDAEMON.EXE
DEVICE=D:\NETWARE\NWREQ.SYS
IFS=D:\NETWARE\NWIFS.IFS
RUN=D:\NETWARE\NWDAEMON.EXE
rem DEVICE=D:\NETWARE\NETBIOS.SYS
rem RUN=D:\NETWARE\NBDAEMON.EXE
DEVICE=D:\OS2\MDOS\LPTDD.SYS

```

Note: In CONFIG.SYS, the TOKEN.SYS MLID was REMarked out. This was done by the LAPS program when the ODI2NDI.OS2 driver was added, and it happened because you selected NetWare Requester Support in LAPS. Not listed is the IBMTOK.OS2 MAC driver, which was also added by the LAPS program.

Figure 10. ODI2NDI Configuration Files (continued on page 66)

PROTOCOL.INI with ODI2NDI

```
[PROT_MAN]
    DriverName = PROTMAN$
[IBMLXCFG]
    LANDD_nif = LANDD.nif
    NETBEUI_nif = NETBEUI.nif
    ODI2NDI_nif = ODI2NDI.nif
    IBMTOK_nif = IBMTOK_nif
[LANDD_nif]
    DriverName = LANDD$
    Bindings = IBMTOK_nif
[NETBEUI_nif]
    DriverName = NETBEUI$
    Bindings = IBMTOK_nif
[ODI2NDI_nif]
    DriverName = ODI2NDI$
    Bindings = IBMTOK_nif
    TOKEN-RING = "yes"
    TOKEN-RING_SNAP = "no"
    ETHERNET_802.3 = "no"
    ETHERNET_802.2 = "no"
    ETHERNET_II = "no"
    ETHERNET_SNAP = "no"
    TRACE = 0x0
[IBMTOK_nif]
    DriverName = IBMTOK$
```

Note: In PROTOCOL.INI, the MAC driver is IBMTOK. In the [IBMXCFG] section, the ODI2NDI driver has been added.

NET.CFG for ODI2NDI Using Buffers 14 4202

```
link support
    buffers 14 4202
```

The link support section is optional, for performance enhancements. Add MAXBUFFSIZE to PROTOCOL.INI under the heading [IBMTOK_nif]:

```
[IBMTOK_nif]
    MAXBUFFSIZE = 4208
```


MAXBUFFSIZE is 4208 in PROTOCOL.INI because it must be six bytes larger than the Buffers 14 4202 in NET.CFG. The NetWare packet size is no greater than 4202. The six-byte difference is for additional header information about the packet going through the NDIS interface.


The MAXBUFFSIZE in PROTOCOL.INI, along with Buffers 14 4202 in NET.CFG, ensures that the workstation can utilize a 4 KB packet when using a token-ring board running at a data-transfer rate of 16 Mbps. Also, make sure that your token-ring board is set up with a RAM segment size of 16 KB or higher.


Figure 10. ODI2NDI Configuration Files (continued from page 65)

NEW SERVICE—PERSONAL SYSTEMS REPRINTS!

Toot thine own horn; be true.

 Sometimes, you need to toot your own horn. Yet to be truly effective, you must be true—that is, be able to back up your words. So—how to convince clients, competitors,

 managers, (your friends, even) just how great you *really* are? Simple... just order a set of *Personal Systems* reprints. For less than 50 cents per copy, you can get 1,000 black-and-white reprints of

 an eight-page article! When people can actually see your name or product in print, they'll be convinced. For price quotes, call Lia Wilson at (817) 961-6267.

Unit Costs for Reprinting 8-Page Article (1,000 Copies)

Black-and-white less than 50 cents

Two-color just over 70 cents

Four-color a little over a dollar

Questions and Answers

I am attempting to delete a program object using REXX SysDestroyObject. I understand that this command needs an object ID, but I'm not sure how to determine what that ID is.

My other option is to delete a program using the object name; however, this process deletes the program without deleting the folder from the desktop. I have a hunch that what I'm trying to do cannot be done for objects with no object ID. Am I right?

You are basically correct. You can use the object ID or the path name to delete an object via SysCreateObject; however, not all objects in the Workplace Shell (WPS) have object IDs, nor are they required for the WPS to function.

An easy way to find an object's ID is to use the Deskman/2 utility to display the object's settings. However, if the object was created from the OS/2 templates or by the drag-and-drop method, it will not have an object ID.

You can destroy certain objects by using the filesystem path. The example in Figure 1 will delete a folder named a_fold from the desktop.

Data files can also be deleted from the desktop. The example in Figure 2 will delete a file called data.

To delete a program reference object from the desktop using the filesystem path, you might try to delete the actual executable file as in Figure 3.

This deletes the executable file from the hard disk but does not delete the program reference object from the desktop. You can't destroy a program reference object if you don't have its object ID. REXX doesn't have anything to "grab onto."

Any object appearing on the desktop that you can also see via an OS/2 command

```
/* REXX */
call RxFuncAdd SysLoadFuncs, REXXUtil, SysLoadFuncs
call SysLoadFuncs

rc = SysDestroyObject("d:\desktop\a_fold");
if rc = 1 then
    say 'object destroyed successfully'
else
    say 'unable to destroy object'

/* end of program */
```

Figure 1. REXX Procedure to Delete a Folder

```
/* REXX */
call RxFuncAdd SysLoadFuncs, REXXUtil, SysLoadFuncs
call SysLoadFuncs

rc = SysDestroyObject("d:\desktop\data");
if rc = 1 then
    say 'object destroyed successfully'
else
    say 'unable to destroy object'

/* end of program */
```

Figure 2. REXX Procedure to Delete a File

```
/* REXX */
call RxFuncAdd SysLoadFuncs, REXXUtil, SysLoadFuncs
call SysLoadFuncs

rc = SysDestroyObject("d:\os2\program.exe");
if rc = 1 then
    say 'object destroyed successfully'
else
    say 'unable to destroy object'

/* end of program */
```

Figure 3. REXX Procedure to Delete a Program Reference Object

prompt (i.e., it is in the desktop subdirectory) can be deleted from the desktop with SysDestroyObject. Objects appearing on the desktop that do not appear in the desktop subdirectory (e.g., shadows,

program references, abstract objects) require an object ID for deletion.

Note: You might have more than one \desktop subdirectory. These will be


```

/* REXX */

call RxFuncAdd SysLoadFuncs, REXXUtil, SysLoadFuncs
call SysLoadFuncs

objID = '<WP_OS2SYS>'
hide='NOTVISIBLE=NO;'
call SysSetObjectData objID, hide
objID = 'c:\desktop\cm!2'
call SysSetObjectData objID, hide
objID = 'c:\desktop\upm'
call SysSetObjectData objID, hide
objID = 'c:\desktop\ffst!2'
call SysSetObjectData objID, hide

```

Figure 4. REXX Procedure to Hide Icons

REXX source:

```

/* var is a REXX variable assigned a value of 'Help' */
var = 'Help'
/* CALL command is used to call a CL program with a */
/* parameter */
'CALL PGM(CLPGM) PARM('var')'
/* DSPDTAARA command is used to verify that the CL */
/* program worked */
'DSPDTAARA DTAARA(*LDA)'

```

CL source:

```

/* THE CL PROGRAM ACCEPTS ONE PARAMETER */
PGM PARM(&INPUT)
/* THE SINGLE PARAMETER IS A TEN CHARACTER VARIABLE */
DCL VAR(&INPUT) TYPE(*CHAR) LEN(10)
/* CHGDTAARA IS USED TO LOAD LOCAL DATA AREA WITH THE */
/* PASSED VALUE */
CHGDTAARA DTAARA(*LDA (1 10)) VALUE(&INPUT)
/* THE CL PROGRAM RETURNS TO THE REXX PROGRAM */
ENDPGM

```

Figure 5. Sample REXX Procedure and Sample CL Program

named \desktop1, \desktop2, etc. To find the active desktop subdirectory, open any Settings folder and examine the path, which can be found on the File page.

I would like to run a REXX exec from bootable OS/2 2.1 diskettes. What files do I have to copy to Diskette 1 to run REXX?

You need the following files to run a REXX exec from bootable OS/2 2.1 diskettes:

```

\OS2\DLL\REXX.DLL
\OS2\RXQUEUE.EXE

```

```

\OS2\DLL\REXXAPI.DLL
\OS2\RXSUBCOM.EXE
\OS2\DLL\REXXINIT.DLL
\OS2\REX.MSG
\OS2\DLL\REXXUTIL.DLL
\OS2\REXH.MSG

```

These files take approximately .4 MB of disk space.

I have been able to hide icons on the desktop using the REXX SysSetObjectData command with a parameter of NOTVISIBLE = YES. I can't figure out how to hide Communications Manager/2, Main,

First Failure Support Technology/2 (FFST/2), User Profile Manager (UPM), and LAN Services. Can you help me?

Figure 4 contains the REXX code you need to hide icons. The system folder's ID is known. The other objects do not have IDs, so you need to specify the path instead of the object ID.

Note: You might have more than one \desktop subdirectory. These will be named \desktop1, \desktop2, etc. To find the active desktop subdirectory, open any Settings folder and examine the path, which can be found on the File page.

I would like to have a REXX procedure call a control language (CL) command. Is this possible, and, if so, how? I would also like to know how to submit a REXX job in batch mode.

CL commands are called from REXX procedures by simply specifying the command enclosed in single quotation marks. CL programs are called from REXX procedures by issuing the CALL command and passing the program's name in the program (PGM) parameter.

Figure 5 contains a sample REXX procedure and a sample CL program. The REXX procedure issues the CALL command to initiate the CL program.

Notice that when these programs run, the parameter passed to the CL program is converted to all upper case. This is natural for REXX programs calling CL programs and is documented in the *REXX Programmer's Guide* (SC24-5553) starting on page 7-13.

To submit a REXX procedure for batch processing, use the Submit Job (SBMJOB) command to submit the Start REXX Procedure (STRREXPRC) command. The command string looks like this:

```

SBMJOB CMD(STRREXPRC
SRCMBR(rexpgm)
SRCFILE(rexfile)
PARM(parms)) <This is all
one line>

```

See the *Control Language Reference* (SC41-0030) for details about the SBMJOB and STRREXPRC commands.

Corrective Service Information

Figure 1 shows maintenance release levels for the listed products. This information is effective as of July 25, 1994. CSDs may have been updated since press time.

To order all service packages—except for the OS/2 2.0, OS/2 2.1, OS/2 2.1 for Windows, and OS/2 2.0 Toolkit ServicePaks—call IBM Software Solution Services at (800) 992-4777. For the OS/2 2.0 ServicePak (XR06100), OS/2 2.1 ServicePak (XR06200), OS/2 2.1 for Windows ServicePak (XR06300), or the

IBM Developer's Toolkit for OS/2 2.0 ServicePak (XR06110) on diskettes or CD-ROM, call (800) 494-3044. Most OS/2 service packages are also available electronically from the following sources:

- **OS/2 Bulletin Board Service (BBS):** Once connected, select Option 2. (Corrective services are also listed under the General category on the IBMLink BBS.) To subscribe to the OS/2 BBS, call (800) 547-1283.
- **IBM Personal Computer Company (PCC) BBS:** Call (919) 517-0001.

Service packages are located in Directory 4.

- **CompuServe:** Download service packages from the IBM OS2 FORUM library (GO IBMSERV).
- **Internet:** Do an anonymous FTP from software.watson.ibm.com. Most packages are located in the \PUB\OS2 directory. TCP/IP packages are located in the \PUB\TCPIP\OS2 directory.

—Arnie Johnson, IBM Corporation, Austin, Texas

Product/Component	Release	CSD Level	PTF Number	Change Date	Comments
OS/2 Standard Edition	1.3	XR05150	XR05150	2-10-93	
OS/2 Extended Edition	1.3	WR05200	WR05200	5-12-93	WR05200 replaces WR05050, which can no longer be ordered on diskette.
OS/2	2.0	XR06100	XR06100	9-1-93	XR06100 replaces XR06055.
OS/2 2.10 ServicePak	2.1	XR06200	XR06200	3-1-94	This package is not for OS/2 2.1 for Windows.
OS/2 2.11 for Windows ServicePak	2.11	XR06300	XR06300	5-24-94	
OS/2 Toolkit	2.0	XR06110	XR06110	9-1-93	
	1.3	XR05053	XR05053	3-23-92	
OS/2 LAN Server/Requester ServicePak	2.0	IP06030	IP06030	4-25-93	
OS/2 LAN Server/Requester ServicePak	3.0	IP07045	IP07045	4-28-94	Includes IP07001, IP07003 (DLR), and I07005. This package has a co-requisite for WR07045 NTS/2 ServicePak, which ships with it. If you already have a refresh level of 3.00.1, you don't need this package.
OS/2 Extended Services Database Manager ServicePak	1.0	WR06035	WR06035	11-18-93	Supersedes WR06001, WR06002, WR06003, WR06004, WR06014, and WR06015.
Database Manager DB2/2	1.0	WR07015	WR07015	1-19-94	Supersedes WR07010 and WR07012. Download from one of the BBSs.
DDCS/2	2.0	WR07016	WR07016	1-19-94	
DBM DB2/2 Query Manager SelectPak	1.00	WR07022	WR07027	7-6-94	
DBM DB2/2 V1 PC DOS REQ SelectPak	1.00	WR07023	WR07023	5-6-94	
DBM DDCS/2 V2 SelectPak	1.00	WR07024	WR07024	5-6-94	
DBM DB2/2 V1 and DBAT SelectPak	1.00	WR07025	WR07026	7-6-94	
Extended Services Comm Mgr ServicePak	1.0	WR06025	WR06025	11-29-93	
System Performance Monitor (SPM/2) ServicePak	2.0	WR06075	WR06075	12/10/93	

Figure 1. Maintenance Release Levels

Product/Component	Release	CSD Level	PTF Number	Change Date	Comments
OS/2 Network Transport Services/2 SelectPak	2.00	WR07045	WR07045	4-27-94	
OS/2 LAN Adapter and Protocol Support SelectPak	2.20.2	WR07045	WR07045	4-27-94	
Communications Manager/2 Version 1.01 ServicePak	1.01	WR06050	WR06050	6-11-93	Available only on diskette.
CM/2 Version 1.11 ServicePak	1.11	WR06150	WR06150	5-31-94	
DOS	4.0, 4.01	UR35284	UR35284	9-26-91	
	5.0	UR37387	UR37387	9-22-92	
C Set/2 Compiler	1.0	CS00050	XR06150	6-29-93	
C Set C++ Compiler	2.0/2.01	CTC0002	XR06102	12-15-93	
C Set C++ Compiler	2.0/2.1	CTC0009	XR06180	5-9-94	
TCP/IP for OS/2 Base and Application Kit	2.0	UN56401	UN56401	6-21-94	
TCP/IP for OS/2 Applications	2.0	UN52840	UN52840	12-31-93	
TCP/IP for OS/2 DOS Box Kit	2.0	UN50383	UN50383	12-31-93	
TCP/IP for OS/2 Extended Networking	2.0	UN60005	UN60005	6-21-94	
TCP/IP for OS/2 Programmer's Toolkit	2.0	UN57887	UN57887	6-21-94	
TCP/IP for OS/2 Domain Name Server	2.0	UN54143	UN54143	12-31-93	
TCP/IP for OS/2 Network File System	2.0	UN57064	UN57064	6-21-94	
TCP/IP for OS/2 X-Windows Server	2.0	UN52841	UN52841	12-31-93	
TCP/IP for OS/2 X-Windows Client	2.0	UN52842	UN52842	12-31-93	

Figure 1. Maintenance Release Levels (Continued)

TRADEMARKS

Personal Systems has made every effort to supply accurate trademark information about company names, products, and services mentioned in this magazine. Trademarks not known that may have been inadvertently used in this magazine are the property of their respective owners.

Advanced Peer-to-Peer Networking, AIX, APPN, AS/400, CICS, Common User Access, Communications Manager, CUA, CUA '91, DATABASE 2, Database Manager, DB2/2, DB2/400, DB2/6000, DDCCS/2, Distributed Database Connection Services/2, Distributed Relational Database Architecture, DRDA, Extended Services, FFST/2, First Failure Support Technology/2, IBM, IBMLink, LAN Distance, NetView, OS/2, OS/400, Presentation Manager, Personal Systems/2,

PS/2, SAA, ServicePak, SNA, Ultimea, WIN-OS/2, and Workplace Shell; International Business Machines Corporation.

ANSI; American National Standards Institute
 Apple, AppleTalk, and Macintosh; Apple Computer, Inc.
 C++; AT&T Inc.
 CASE; CASEWORKS, Inc.
 CompuServe; CompuServe, Inc.
 CP/M; Digital Research, Inc.
 CS and Edit/2; Multitask Consulting
 DEC; Digital Equipment Corporation
 DeScribe; DeScribe Inc.
 DeskMan/2; Development Technologies
 Ethernet; Xerox Corporation
 GpfRexx; Gpf Systems
 Hewlett Packard and HP; Hewlett Packard Co.
 Hockware, VisPro/C, VisPro/C++, and VisPro/REXX; HockWare, Inc.

IEEE; Institute of Electrical and Electronics Engineers
 Intel, 80386, and 80486; Intel Corporation
 Internet; Internet, Inc.
 KnowledgeWare; KnowledgeWare, Inc.
 Lotus, Lotus 1-2-3, Ami Pro, and Notes; Lotus Development Corp.
 NetWare, NetWare Loadable Module, NLM, CNE, ODI, Open Datalink Interface, Novell, and UNIX; Novell, Inc.
 Smalltalk; Digitalk, Inc.
 Sytos; Sytron.
 WATCOM and VX-REXX; WATCOM International Corporation
 Windows, Windows NT, Windows for Workgroups, MS-DOS, Word, LAN Manager, MS-NET, and Visual Basic; Microsoft Corporation
 WordPerfect; WordPerfect Corporation
 X/Open; X/Open Co., Ltd.
 3Com; 3Com Corporation

IBM Personal Systems

These back issues of *Personal Systems* are available to provide valuable information. Indicate the desired quantity for the issues you want to order and complete the information on the following page.

July/August 1994

IBM's Personal Systems Support Family—Customer-Influenced Design
 OS/2 Times and Scores the 1994 Indianapolis 500
 Software Compatibility: Good Relationship or One Night Stand?
 Migrating Windows Applications to OS/2: Easing the Migration Path
 OS/2 Conference Draws Praise
 DCE: An Application Primer
 Distributed Performance Characteristics of IBM DCE for OS/2
 Architecture Soup: Understanding Modern IBM PC Architecture
 TSHELL: A Text-Based Alternate Shell for OS/2
 Extended Attributes for Files
 Developing Lotus Notes Applications
 Conserving Power with Personal System Power Management
 Superstor/DS Data Compression in PC DOS 6.x
 LAN NetView Object Registration Services

May/June 1994

"Wrightsizing" at USAir
 Getting the Word Out at Chemical Banking Corporation
 Back Up for the Future
 Lost in Cyberspace
 The Book Shelf
 Threads
 Redirected Installation of OS/2 2.x
 LAN Server Ultimedia 1.0 Performance and Tuning

March/April 1994

If I Only Had a Brain
 Speech Recognition Products Untie Your Hands
 Telecommuting in the '90s
 Point of View: Not Just Another Database Article
 Professional Certification Program from IBM
 Celebrate the Past on Your Trip Back to the Future!
 OS/2 2.1 Performance Tuning Tips—Part II
 PC File Systems
 NetWare 4.01 for OS/2: Features and Installation
 NetWare Requester for OS/2, V2.01: Features and Installation
 What's New in Novell NetWare 3.12?
 LAN Analysis Using IBM's DataLANce Network Analyzer
 NetWare Questions and Answers

January/February 1994

Plan, Plan, Plan Your NetWare 4.01 Network
 LAD/2 in the LCU and NetView DM/2 Environments
 Easy Setup of CID Code Servers
 Managing Token-Ring Bridges with IBM's LAN Network Manager
 IBM DCE for OS/2 Multiuser Application Performance
 Performance of Key Functions in DCE for OS/2
 VisualAge: Its Features and Virtues

November/December 1993

IBM PC-DOS 6.1: More Features than MS-DOS 6.0
 SystemView Information Warehouse DataHub
 Developing DataHub Tools
 Using MMPM/2 to Create Multimedia Applications
 Advanced Client/Server Computing Using the IBM ThinkPad
 Communications Manager/2: A New Look
 Overview of IBM NetWare 4.01
 OS/2 2.1 Performance Tuning Tips

September/October 1993

IBM PSP's LAN Systems Solutions
 An Introduction to PCMCIA
 PCMCIA Software: The Key to Compatibility
 OS/2 Support for PCMCIA Memory Cards
 Improving Remote Initial Program Load Performance
 Installing and Configuring CM/2 1.0
 Writing CID-Enabled Applications
 Integrating LAD/2, CM/2, and DB2/2 with IBM LAN NetView Start
 LAN NetView Start
 DB2/2—DB2 Comes to the Desktop

July/August 1993

OS/2 2.1—Everything You Wanted It to Be and More
 Using REXX to Customize the Workplace Shell—Part II
 Client/Server Application Development with OS/2 and CICS/ESA
 Upgrading to OS/2 LAN Server 3.0—Advanced
 Developing OS/2 LAN Server Services
 PCMCIA PC Cards Provide Expandability and Network Interfacing
 Using the IBM ThinkPad with OS/2 and CM/2

April 1993

XGA-2: Improving on a Good Thing
 IBM Personal Software Products: Product Line Update
 Using REXX to Customize the Workplace Shell
 OS/2 Distributed Systems Management with LAN NetView
 Priming and Querying Your Start Network
 Multimedia Applications on IBM Token-Ring LANs
 OS/2 2.0 Print Tips
 Testing OS/2 PM Applications
 Accessing a Remote AS/400 Using OS/2 Extended Services
 Virus Information and Protection
 Migrating from APPC/PC to Networking Services/DOS
 OS/2 2.0 Resources
 OS/2 32-Bit Application Migration Workshops
 IBM OS/2 Products Available on CD-ROM

January 1993

PS/2 Desktop Security
 IBM 486SLC2: System Performance Implications
 Micro Channel Developers Association
 Trackpoint II: The In-Keyboard Pointing Device
 Why OS/2 2.0?

OS/2 Distributed Systems Management
CID: Remote OS/2 Configuration, Installation, and
Distribution of PC Software
Start/2: Putting the Configuration into CID
LAN Server 3.0: New Thresholds in High-Performance Network
Software
The Future of IBM LAN Network Management
Understanding and Using the Workplace Shell
Distributed Processing: A Case Study
Parallel Port Protocols
Developing OS/2 PM Applications with Micro Focus COBOL
OS/2: How About Notebooks?
Loadable BIOS

October 1992

Exploring File Server Performance
PS/2 3.5-Inch Rewritable Optical Drive
Programming the XGA Video POS Registers
Video Monitoring on Personal Computers
Memory Address Space
OS/2 2.0 Installation and Performance Considerations
OS/2 2.0 Application Support
Cleaner Installation of Applications Under OS/2
Creating Resizable Pushbuttons
Configuring Parallel Ports for OS/2
Performance Characteristics of ES 1.0 Database Manager
AlertVIEW
Screen Reader/2

July 1992

IBM PS/2 Server 295: New Thresholds for Client/Server
Networking
Comparing Architectures: Micro Channel and EISA (Part 2)
Synergy by Design
Pen-Based Computers
Why Doesn't My Portable's Battery Last Longer?
Planning Guidelines for Token-Ring Cabling
Installing and Migrating Applications in OS/2 2.0
Printing Under OS/2 2.0
Installing the IBM 4029 LaserPrinter Under OS/2 1.3

Serviceability Tools in OS/2 2.0
Online Communication Using the OS/2 2.0 PM Terminal
IBM Extended Services Database Manager
NetWare for SAA
Using the IBM DOS 5.0 Driver EMM386.EXE and
Upper Memory
The Solutions Evaluation Tool

April 1992

Comparing Architectures: Micro Channel and EISA
Portable Computer Trends and Directions
LCD Panel Technology
The OS/2 Workplace Shell
New Applications in OS/2 2.0
Unattended Installation of OS/2 2.0
OS/2 Communications Manager Trace Events
IBM and Novell LAN Software Coexistence
IBM 8209 LAN Bridge Connects Ethernet Clients to Novell
and IBM Servers
Backup and Restore in an IBM NetWare Environment
The DOS Protected-Mode Environment
DOS Disk Management
Customizing Alphanumeric Screen Dimensions

January 1992

Additions to the IBM PS/1 Family
IBM LaserPrinter 4029 Series Print Quality Enhancements
OS/2 2.0: The Integrating Platform
Multiple Virtual DOS Machines
IBM OS/2 LAN Server 2.0
OS/2 2.0 Memory Management
Coding for Performance Under OS/2 Version 2.0
Extending the Functions of OS/2 REXX
Protecting User Exits Under OS/2 1.x
GDDM-OS/2 Link
IBM Upgrade Enhanced Install Utility/DOS 5.0
Advanced Peer-to-Peer Networking: An Overview
Using IBM SAA Networking Services/2
The AAI Family of Products
Securing the Enterprise Workstation

BACK ISSUE ORDER FORM

NAME _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE (_____) _____

Price is \$12.00 per issue, plus \$3.95 shipping & handling per copy. Overseas orders add \$9.95 shipping & handling per copy.

Texas residents add applicable sales tax.

I have enclosed a: Check Money order
Charge to: VISA MasterCard AmEx Diners

CREDIT CARD NUMBER _____

SIGNATURE _____ EXPIRES _____

Send this form with a check or money order, payable to **NCM Enterprise**, to: NCM Enterprise, P.O. Box 165447, Irving, TX 75016-9939. You can also fax both pages of this form to **(214) 518-2507** (please include VISA/MasterCard/AmEx/Diners number and expiration date), or call **(800) 678-8014**. All orders must be prepaid. Checks must be in U.S. dollars.

Maybe You Can't Be In Six Places At Once...

*Relish®
Version 2.2 Spreads
Easier than Ever*

But now your

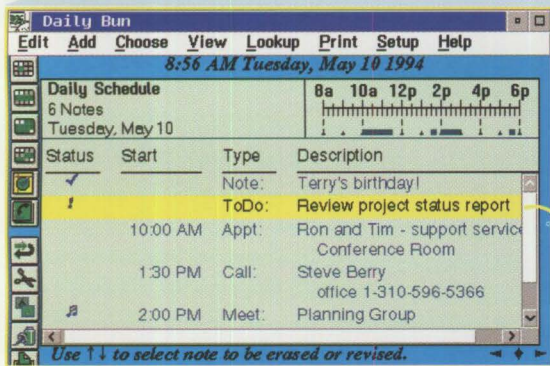
- Calendar,
- Phone Book,
- To Do List,

and other personal information can



Amazing Relish 2.2 Hot Buns

They're not files, not folders, not programs. They're a new kind of Workplace Shell object for dynamically viewing Relish information.



Personal information where you need it, when you need it, always up-to-date, always consistent — a sure way to Relish every moment of your day!

Only Relish 2.2 Gives You All This

- * NEW! Iconbar for one-click access to your preferred functions
- * NEW! Type-to-search lets you "just start typing" to find things fast
- * NEW! Floating entries (without times or dates) appear on current date
- * Entries can span any length of time from a few minutes to many days
- * Intelligent time and date recognition gives you quick keyboard entry
- * Alarms can be scheduled as far ahead of an event as you want
- * Quick-access reference calendar can remain on the desktop
- * Reminders are automatic; so is schedule synchronization on a LAN
- * The most extensive drag-and-drop support of any OS/2 program

Here's how they work... Buns give you views of your information when and where you need it. Use them just as you would Relish - or customize them to match your needs.

Aesthetically, change colors, fonts, window sizes and positions to fit your taste. And, there's a whole lot more.

Every bun can be copied and modified to present different collections of information.

Think of the possibilities. One bun can show what To Do items remain for that important project. Another can have your meetings with Fred. Others for each category within your phone book. Put them in folders, on your desktop, anywhere you need them.

When you want just those project-related To Dos, open that bun and there they are. Need a printed copy? Just drag your bun right to a printer without even opening it.

Best of all, everything is always up to date. If you make a change through your complete To Do list, or put another project To Do on your calendar, your customized bun is automatically and instantly updated, right before your eyes!



Relish

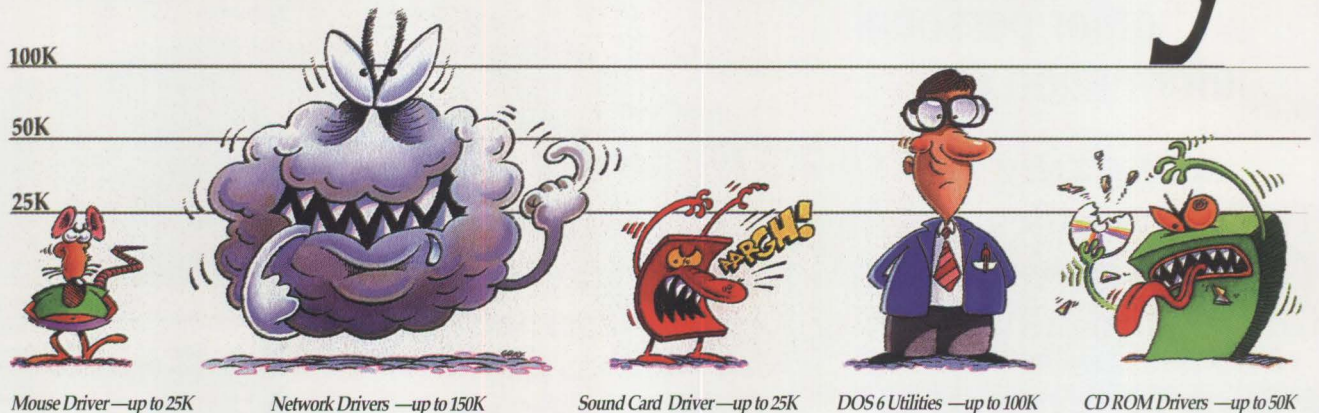
NEW Version 2.2 — one part of Sundial System's "Personal Information Object" strategy for giving you access to information like you've never had it before.

Call 310•596•5121

Sundial Systems Corporation

909 Electric Ave, Suite 204,
Seal Beach, CA 90740 USA

What's Eating Your Memory?



Mouse Driver—up to 25K

Network Drivers—up to 150K

Sound Card Driver—up to 25K

DOS 6 Utilities—up to 100K

CD ROM Drivers—up to 50K

Memory munchers lurk in your system.

They consume memory your DOS programs need to work smoothly, or even to load at all—whether they run in Windows or from the DOS prompt.

When You Lose Memory, You Lose Power

When you install new hardware or software, haven't you wondered where all your memory goes? Or why things start to slow down?

Beep! Not enough memory to run. Beep! General Protection Fault! Beep! Crash. Things are getting a little unpredictable. What's happening to your computer? What can you do?

If You Need It, You Have to Feed It

Software drivers all take a bite out of your 0-640K memory area; and if they eat too much, your applications will grow sluggish or even refuse to run at all.

But drivers are necessary for all the things you want to use: most programs talk to a mouse driver, not to the mouse itself; a CD-ROM drive needs one so that DOS can recognize it. A sound card usually needs a driver for applications to talk to, and so on.

All in all, the more you want from your PC, the more mouths you'll need to feed.

Get it All Back—and More

QEMM 7 delivers as much conventional (below 640K) memory area as possible by relocating these hungry drivers into vacant memory space above 640K. That frees up the area in conventional memory that drivers were stealing. Memory needed by games, data bases, and other programs. You could find yourself with a bonus 250K that you never knew you had!

QEMM 7—the Safe, Fool-proof Solution

Nothing could be simpler than installing QEMM. It automatically calculates millions of memory configurations in minutes to make sure that your PC's memory is at its optimum. And that ensures your software will run its best.

QEMM 7 employs its patented Stealth™ and DOS-UP™ technology to give 8K-24K below 640K for best Windows performance, load the DOS SHARE program used by Windows OLE, install a mouse, sound card, CD-ROM and a network... and still have more than 630K! No wonder QEMM outsells all the other memory managers put together!

See your software dealer today to find out more about how QEMM protects your memory and your productivity.

Quarterdeck

Quarterdeck Office Systems, 150 Pico Boulevard, Santa Monica, CA 90405 (310) 392-9851 Fax (310) 314-4219
Quarterdeck International Ltd, B.I.M. House, Crofton Terrace, Dun Laoghaire Co. Dublin, Ireland Tel.(353) (1) 284-1444 Fax: (353) (1) 284-4380

©1994 Quarterdeck Office Systems. Trademarks are property of their respective owners.

Please circle #13 on reader service card.