

# Personal Systems

IBM's MAGAZINE FOR TECHNICAL COORDINATORS

JULY/AUGUST 1994

## Architecture Soup: A Choice Recipe

**DCE Performs**

**OS/2 in Under 3 MB**

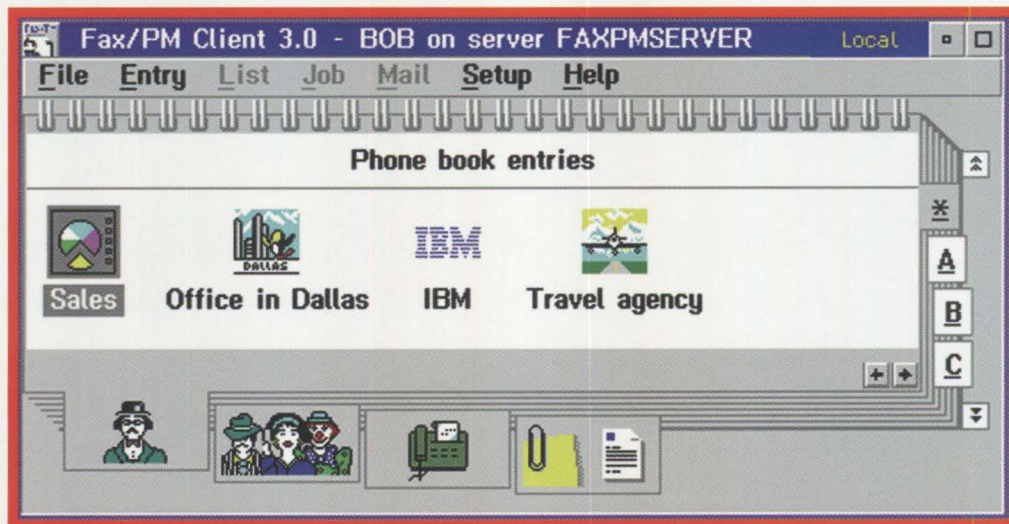
**Developing with Lotus Notes**

**OS/2 Races at Indy 500**

**IBM**

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

# Can a DSOM Client Server product



## solve Fax and Email problems?

1. Yes, when you need to fax from Dos, Windows, and OS/2 applications,



2. Yes, when you want to email simple messages or complete objects.



3. Yes, when you want a common user interface for multiple platforms (OS/2, Windows, AI/X...).

5. Yes, when you look for power and security.

4. Yes, when you want to be LAN independent (Novell, LAN Server, NetBios, TCP/IP).

6. Yes, when you need migration to DB2/2 and OpenDoc.

# Fax/PM

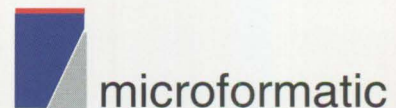
## ClientServer

Distributed  
SOM



**Fax/PM ClientServer.**

To order or find out more about Fax/PM:  
USA, call 1-203-644-1708, fax 1-203-648-9587.  
Europe, call 33-1-48701900, fax 33-1-48702729.



Fax/PM is a trademark of Microformatic. All other referenced products are trademarks or registered trademarks of their respective manufacturers.

Please circle #1 on reader service card.

**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: (310) 550-8575  
 Fax: (310) 550-1152  
 BBS: (310) 550-5980

IBMLink: DEV2203  
 CompuServe: 76426,363  
 OS2BBS1: LANUTIL

# Information: You Asked for It!



This is our biggest, most information-packed issue ever! Our IBM authors, consultant authors, and product developer authors have limbered their minds and their fingers to bring you up to date on topics from architecture to text-based alternatives, from conserving power to developing power, from fact to opinion.

This issue includes another article in our series on Distributed Computing Environment (DCE) performance. Some of you told us you would like to know a little more about the basics of DCE. So, after we found out that Phil Lieberman of Lieberman and Associates is an expert on DCE, we asked him to write a primer—describe what DCE is and how it works.

Our cover story on Architecture Soup gives you an in-depth view of all of your options for hardware architecture. Chet Heath uses his vast, and frequently rewarded, knowledge of personal computer architecture to describe the mysterious acronyms floating in this “alphabet” soup: ISA, PCI, EISA, VESA, PCMCIA. Where does it all end? You’ll understand where they all fit and what they can do for you after reading Chet’s very understandable article.

Still hungry for more information? Don’t miss Executive Computer Systems’ chief consultant Bill O’Connor’s article on making OS/2 work on your less-than-loaded-with-memory computer. Bill describes how this free software, available for just a phone call to your favorite bulletin board system, lets you install and use the powerful 32-bit multitasking capabilities of OS/2 with less than 4 MB of RAM!

## Keep Telling Us!

The only way we know what you need is for you to tell us. These articles are in response to your requests through a variety of vehicles. For instance, we just surveyed many of you. To those who responded, we thank you! You made many suggestions for content, and you’ll see the results of those requests in future issues of *Personal Systems*.

Every issue of *Personal Systems* has an editorial evaluation card—use it to tell us how useful these articles are to you and what you would like to see in subsequent issues.

All you bulletin board users out there can get directly to us through Internet. Just put your comments and suggestions in a note to [psts@vnet.ibm.com](mailto:psts@vnet.ibm.com).

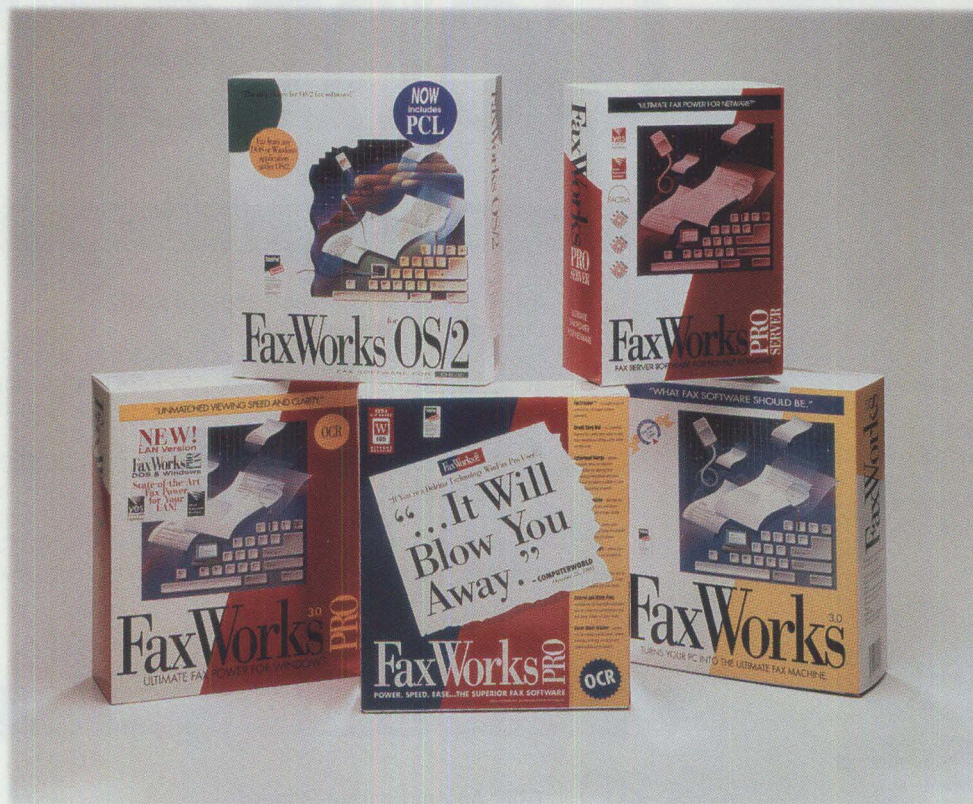
And don’t forget the valuable product information available from *Personal Systems*’ advertisers. On the product information card in this magazine, circle the numbers of the products in which you are interested. The information is free, and you don’t even need any postage on the card! Mail yours today!

## September/October Issue

You asked for it, you’ll get it—in *Personal Systems*’ September/October issue. The topic for which we get the most requests is REXX. We’ve gone out and beat the bushes for pertinent, up-to-date REXX articles—from IBM product specialists, industry consultants, and independent software developers. Don’t miss it!

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**

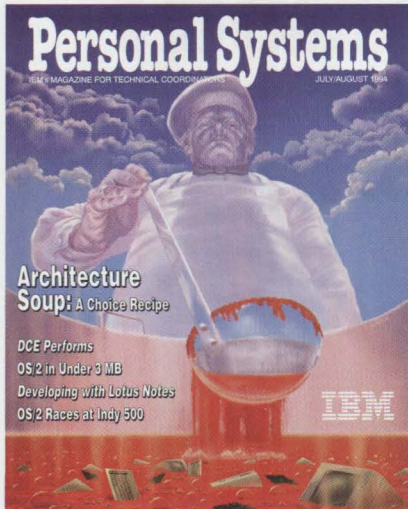
**SOFNET**

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

IBM's Micro Channel architect, Chet Heath, characterizes as "alphabet soup" the wide and robust choices available to you today for personal computer architectures.

**Editor and Publisher**  
Betty Hawkins

**Business Manager/  
Editorial Coordinator**  
Donna Su

**Editorial Assistant**  
Lia Wilson

**Database Manager**  
Van Landrum

**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

- 6 IBM's Personal Systems Support Family—  
Customer-Influenced Design**  
Announced in May, IBM's new Personal Systems Support Family provides a menu of support options from which you can tailor a support program that meets your organization's specific needs. This article details these choices.
- 10 OS/2 Times and Scores the 1994 Indianapolis 500**  
Another Indianapolis 500 race has been successfully timed and scored using IBM's OS/2 operating system. Patrick Karle of Patrick Karle Associates describes OS/2's highly accurate and speedy score reporting and elaborates on IBM's long partnership with the Indianapolis Motor Speedway.

## TECHNICAL

- 20 DCE: An Application Primer**  
Philip Lieberman of Lieberman & Associates presents a tutorial on the basics of the Distributed Computing Environment (DCE) and how developers use it to build client/server systems.
- 27 Distributed Performance Characteristics of IBM DCE for OS/2**  
This article presents a study of IBM DCE for OS/2's performance characteristics using hardware and software options that distribute the application workload within a heterogeneous, single-cell environment.
- 36 Architecture Soup: Understanding Modern IBM PC Architecture**  
The computer user and computer designer now have hundreds of standards from which to select the optimal recipe. This article describes these choices, focusing on local buses, I/O buses, file interfaces, and parity in RAM.
- 48 TSHELL: A Text-Based Alternate Shell for OS/2**  
Bill O'Connor, Executive Computer Systems' chief consultant, describes TSHELL, which allows OS/2 to run on computer systems with low memory and hard-disk resources. He describes the philosophy behind TSHELL, discusses system requirements, and provides case studies of customers for whom TSHELL is the appropriate user interface.
- 57 Extended Attributes for Files**  
This article discusses hints and tips for using extended attributes in programming. It also provides a sample code program to view attributes in an ASCII file.

## LITTLE SOLUTIONS

- 90 OS/2 Tips & Techniques**  
A variety of OS/2 tips and techniques are provided in this issue.



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](mailto: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.

#### **Personal Systems Advertising Contacts**

*Personal Systems* accepts a limited amount of 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

## **13 Software Compatibility: Good Relationship or One Night Stand?**

Bob Angell, a principal with Applied Information and Management Systems, discusses the evaluation criteria his firm used when first installing OS/2. He also discusses benchmarking applications running under OS/2.

## **16 Migrating Windows Applications to OS/2: Easing the Migration Path**

With the rapidly increasing OS/2 installed base, tools are needed to assist in migrating current applications smoothly to OS/2. One Up Corporation's Jeff English discusses the major issues of porting applications across platforms and reviews their SMART product.

## **19 OS/2 Conference Draws Praise**

This article reviews the successful spring OS/2 Conference held in San Francisco.

## **63 Developing Lotus Notes Applications**

Lotus Notes is a tool allowing users to more effectively share, distribute, and collect information across a computer network. The secrets for developing your own programs in a Lotus Notes 3.0 environment are shared in this article.

## **68 Conserving Power with Personal System Power Management**

The new generation of portable and desktop systems uses power management techniques to conserve power. This article gives an overview of power management and power management support implementation.

## **73 Superstor/DS Data Compression in PC DOS 6.x**

The major new feature in IBM PC DOS 6.x is a real-time disk compression capability called Superstor/DS. This article gives an overview of data compression, followed by a detailed description of Superstor/DS.

## **79 LAN NetView Object Registration Services**

The LAN NetView family of products uses Object Registration Services (ORS) to allow managing systems to know which objects are installed on managed workstations. This article provides a closer look at the ORS database and registration process.

## **92 Corrective Service Information**

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

# IBM's Personal Systems Support Family— Customer-Influenced Design

As IBM began designing its new support structure, customers were asked what they expected and what they wanted in technical support. In response, IBM announced the addition of Personal Systems (PS) Support Family to its Family of Support.

## IBM Director Says, "Satisfaction Guaranteed"

"We make it... easy to customize and easy to use... We will meet all the Personal Systems platform needs of our customers—satisfaction guaranteed," states Dell Rieth, IBM's Director of End User and Software Solutions Services and Support.

## Support Family is Flexible

You can "mix and match" PS Support Family offerings to best meet your organization's business needs. Figure 1 lists the support elements and their benefits to you.

## Support Family Is Easy to Use

Just dial (800) 799-7765 and an IBM support specialist will work with you to create a customized contract that spans these IBM platforms: System/390, Networking, AS/400, AIX/6000, and now, Personal Systems. You will gain access to a responsive team of support specialists (phone, electronic, and/or on-site, depending on the services you choose.)

## What's Free versus Fee?

The first question most of you are asking is, "What do I not have to pay for?" One of Support Family's major design points is to enable you to be self-supportive should you so choose. To achieve this, we ensured not only that you could purchase as much or as little support as desired, but also that you have access to

some support at no charge through the INFORMATION FREEWAY.

To access the on-ramp, call (800) 992-4777. The following features are available:

- "Most frequently asked" Personal Systems Q&As preloaded and updated frequently
  - Fax-back Q&A database
  - Pre-sales and marketing information
- Also available at no charge are:
- Technical Q&A databases accessible via TALKLink (requires that you pay \$18 per month for the TALKLink ID)
  - A 60-day "Getting Started" period

## Questions and Answers

**Q.** How will customers benefit from Personal Systems Support Family?

**A.** Customers benefit from the following features:

- Single point of entry—via an 800 number—into IBM's new technical support structure
- Optional single, integrated contract across all IBM platforms: System/390, AS/400, AIX/6000, and OS/2
- Choices from a comprehensive portfolio of services
- Voice or electronic access to a responsive team of support specialists
- Option of on-site assistance to supplement critical skills

**Q.** When can customers start purchasing Personal Systems Support Family services?

**A.** We began writing contracts May 10, 1994. We'll begin fulfilling services July 11, 1994.

**Q.** When product support overlaps Support Families (for example, between Personal Systems and Networking), how does the support differ?

**A.** With the Networking Support Family, IBM specialists will determine the problem and identify its source by isolating questions and problems to a specific product. If the question or problem relates to a supported product of the Personal Systems Support Family, then IBM will support that product on either a per-incident or annual basis through the Personal Systems Support Family.



# Learning OS/2

Get 25% off OS/2 training videos and OS/2 2.1 computer-based courses.

# just got

The most cost-effective way to build valuable OS/2 skills just got even more cost-effective. You can now purchase Skill Dynamics' *Working with OS/2 Version 2* video for only \$75 (regularly \$100). And our interactive computer-based *More for You in OS/2 2.1* for just \$59 (regularly \$79).

Individuals or even entire departments will find that our video and computer-based courses are also the most convenient way to bring vital skills up to date and take full advantage of the power of IBM's flourishing, award-winning OS/2. You can train at your own pace and at your convenience. Even at your own home. You can learn new skills and refresh yourself on old ones. It's also an inexpensive way to start an OS/2 training library. Perfect for helping new people get up to speed quickly and correctly. And because these materials were prepared with direct input from IBM, you know you're getting the inside story.

Start working better, faster and easier right away.  
To order, call 1 800 IBM-TEACH, ext. 1150.

**More for You in OS/2  
2.1 Computer-Based Course #PS284 \$59**

*(Designed for new OS/2 users, experienced OS/2 users and experienced Windows users. Completion time approximately 8 hours.)*

**Working with OS/2  
Version 2 Training Video  
# PS147 \$75**

*(Designed for new users of OS/2 Version 2.1. Completion time 4-6 hours.)*



# 25% easier.

See our Special Pass coupon in this publication for 10% off all LAN and OS/2 certification classes.

 Skill Dynamics™  
An IBM Company

OS2 and IBM are registered trademarks of International Business Machines, ©1994

Please circle #4 on reader service card.

| Support Element                                 | Benefits  |
|---|---|
| Support Line                                    | Phone and electronic access to technical specialists in areas such as OS/2, LAN Server, DB2/2, and more than 100 other products.  |
| Consult Line                                    | Conference calls with technical specialists who can assist with application design and planning, performance and tuning, implementation, plus LAN Systems design and systems management.    |
| House Call                                      | On-site assistance and/or specialized support at your location from IBM technical specialists.  |
| Forum   | A bulletin board service for customer-to-customer communication plus technical information for IBM TALKLink subscribers.  |
| Desktop Application Assistance                  | Technical assistance for over 250 industry application software products running under OS/2, DOS, and Windows.  |
| Customer Application Assistance                 | Assistance with problem determination for multiple products, customer applications, and integrated environments as well as complex debugging, application analysis, or design/code reviews. |
| Technical Education Coupons                     | In-depth classes on diagnostics, trace techniques, kernel debugging, and more from Skill Dynamics and others.   |
| "Technical Connection Personal Software" CD-ROM | An OS/2 on-line book collection, installation/tuning tips and techniques, a database of Q&As, plus a search tool.   |
| <i>Personal Systems</i> Magazine                | That's us! Need we say more?  |

Figure 1. Core Support Elements

- The ability to submit defects to IBM via fax or mail (IBM responds in kind) once your 60-day Getting Started period passes
  - Access to code fixes and the problem database defect resolution
  - Advanced technical support and consulting
  - Application code debugging
- Fee support includes:
- Installation and usage support for Personal Systems products

### Call 800 799-7765 for Details

For more detailed information, call Monday through Friday from 8:00 a.m. to 5:00 p.m., your time.

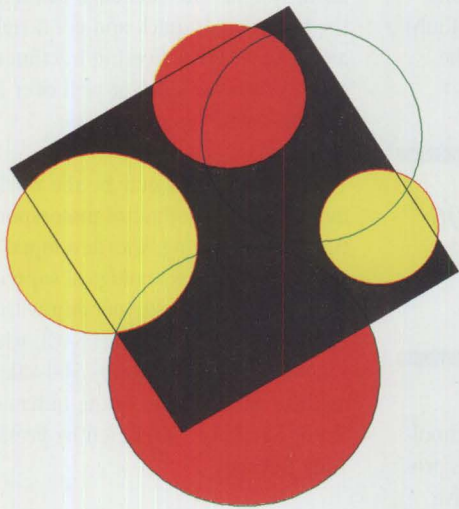


**Donna Su** is Editorial Coordinator and Business Manager for *Personal Systems*. She has worked on several publications since joining IBM in 1988, including

designing the Technical Coordinator Program newsletter. She received a BA degree in art from the University of Southwestern Louisiana in Lafayette and an MA degree in English from the University of North Texas in Denton.

Personal Systems Competency Center staff, Rose McAlister, Rene Gracia, Brent Allen, and Bryan Heft prepare for a Consult Line call.

**ALL New DCF/2**  
**HPFS Version Update**  
**Available NOW!**



# DCF/2

The only pure OS/2 on-the-fly  
data compression product  
**available today!**

## DCF/2 Utilizes:

- ✓ OS/2's High Performance File System (HPFS)
- ✓ All OS/2 DASD - FAT and Floppy Volumes
- ✓ Removable Media and Network Disks
- ✓ Password Protection and Volume Mirroring
- ✓ Existing Disks (no re-formatting)
- ✓ Dynamic Physical Disk Space Allocation
- ✓ OS/2's Presentation Manager

**\$149.95** *plus Shipping & Handling*

**Increase Your Disk Capacity. How Much?**  
**See For Yourself! Don't take our word for it ...**

**FREE!**



**Try Before You BUY!**

Call today for a **FREE** copy of our  
Disk Compression Analysis Tool (DCAT).

**Find out how much disk space you really could have ...**

Call 1-800-666-4672, (303) 484-2665 TODAY for your free copy of the DCAT or pick it up off of your favorite BBS, IBMLink (OS2DCF2), CompuServe (GO OS2AVEN/Proportional), or from Internet on the FTP.CDROM.COM.

**COMDEX Special Offer: "Two Great Products, One Incredible Deal" bundle of the DCF/2™ and DeskMan/2™ from DevTech - now until the end of November. DeskMan/2 is the ultimate desktop manager for OS/2 and includes the new VUEMan/2 virtual desktop manager. Anyone who has ever had to rebuild their desktop will know the value of the DeskMan/2!**  
**Call for details and come see us on the main floor IBM PSP Booth L860 Pedestal #31.**

Proportional Software, 1717 Linden Lake Road, Fort Collins, CO 80524, Tel. (303) 484-2665, FAX (303) 484-2670, CIS 71333,2765  
Development Technologies, Inc., 308 Springwood Road, Forest Acres, SC 29206-2113, Tel. (803) 790-9230

Please circle #5 on reader service card.

# OS/2 Times and Scores the 1994 Indianapolis 500

*In the September/October 1993 issue of Personal Systems, Patrick Karle wrote "Searching for Speed," documenting how IBM ThinkPads aid Indy drivers in their quest for safety and speed. In this article, Karle describes the highly accurate OS/2-based method of timing and scoring used by the Indianapolis Motor Speedway.*

The Indianapolis Motor Speedway coils like a 2.5-mile clock face at the crossroads of America. When the world's top drivers met there in May to match their skills against that great clock's sweeping second hand, the driver who beat the clock won big. Al Unser, Jr., winning his second Indianapolis 500,

took home 1.37 million from a record \$7,864,800 that was divided among this year's competitors, and it was the IBM OS/2 2.1-based IMS/USAC (Indianapolis Motor Speedway/United States Auto Club) Timing & Scoring System that told the Speedway how to pay the money.

---

*...scoring the Indy 500 requires extremely precise timing. . .*

---

Applying IBM timing and scoring technology, the world's richest auto race has progressed—under the management of five Directors of Timing and Scoring—from

punch cards to mainframe computers to IBM personal computers to an IBM local area network (LAN) with IBM's OS/2 operating system. The challenge has remained the same—get it quick and get it right in an arena where a fleeting fraction of a second can cost a driver well over half a million dollars.

The system, which may be the world's most accurate method of motorsports timing and scoring, was developed with equipment and technological support from IBM under a partnership with the United States Auto Club (USAC), which sanctions the Indianapolis 500-Mile Race, and the management of the Indianapolis Motor Speedway (IMS), led by President Tony George.



Used for the first time in 1990, the IMS/USAC Timing & Scoring System gives USAC a way to automatically score thirty-three 225-miles-per-hour (mph) race cars without having to click a stop watch, punch a time clock, or write numbers in a notebook.

## A Revolution in Timing and Scoring

Art Graham, USAC Director of Timing & Scoring, says the IMS/USAC system has revolutionized not only race day competition but also practice and qualification at the Speedway, and it has made a wealth of information available to race officials, competitors, spectators, and the media at speeds that rival today's Indy cars.

Graham says the system gives USAC officials the ability to track every race car from the moment it leaves the garage until it returns to the garage. There are now a total of 22 antennas buried in the track, pit areas, and new warm-up lanes. A car completing a lap on the track crosses 11 lines determining its speed in all turns and straightaways, and that's just the beginning of the information the system can supply.

He says scoring the Indy 500 requires extremely precise timing, because a close finish can make a big difference in how the money is paid. In 1988, Michael Andretti and Bobby Rahal finished fourth and fifth, separated by only 1/1,000th of a mile per hour. Andretti made \$41,300 more than Rahal. When Al Unser, Jr. won his first Indy 500 in 1992, he edged past Scott Goodyear by .043 of a second in the closest Indy 500 finish in history. He won \$1,244,184 to Goodyear's mere \$609,333.

"When I first came here 25 years ago, one lap took almost a minute and people were writing the car's times on IBM punch cards. Now, drivers turn laps in less than 40 seconds, and we have to supply more information to more people in less time than ever. To do it the old way would be like trying to write down the license plate numbers of cars as they speed by on the freeway," Graham declares.

## Accuracy Without Intervention

Graham says the process is basically very simple: the cars time themselves by

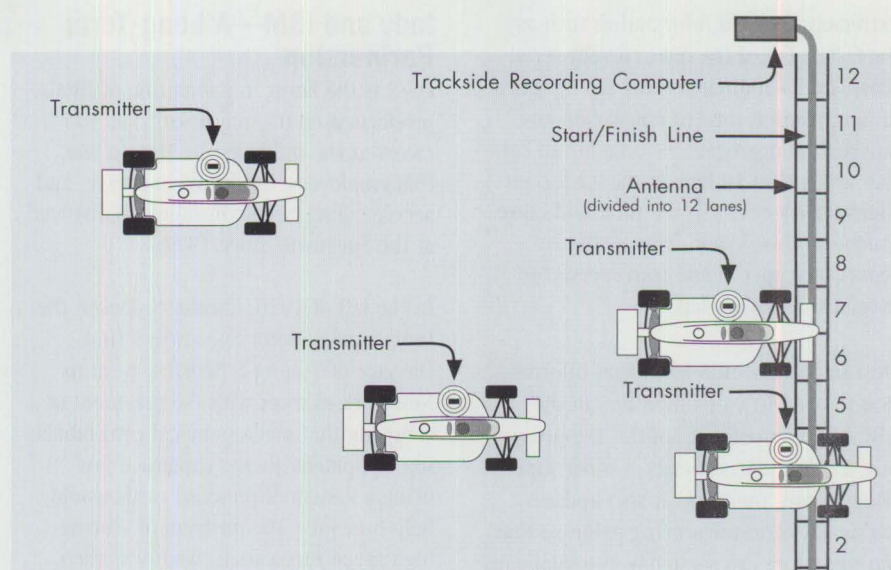


Figure 1. Transmitter and Antenna Placement

passing over the loop antennas buried in the track.

The core technology of the IMS/USAC system is the Dorian DATA-1 on-board digital radio-frequency (r-f) transmitters and a total of 22 antennas embedded in the track surface all around the race track. These antennas are connected to a PS/2 local area network and an OS/2 2.1-based data management system called Integrated Race Information Systems (IRIS)<sup>1</sup>, which refines and distributes all the data for the Indianapolis 500.

Each car carries a Dorian DATA-1 transmitter about the size of a cigarette pack fixed to the floor of its left side pod (Figure 1). As the car passes over the antennas in the track, the onboard transmitter electronically identifies itself by number. The antennas pick up the signal and feed it to a trackside recording computer (TRC) located in a weather-proof box mounted inside the pit wall. The TRC converts the signal to a car number and records the precise time of day. The TRC passes all the data to a PS/2 microcomputer, which organizes the signals for all the scoring events and passes the signals to the USAC LapManager computer—also a PS/2. The LapManager program refines and forwards the timing information according to USAC's rules and procedures.

<sup>1</sup>IRIS is the acronym for Integrated Race Information System's collection of application programs written for scoring the Indy 500. It is not an IBM product.

## Innovative Client/Server Technology

An integral part of the IMS/USAC system is IRIS, a token ring attached to a token-ring LAN communication system and a centralized OS/2 database management system running on 11 PS/2 and ValuePoint computers.

OS/2 2.1, IBM's 32-bit multitasking operating system, DATABASE2 OS/2 (DB2/2) 1.0 SQL-based database manager, and OS/2 LAN Server 3.0 networking operating system form USAC's strategic client/server platform. Graham says that USAC officials using PS/2s can log on to the IRIS server, make basic queries, and get answers instantly.

LAN Server 3.0 provides industrial strength networking capacity to keep mission-critical, split-second racing statistics flowing among the PS/2s, which monitor the race from various checkpoints around the track.

IBM's Personal Software Products (PSP) division makes this software available to USAC. Headquartered in Austin, Texas, PSP is an industry leader in the development of operating systems, including PC DOS and OS/2, as well as networking software and other advanced technologies.

IRIS' ability to organize and store all the radio-frequency based data (such as a race car's speed at any point on the speedway) allows the timing and scoring team to more efficiently distribute

extremely detailed information such as the fastest lap of the race, current serial scores and standings for any lap or series of laps, elapsed time of any car on any lap, current aggregate pit time for all cars that are within 10 laps of the leader, pit times for every car in the race, and more. Unique to this system is its ability to report turn speeds and trap speeds for every car for every lap.

Most important, IRIS feeds that information to over 50 video monitors in the ABC Sports broadcast booths, the press box, and to USAC officials at other locations around the track. It also updates car numbers on the scoring pylon so that the spectators can see moment-to-moment standings.

### Trouble Tracking

Programmers from the IBM Lab in Boca Raton, Florida, helped USAC develop IRIS to handle the large amounts of data generated by the new antennas. IRIS runs on a 66 MHz PS/2 Model 95 server via a 16 MB token-ring network with nine IBM workstations. One database records high volume timing and scoring events; a second database manages administrative information, such as that needed by Bob Cassaday in the USAC Registrar's office throughout the month of May about entries, car numbers, drivers, lap speeds, and so on.

The IRIS system also helps USAC officiate the race, especially observers looking for infractions, such as passing under the yellow flag or pit lane speed violations. Chief Steward Tom Binford says that the DATA-1 antennas and IRIS system give executive officials valuable information. "I'm very impressed with the increased capability that the system gives us to spot infractions," he says.

The Dorian DATA-1 design, employing adjacent antenna loops at each antenna site, can also pinpoint the race car's lateral position on the track at each antenna. By analyzing the system's data, the crew chief can see the car's fastest line. Graham said USAC now routinely supplies race teams with information from the antennas during test runs, practice, and qualifying, as well as from the race.

### Indy and IBM—A Long-Term Partnership

OS/2 is the latest in a long line of IBM products used to support officials and racers at the Indianapolis 500. In fact, IBM employees, hardware, software, and services have been continuously involved at the Speedway since 1927.

In the fall of 1926, Chester S. Ricker, the Indianapolis Motor Speedway's first Director of Timing & Scoring, went to New York to meet with the president of a company that made punched card tabulating equipment. Ricker explained how using a system of punched cards would help him solve the problem of scoring high-speed race cars in what was then called the 500-Mile International Sweepstakes.

Always delighted to have customers develop new applications for his equipment, T. J. Watson, the founder and president of that New York-based firm, International Business Machines, leased key punches and tabulators on a special one-day basis to score the 1927 Indianapolis 500, thus forming a 67-year technology partnership between IBM and the Speedway. With its formation in 1956 to replace the defunct AAA Contest Board, USAC joined this partnership and continues to take the lead in applying current technologies to the information needs of the automated racing industry.

Because of the unbroken history of IBM partnership and performance, the Indianapolis 500 and USAC chose OS/2 running on PS/2s. IBM made more than 100 IBM Personal Systems machines available to support the activities of IMS, USAC, and race teams throughout the month of May.

Robert D. Lohman, Program Executive, Sports & Image Events, IBM Personal Systems, says IBM saw a great opportunity to lead with technology at the Indianapolis Motor Speedway and to assist USAC in developing the computerized IMS/USAC automatic transmitter-based timing and scoring system.

"USAC's mission at the Indianapolis Motor Speedway is extremely critical and depends upon the reliability of OS/2," he says.

### IROC and Winston Cup Join Team

Though the system was developed for Indy cars, IROC (International Race of Champions) series cars and the Winston Cup cars each carried a transmitter during their respective tire tests at Indy. For the most recent NASCAR tire tests August 16-17, 1993, transmitters were attached under each car. Data acquired during the test runs was printed out and presented to the teams and NASCAR officials for analysis.

Graham says IMS now requires any car that runs here in testing or practice to carry a transmitter to collect and report the information that the system generates. He says IMS and USAC will run the automatic transmitter-based system during the inaugural Brickyard 400 race for the Winston Cup cars on August 6, 1994 to supplement NASCAR's official timing and scoring operation. Graham says the USAC LapManager will supply turn and trap speeds to augment the start and finish line activities that NASCAR officials will record.

Graham says that IMS and USAC have created the most extensive electronic timing and scoring system in the world, but no matter how pervasive technology becomes, it will not replace the skill and judgment of people in the timing and scoring process.

"Without experienced officials to exploit this information, this race'd just be a bunch of cars going around in circles," Graham states.



**Patrick Karle** is managing director of Patrick Karle Associates, a high-tech marketing communications firm in Ann Arbor, Michigan. His specialty is IBM solu-

tions for the automotive industry. Patrick also teaches public relations at Eastern Michigan University.

# Software Compatibility: Good Relationship or One Night Stand?

*Bob Angell, a principal with Applied Information and Management Systems, discusses the evaluation criteria his firm used when first installing and evaluating OS/2. He also discusses the performance and benchmarking of applications running under OS/2, based on experiences within his organization and with his clients.*

When our firm discusses OS/2 with a potential user, we are often asked, "Will XYZ software package work well under OS/2?" This may seem like a trivial question, but whether you have a hundred dollars or a million dollars tied up in software, you want to be certain that it will work well under OS/2 before you invest time and money.

When OS/2 2.0 went GA (general availability) in April 1992, our firm, Applied Information and Management Systems (AIMS), decided that we would evaluate OS/2 based on five criteria:

1. Will our software work with OS/2?
2. What are the "gotchas" if it does work with OS/2?
3. Will the software performance be as good or better?
4. How severe is the learning curve with OS/2?
5. What happens if this does not work for our company?

With these questions in mind, we loaded OS/2 for evaluation.

After using OS/2 in a dual-boot environment, we found that the combination of hardware and software we were using produced unstable systems. We kept getting corrupted extended attributes whenever we booted into OS/2 after having last used DOS. Because of this instability, OS/2 was just not working well on our equipment. Boot Manager was necessary to resolve the dual boot and extended attribute problems in our early testing phases; it allowed us to isolate the operating system when it needed to be reinstalled or tweaked.

Since the availability of OS/2 2.0, we have used several DOS and Windows programs extensively:

- Paradox
- WordPerfect 5.1 for DOS and Windows
- Word for Windows 2.0c
- Borland C++
- ProComm
- Quattro Pro for DOS and Windows

We also use a number of other, obscure applications ported to DOS from the UNIX world (emacs, TeX/LaTeX, awk, etc.). We have found that all these programs work better with OS/2 than they did under DOS or DOS/Windows.

### Performance

OS/2's ability to multitask increased our programmer's productivity about 40 percent, especially with the database design and development side of the house. Compare the ability to go from window to window in OS/2, coding, testing, and making changes to four or five

applications at the same time with doing the same tasks under DOS, where the programmer could only work with one application at a time.

One of AIMS' areas of expertise is data integration over multi-platform computer systems. As a by-product, we design and develop customized databases for our clients on several different computer systems. We are also an Independent Software Vendor (ISV) with several OS/2 database applications under development. As a result of our work requirements, we use Paradox and its scripting language extensively for DOS-based client needs.

AIMS has developed several Paradox applications that, on occasion, require from a few hours to several days to produce the desired results. Before OS/2, we had to run all applications one at a time on a single workstation or dedicate several workstations to the task while being almost completely unproductive. With OS/2, we discovered that we could run these applications simultaneously with very little performance degradation.

To develop and program our database applications the old way was almost unbearable. We knew that development under UNIX was quicker and easier, but we didn't know that the Intel-based computers were capable of much more than using DOS. In those pre-OS/2 days, we would fire up the ol' editor to write the code, break out to Paradox (forget about Paradox's built-in editor; it's an oxymoron) to test and debug what we just wrote, then go back to the editor to make more changes. We continued with this

## ***Bi-Weekly/2:* You Heard it Here First!**

*Bi-Weekly/2* is a newsletter providing you with the low-down on OS/2: tips and traps, IBMLink hints, where to go for help, and what's the latest from "those in the know."

Published by Applied Management Informatics, LLC, and edited by Bob Angell, this informative newsletter will keep you up to date as you install OS/2 and OS/2 applications. You'll also get opinions, news about OS/2 user groups, information on the latest OS/2 games, and okay we'll admit it, even gossip.

To start *Bi-Weekly/2* coming to you every other week, send a check or money order for \$31.95 for a year's subscription (20 issues) to:

Applied Management Informatics,  
LLC.  
P. O. Box 58705  
Salt Lake City, UT 84158-8705

cycle until an application was written and ready for the client.

If you are writing and maintaining 50,000+ lines of code as we were, you may find it extremely tedious. OS/2's excellent DOS support has enabled us to speed up our coding, testing, and debugging cycles for all of our existing database projects, plus code maintenance has become much easier.

### **Benchmarking**

When OS/2 2.1 and Windows NT were new to the market, we ran some database applications on both operating systems to benchmark them against DOS 5.0 to see how they compared for AIMS' needs. We found that OS/2 2.1 runs 30 to 40 percent faster than DOS 5.0, and Windows NT runs about 20 percent slower than DOS 5.0.

We also found that the High-Performance File System (HPFS) works better for our development situation; in fact, a File Allocation Table (FAT) file system is extremely slow and too cumbersome for these databases. The HPFS actually sped up the database computing by almost another 20 percent. Overall, with OS/2 we are seeing a 50 to 60 percent increase in performance.

Until recently, HPFS had some problems reading and writing hundreds of small files without corrupting any data or getting a "sparse allocation" error. These problems were fixed with the latest OS/2 refresh (2.11). Here is an example of the magnitude of the disk activity required for some of our database applications: one database starts out using 5 MB (about 100 files) and expands to over 500 MB (about 10,000 files) of utilized DASD (direct access storage device).

OS/2 was chosen by default in several of our development projects because there wasn't a comparable DOS product able to handle the stress on the system. For example, printing reports (more than 2,000) for the database application discussed above would take 24 hours of continuous activity on a Hewlett-Packard LaserJet III with 4 MB additional RAM (normal rating on the HP is about eight pages per minute). Under normal conditions, this printer is beefy enough to handle medium to large print jobs. At peak times during the report printing process, the print queue contains an average of 600 to 700 files waiting to print.

When we tried to use DOS or DOS/Windows, the printer couldn't handle the workload and died miserably after approximately 200 files were queued to print! There was no DOS or DOS/Windows print spooler able to handle what we gave it. We were amazed; OS/2 could be chosen for its robust printing capabilities alone!

After testing all of our software with OS/2, our questions 1 through 3 about performance and compatibility were answered to AIMS' satisfaction. In fact, we were surprised at how well OS/2 worked; it allowed us to get more out of our DOS and Windows software than ever before. We did determine that when we make an investment in OS/2-specific software and

applications, as we have done in the last six months, it should be multithreaded and work well with the Workplace Shell (WPS) for optimal performance. This, however, is a topic for a future article!

### **Learning Curve**

AIMS' background stems from the UNIX world, which has dedicated system administrators to help curb and make sense out of chaos; pervasive password protection; telnet, ftp, and other file transfer quirks; and a vast web-like file system that we maneuver like the back of our hands. (By the way, wouldn't it be nice to have mountable file systems and better linking capabilities in OS/2? Just a thought . . .)

With this UNIX background, our learning curve for OS/2 was not a major concern. In the first few days we used OS/2, however, we experienced problems such as cross-linked attributes, corrupted extended attributes, and other weird, unexplained behavior to which we were not accustomed. These were not insurmountable frustrations but mere inconveniences that were slowly overcome with the use of online services such as the OS/2 Bulletin Board System (OS/2 BBS), user groups, and the IBM support line.

However, can you imagine a new user who can barely speak computer—let alone begin using one—with OS/2? With a pre-loaded system and a good tutorial, this learning curve can be minimized, but hand-holding is still required.

We have noticed that if our clients were never Windows users, OS/2's Workplace Shell is even more intuitive. Users who were heavy Windows users had some trouble using the WPS because its notebook concept was foreign to them (they were probably still looking for that .PIF file editor [big grin]). After users get used to the WPS, it is amusing to watch them go back to the Windows environment and try to use the right mouse button to change their icon (object) settings!

### **Installation**

We found OS/2's installation process to be a drawback. This operation is not for the faint of heart! In my opinion, this is one area that really needs some work before the masses can begin to feel comfortable using OS/2.



We constantly work with clients, friends, and colleagues to assist them through OS/2 installation. Heaven forbid if they want to change, add, or delete a peripheral in the system—this is not a trivial process! And the peripheral problems were even magnified by applying the latest OS/2 2.11 ServicePak (SP). Many users had problems applying the SP while in Super VGA (SVGA) mode, causing OS/2 to have to be installed from scratch. These problems were minimized after they restored their systems to the default VGA mode. We found that once the default had been reset to VGA, the SP could be applied, then SVGA resolution reinstalled without too many problems. This entire process must be simplified for the average computer user.

After mastering (we think) the above installation methodology, we feel more comfortable about tackling any combination of hardware or software problems that come our way. Installation is still not foolproof, however, because many hardware and software designs exist that are simply not compatible with OS/2 because of shortcuts in board design, trimmed budgets for research and development, etc. For example, I came across a client who had a printer that would not work with OS/2. After several hours, I determined there was a problem in the way the motherboard was talking to the parallel port. However, it worked fine under DOS because DOS doesn't utilize the system like OS/2.

### It Works for Us

In the end, we never needed to address our original concern of what would happen if OS/2 didn't work for our company. We cannot find another piece of software shipping today that fits our needs like OS/2. We have looked at and tested Unixware, Solaris for Intel, NT, and other operating systems without much success. We choose to use OS/2 because we want to take advantage of a shipping 32-bit operating system that works now—not something that other vendors promise will be ready sometime in the future.

However, IBM, can't drag its feet on getting fixes and changes to its end users and neglect its customers and their input for OS/2's enhancements. IBM like a good Boy Scout, must *be prepared*. We like OS/2 and we want to see it succeed.



**Bob Angell** is a principal with Applied Information and Management Systems in Salt Lake City, Utah, a management consulting firm. Bob specializes in multi-platform data integration, database design and development, simulation and modeling of complex environments (neural networks), and OS/2 software development.

# VNR KNOWS CLIENT/SERVER



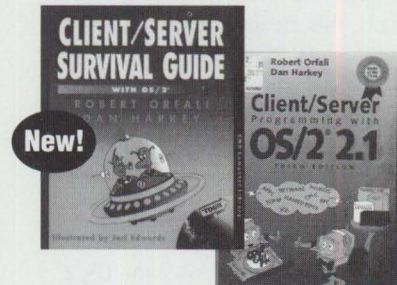
Ask for these and other VNR titles at your local bookstore.

## Van Nostrand Reinhold

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

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

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

Please circle #6 on reader service card.

# Migrating Windows Applications to OS/2: Easing the Migration Path

*With the rapidly increasing OS/2 installed base, tools are needed to assist in migrating current applications smoothly to OS/2. This article discusses the major issues of porting applications across platforms and reviews One Up Corporation's SMART product.*

With OS/2's increasing success, it is becoming paramount to many application developers to have native applications on multiple platforms. For many, this means migrating existing Windows application source code to OS/2. Cross-platform conversion of source code is time-consuming, however, and there is no magic header file to port your code automatically.

There are approximately 4,700 points of difference between Windows and OS/2, and this doesn't even cover the changes required for converting from 16-bit to 32-bit code or for using third-party libraries.

There's much more at stake than attempting to map application programming interfaces (APIs) from one platform to another. The port can become more difficult if the source code includes features that must be redesigned on the target platform. However, if you combine a complete understanding of both the OS/2 and Windows programming environments with having the right tools, you'll possess the key to making migration work.

## Porting Windows to OS/2—The Programming Perspective

Although both Windows and OS/2 are message-based environments, the formats of these messages, as well as the APIs, differ substantially between the two. A number of functional areas found in common with most applications cause the most difficulty and effort in the porting process. While many areas seem to have simple solutions, they can cause you a lot of rework or redesign to integrate properly into the ported code.

Following are some common major issues:

- Resource files must be converted from Windows to OS/2 format.
- Multiple Document Interface (MDI) must be implemented in the application.
- Object linking and embedding (OLE) is not supported in OS/2.
- The OS/2 coordinate system is lower-left origin versus Windows' upper-left origin. All coordinate calculations for positioning windows and drawing must be converted to be relative to the lower-left corner.
- In OS/2, processes must explicitly gain access to shared memory, and all processes having access to shared memory must free it before the memory is freed. This differs from Windows where the shared memory is freed when the creator of the memory frees it.
- Applications have less direct control over printing properties in OS/2. Applications should use the job properties dialog to allow users to modify the printing characteristics.

How much the application exploits and depends on these items will affect how easy the migration will be. Separating

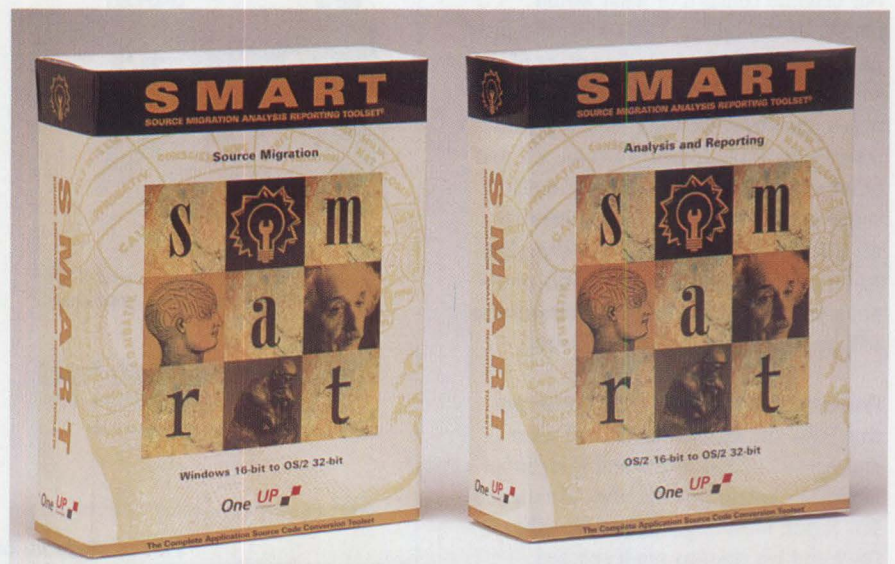
these items into platform-dependent objects is one solution for reducing the effort of porting.

## Porting Tools

Many tools are available that assist in some phases of the porting process. Tools such as GREP (shareware) and OS/2's Seek-and-Scan can help identify individual features in the source. Basic text editors can perform simple text replacement. More sophisticated editors such as SourceLink from One Up provide hyper-link access for source replacement.

The Mirrors toolkit supplied by One Up provides a layer implementing the Windows functionality either internally or via multiple calls to OS/2. Since the Mirrors toolkit was designed as a tactical solution for quickly migrating an application to OS/2, it is not intended as a long-term strategic solution.

A toolset developed by One Up Corporation can assist in all phases of the migration process. These tools, known as SMART—Source Migration Analysis Reporting Toolset, consist of an analysis and reporting tool, a source migration tool, and utilities such as the resource translator. An expert system, the SMART



Toolset was designed to size the conversion effort and provide a road map for the migration. The SMART Source Migration Tool not only changes up to 70 percent of the API and message code, but also continually provides reports to help programmers migrate the remainder of the code.

Industry analysts and software developers believe SMART will open up the market for native OS/2 and Workplace OS applications by automating a large portion of the task of migrating 16-bit code to 32-bit code that will run under OS/2. Perhaps more importantly, SMART provides a path to migrate Windows applications over to the new PowerPC platform.

IBM believes the SMART Toolset will help stimulate sales of OS/2-compatible applications once leading software developers convert their Windows applications. But SMART is not just an OS/2 migration tool. It is a source code massager that can migrate applications *from* any platform *to* any platform.

SMART can size a conversion effort in one day, producing a report that tells developers how long the effort will take and how much it will cost. Sizing of applications is one of the greatest impediments to converting to OS/2, and SMART can shrink up to six person-months of effort into one day's work.

In addition to reporting the number of points of difference between platforms, SMART also reports the relative difficulty of porting an application, with category 050 requiring the most effort and 000 requiring the least. The average of all the data collected leads to a general observation on the distribution of the category of items and where the effort is likely to be spent in porting an application.

Figure 1 shows the frequency of occurrence of each category of items found and the distribution of the effort involved in porting each category. All of the category 010 items and most of the category 020 items can be ported with an intelligent global search-and-replace utility, resolving about 70 percent of all items to be ported. However, it would only complete 17 percent of the required effort.

# The Five Phases of the Porting Process

**P**orting source code comprises five phases, some of which overlap: analysis, automated code replacement, computer-assisted code replacement, implementation of unsupported features, and addition of platform-specific features. Automated tools can help process the first three phases; additional tools can assist with the last two.

## Phase 1: Analysis

Analysis of the code to identify and report all environment-specific issues and amount of porting effort required. This includes a breakdown of all API calls, type definitions, symbols, and messages, including their frequency of occurrence and difficulty of porting. The analysis provides a detailed look at your source and what specific features of the environment you use.

## Phase 2: Automated Code Replacement

Automated code replacement of those items that have a one-to-one mapping from the source to target environment. Also included in this phase is the conversion of resource files.

## Phase 3: Computer-Assisted Code Replacement

Interactive code replacement with input from an application developer for those source items that have an equivalent feature in the target environment, yet require a decision as to either the original intent of the source or which of several choices to use in the target environment.

## Phase 4: Implementation of Unsupported Features

There will ultimately be some features of the source environment that are not directly supported in the target environment. In some cases, it may be possible to simulate these features; in other cases, it will not be possible. The developer will have to provide input to make the changes.

## Phase 5: Addition of Platform Specific Features

Tighter integration of the application with the target environment might be desirable from a marketing or even coding standpoint. This adds features that might make it more difficult to port to other platforms but can add significant benefit to the end user.

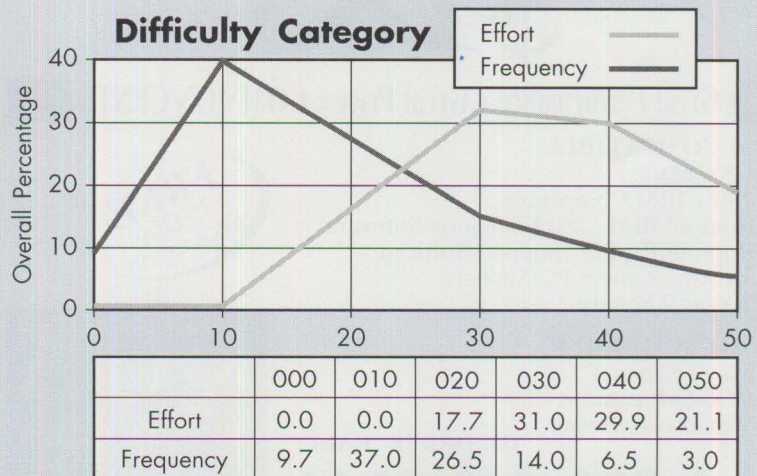


Figure 1. Frequency of Occurrence vs. Effort

Has a software problem ever cost you time, money, or aggravation?

**QES's Better Software makes your Software Better:**

**QES/Architect** for DOS, OS/2, Windows - complete QA & process management  
**QES/EZ** for OS/2 GUI - Quick & easy capture/replay for OS/2 PM GUI

- \* **Manage data:** requirements, specs, validations, tasks, verification, schedules, testing, performance, project, quality, results, etc.
- \* **Reposit Data** items with validation & rule data.
- \* **Capture / replay** testing & QA functions.
- \* **Generate Test Data** via menu picks.
- \* **Edit and maintain** all data *Globally*.
- \* **Prototype Self** - validating specifications.



**QES Inc. 20 Westbrook St.  
E.Hartford, CT 06108-3447**

**QES /Architect** is a PC- based, menu-driven, system that is easy for **both technicians & end users** to use to test any host-emulated or character-based applications from **DOS, Windows, or OS/2**. It automatically generates WYSIWYG pictures of your own application, discerning fields, data items, and responses by studying captured data. You can manage the entire software manufacturing and acceptance process from a Quality Assurance perspective. With QES's relational database, you can control software projects from beginning to end. QES EZ for OS/2 GUI includes simple management and variables.

**No programming ever needed!**

QES target environments include 3270, 3X/AS400, VAX, DOS, OS/2, RISC 6000, HP, etc, via any emulation supported by OS/2, DOS, or Windows

Call us at **203 289 2227** or **FAX 203 289 2009** and find out how to:

**Integrate the MANAGEMENT and AUTOMATION of your software production process**

Please circle #7 on reader service card.

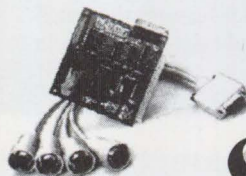
The SMART Toolset, along with migration consulting and support from One Up, works on the remaining 83 percent of the effort by identifying items to port, recommending the appropriate OS/2 code to replace the Windows code, and allowing you to hyper-link among all occurrences of an item.

In addition to porting source code, SMART also includes a resource translator for converting Windows resource files to OS/2 resource files in native format and for converting Windows icons and cursors to OS/2 icons and cursors.

The market for porting and migration tools represents a cottage industry for which the support structure is still being formalized. Until SMART, it hasn't been viable to migrate natively from one platform to another. With the tool, developers will benefit from increased productivity and development cost savings.

For more information on SMART and services available from One Up Corporation, please call (800) 678-01UP.

**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**



**Jeff English** is the Manager of Research and Development with One Up Corporation and is currently working on development of cross-platform porting technology.

He co-authored *Real World Programming for OS/2 2.1* and holds a master's degree in computer science from Iowa State University.

# OS/2 Conference Draws Praise From Technical Coordinators

A record 2,600+ technical coordinators, software developers, LAN experts, MIS managers, and device driver developers converged on San Francisco at the end of April for the Personal Software Products Technical Interchange. The event represented the largest gathering for an IBM PSP conference and is the latest example of the growing industry momentum behind the 32-bit development environment for OS/2.

Technical coordinators, members of IBM's special program of support for large personal computer environments, were there in force, representing almost half of the paid attendees. In addition to

participating in the 200+ technical elective sessions, technical coordinators were treated to a private reception and special breakfast held exclusively in their honor. They received free t-shirts, the Technical Connection CD-ROM, luggage tags, and education discounts, plus they had an opportunity to win over 75 exciting raffle prizes.

Keynote speeches by the likes of Lee Reiswig, IBM's Personal Software Products president, emphasized the future of the 32-bit OS/2 operating environment and highlighted some of the advanced applications that will be available this year.

These technical conferences are designed to share technology, strategy, and tools with all conference attendees. Technical coordinators and product developers enjoyed opportunities to talk "shop" with peers with whom they share interests and challenges. "IBM's got a good roadmap for where we need to go in the future. I like the openness and connectivity and their commitment to providing solutions. . ." states one satisfied attendee.

Watch this space for the next Personal Software Products Technical Interchange. You won't want to miss it!

## Avoid the diskette shuffle while saving both time and money—

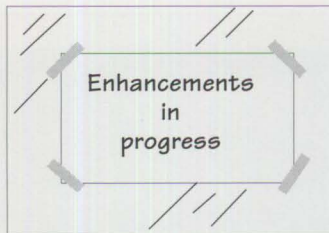
Let LAD/2 automatically install and distribute your operating systems and applications over multiple workstations.

### 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  
**LAD/2** to find out what  
LAN Automated Distribution/2 LADDie is up to.

(International customers please call 817-961-5232)



**LAD/2  
LAB**

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



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

Please circle #9 on reader service card.

PERSONAL SYSTEMS • JULY/AUGUST 1994

# DCE: An Application Primer

*The Distributed Computing Environment (DCE) gives developers a base of software for building client/server systems for a wide variety of computers and operating systems. This article presents a tutorial on the basics of DCE and how developers use it to build client/server systems.*

If you could build the client/server system of your dreams, how would you do it, and what would it look like? Depending on your background, you might prefer a PC-only solution. On the other hand, if you are a mainframer, you would probably use a mainframe and 3270 terminals to build your system. Finally, if your platform of choice is UNIX or AIX, you might tie workstations to a UNIX host.

Each environment certainly has inherent strengths and weaknesses, so which is best for building client/server systems?

## Homogeneity vs. Heterogeneity: Familiar vs. "Best" Solution

When a company decides to create a client/server system, its staff must make a series of decisions about the hardware and software to use for building their design.

Typically, developers prefer to build systems that use the same operating system, hardware, and network components to build both the client and

as *heterogeneous* systems (see Figure 2). These systems optimize cost and performance, while allowing development staff to work with the types of environments they know best (UNIX systems for UNIX people, VM and CICS for mainframers, and so on).

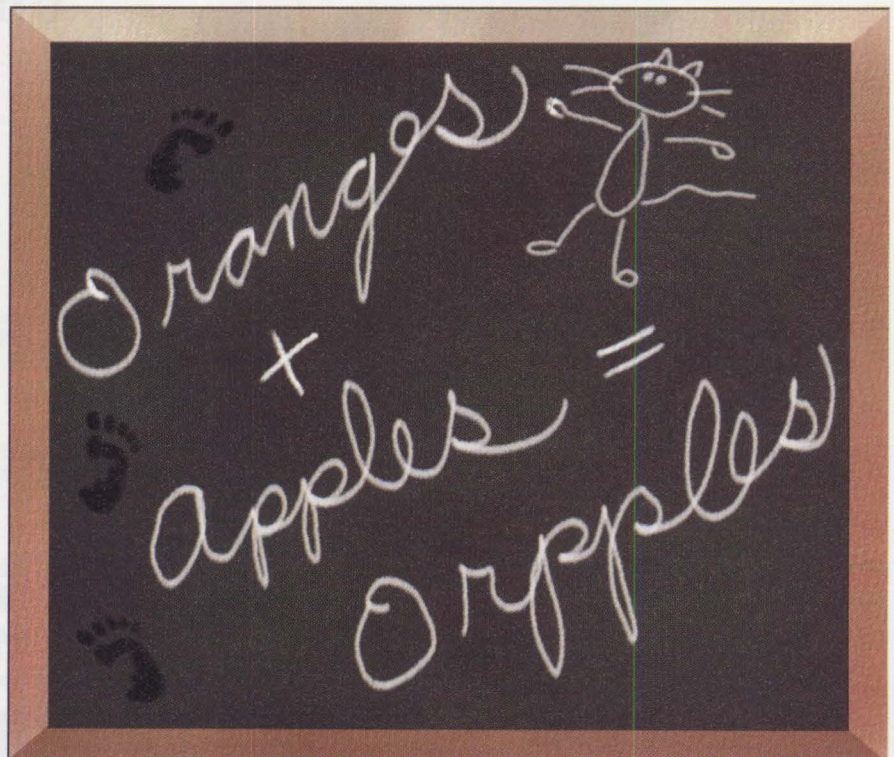
The trick is tying all of this work together. The truth is that heterogeneous systems are hard to integrate. Developers attempting to integrate many different systems find themselves muddled in the incompatibilities of data formats, security, routing, error recovery, and missing or incompatible operating system features. The fundamental problem is the lack of common vocabulary.

The Distributed Computing Environment, known as DCE, solves the heterogeneous system problem. It provides an array of common network services, as well as interface software, that allow developers to create heterogeneous client/server

**Philip Lieberman**  
Lieberman & Associates  
Beverly Hills, California

server sides of the system. When both the client and server use the same platform, it's called a *homogeneous* system. These types of systems are preferred because they are familiar to work with and to maintain. However, homogeneous systems are typically either expensive when they use a powerful platform or very limited in upward scalability if using low-end PCs (see Figure 1).

Some developers build client/server systems that use a variety of operating systems and hardware; these are known



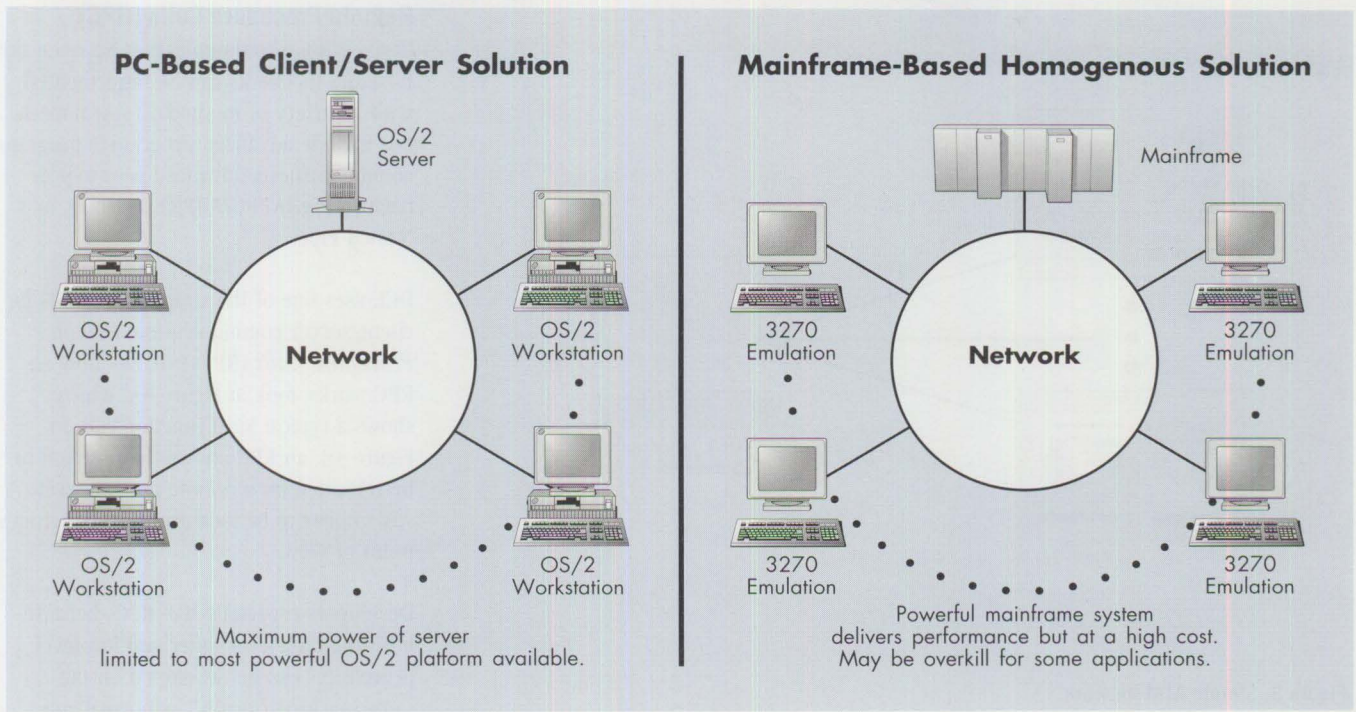


Figure 1. Homogeneous Systems

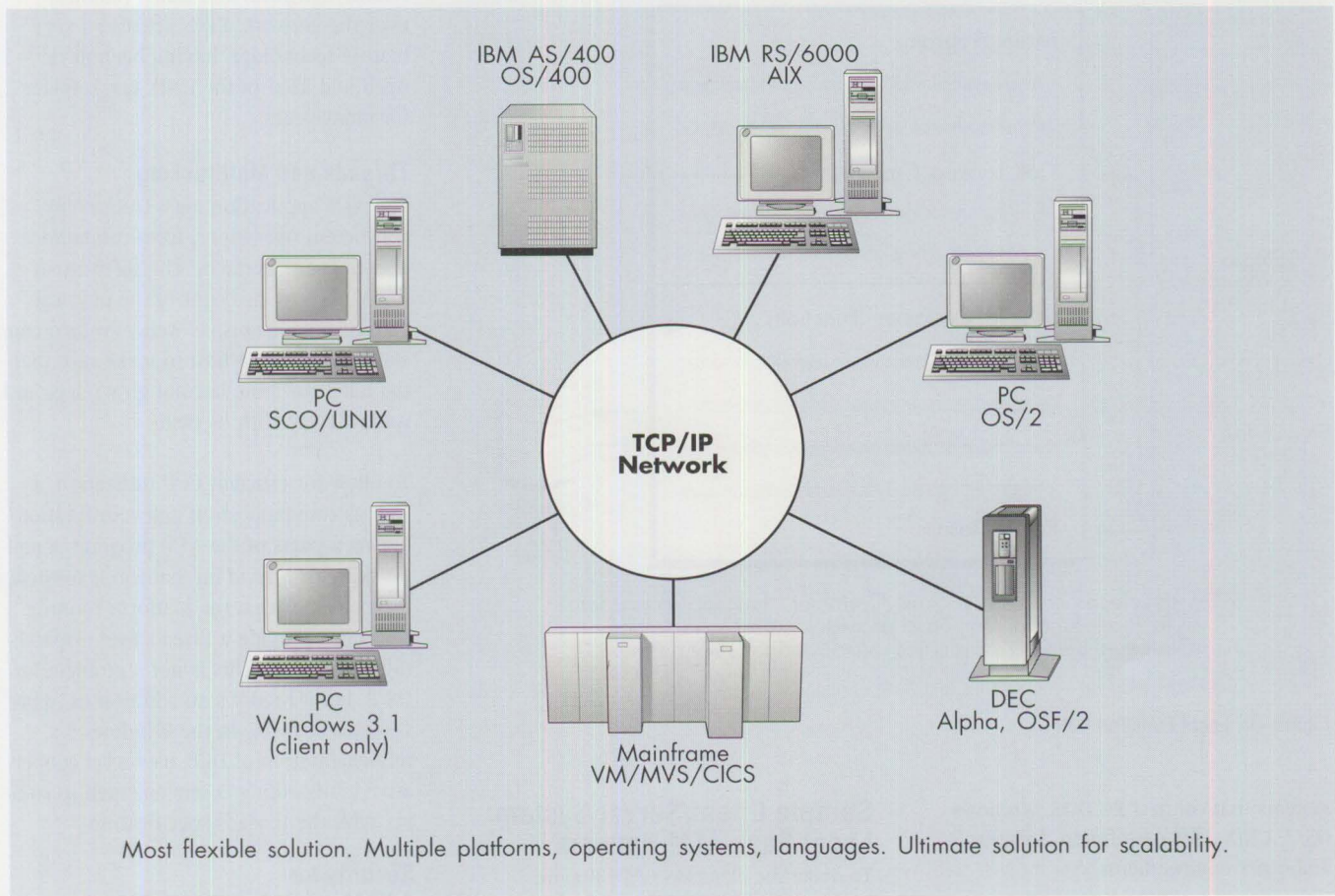


Figure 2. Heterogeneous Systems

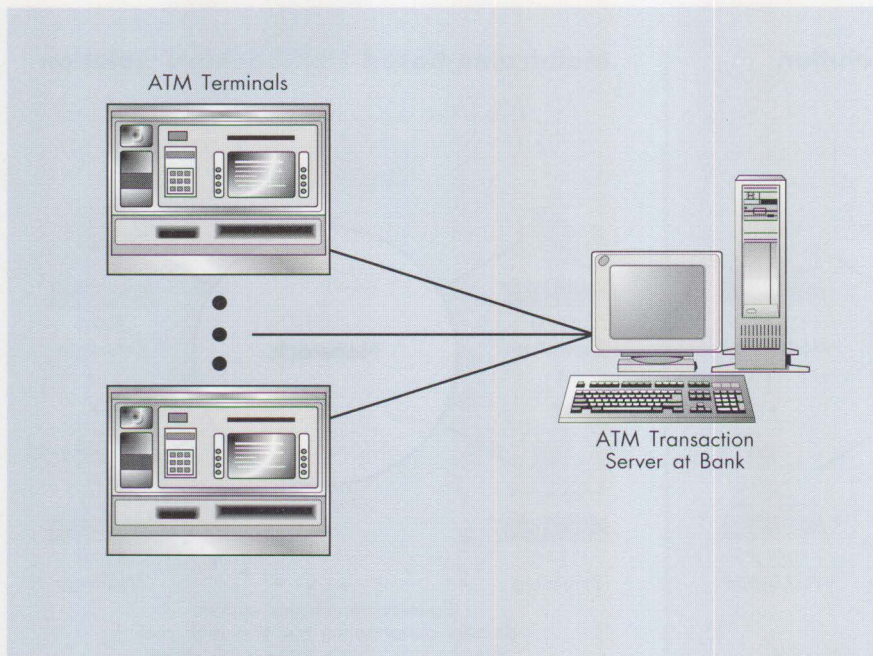


Figure 3. Simple ATM Network

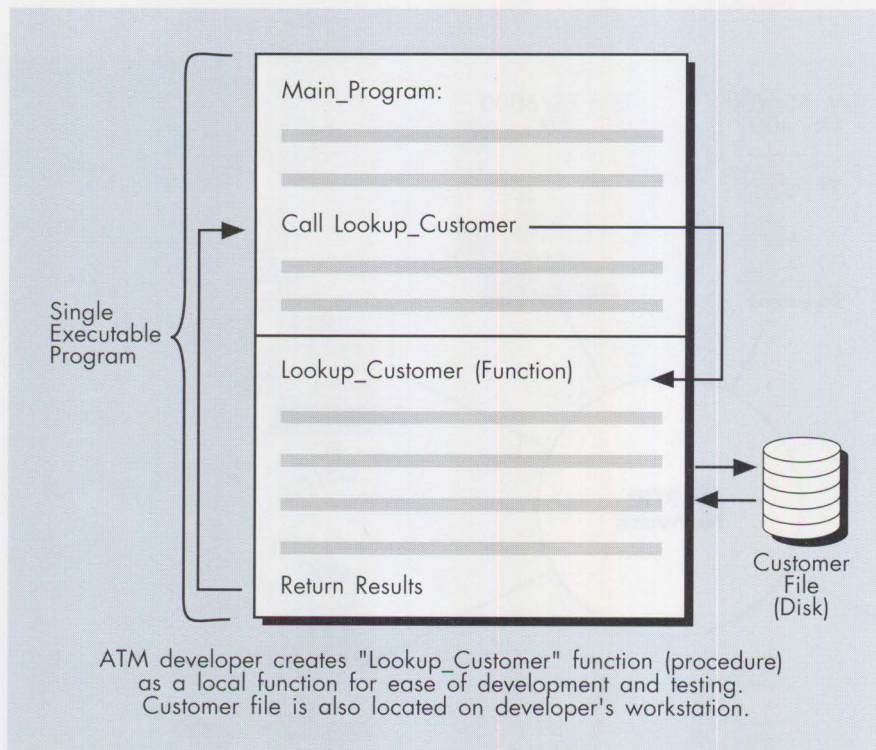


Figure 4A. Local Function Call

systems that connect PC DOS, Windows, OS/2, UNIX, IBM mainframe, AS/400, and other environments.

The key behind DCE is that all of these systems use the same interfaces and can talk to each other—today.

### Sample Client/Server System: Local Bank ATM Network

To make the elements of DCE more concrete, let's create a prototype automated teller system using DCE. Figure 3 shows a series of automated teller machines (ATMs) connected to a mainframe or server system. How would we go about using DCE to run this system?

### Remote Procedure Calls (RPC)

Conventional communication between the host and its clients can be implemented with a variety of methods. Typical methods include advanced program-to-program communications/advanced peer-to-peer networking (APPC/APPN), NetBIOS, and Named Pipes.

DCE uses one of the simplest methods for client/server communication: *Remote Procedure Calls* (RPCs). To see how an RPC works, look at Figure 4A, which shows a typical local function call. In Figure 4B, an RPC allows the function to be relocated on a remote machine, yet still appear to be local from the program's point of view.

Developers especially like RPCs because both the high-level caller and low-level procedures can be developed on the same machine (locally) as normal subroutines. Once the subroutine functions are debugged, they can be relocated to remote machines and called remotely over the network. RPCs are also a very mature technology, having been developed and used in the UNIX marketplace for many years.

### Threads and Multitasking

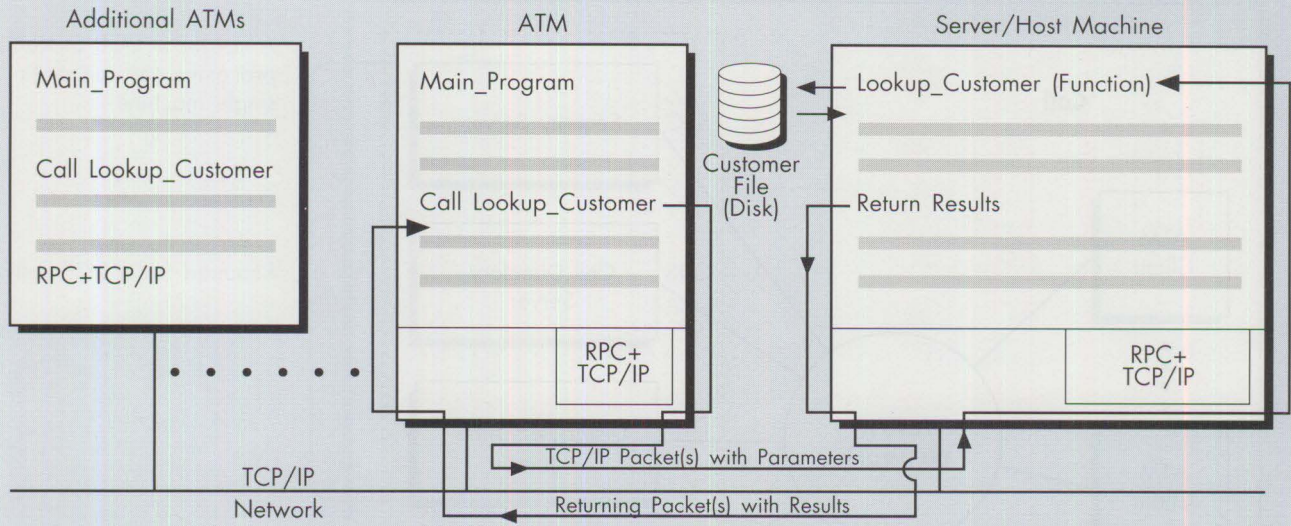
The ATM application must track many concurrent operations, from communication to user interfaces. The software at the ATM must also control the physical cash disbursements, as well as return constant status to the host to make sure that the teller machine has not grown legs and walked away with its cash.

To allow for efficient CPU utilization, as well as communication and coordination between parts of the ATM program, a multitasking or *thread* mechanism is needed. DCE specifies that the platform running DCE must provide a thread-level dispatching mechanism. This is not a problem for OS/2, but it requires an addition to some versions of UNIX. In the Windows 3.x implementation of DCE, user-level cooperative multitasking is implemented to comply with the threads specification.

### Security for Authenticating Transactions

Now that we are communicating with the host, how can we make sure that a real ATM is communicating with the host—not an interloper who has taken over the phone lines? This part of the transaction





The Remote Procedure Call (RPC) mechanism of DCE allows the "Lookup\_Customer" function to be called remotely by ATM. The fact that the "Lookup\_Customer" function is on another machine (Server/Host) is completely transparent to the original "Main\_Program" running on the ATMs.

Figure 4B. Remote Function Call

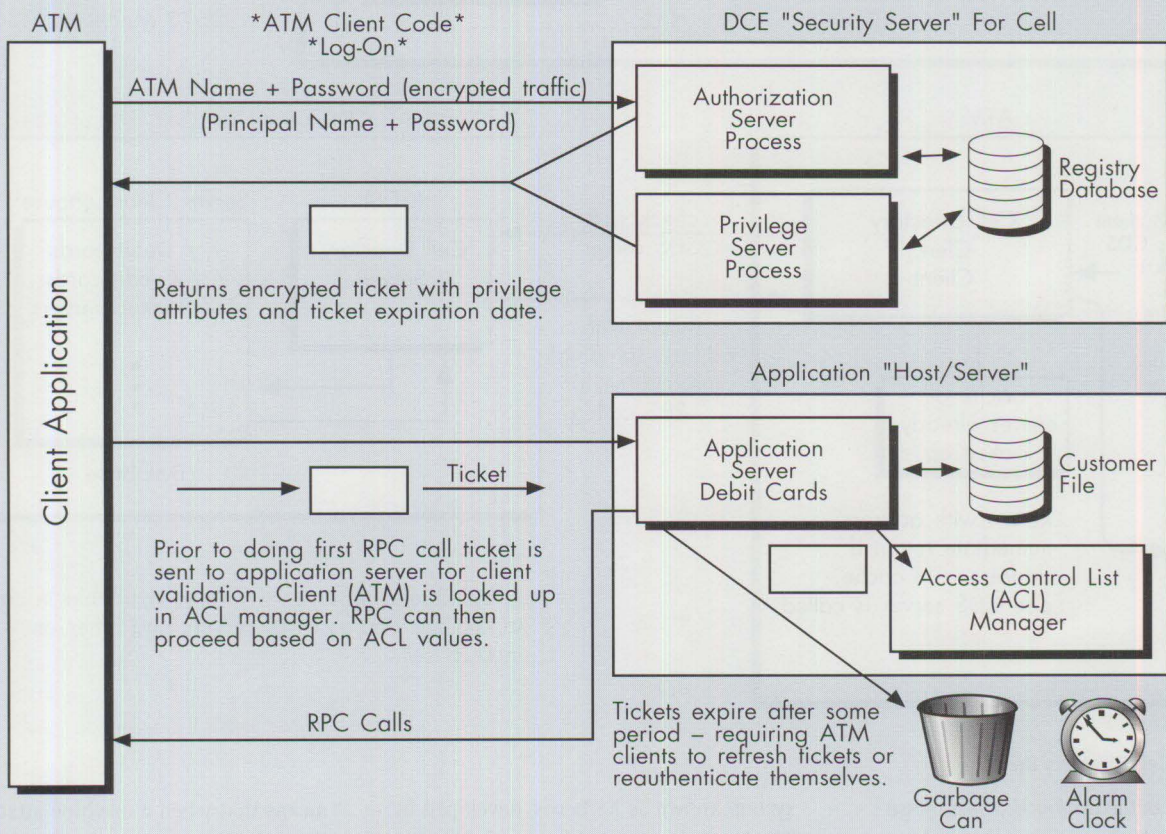
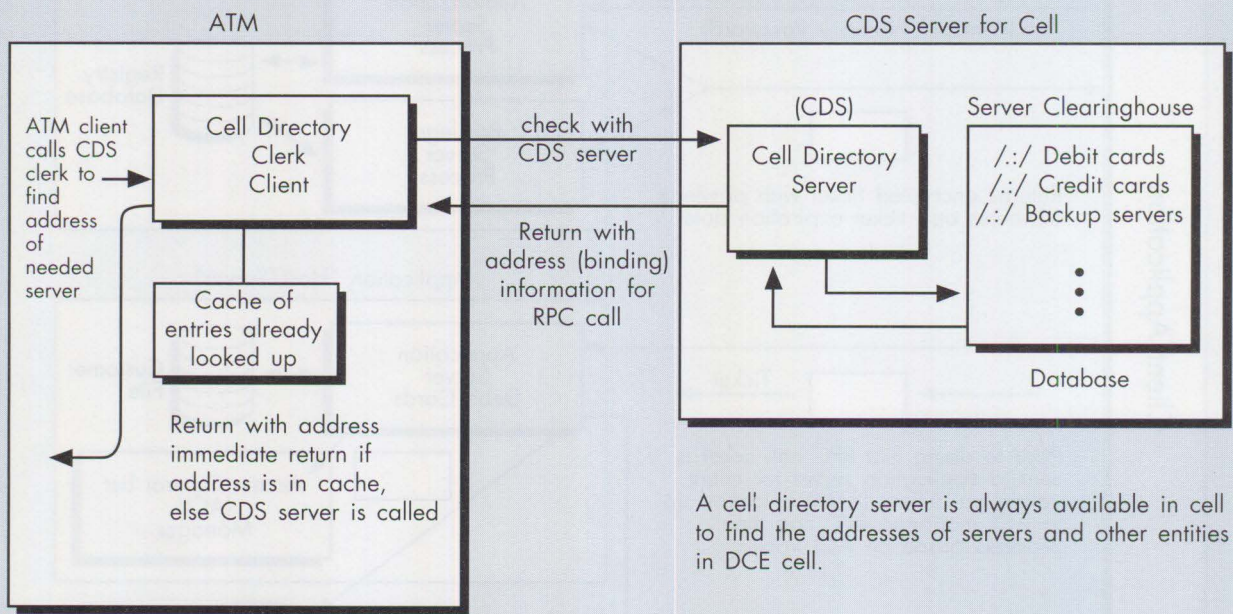
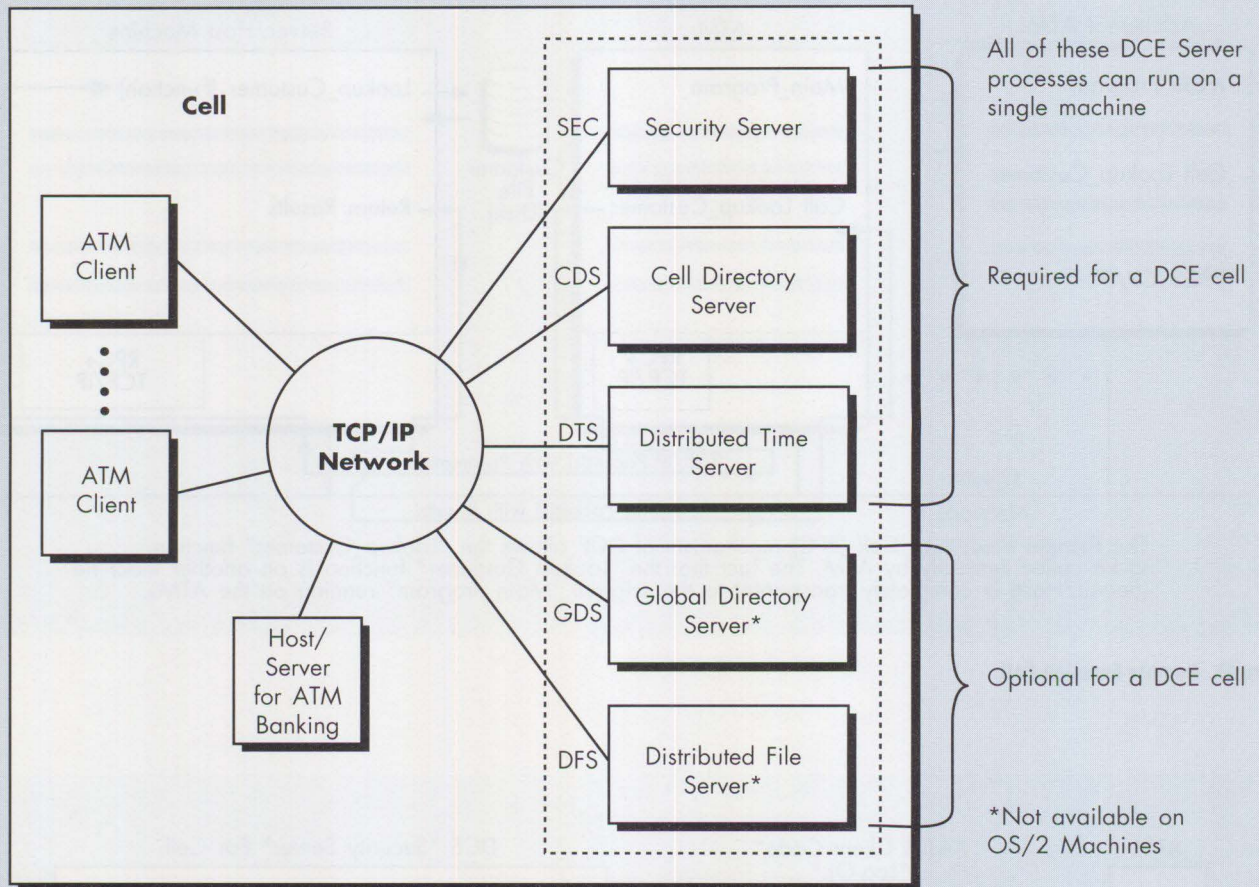


Figure 5. The Authentication Phase



**Figure 6. Cell Directory Server**

is known as the *authentication* phase (Figure 5). Authentication should be implemented so that the information being transmitted is encrypted and only the sender and receiver have the proper keys to confirm their identities. DCE implements a sophisticated security

system known as Kerberos, developed by The Massachusetts Institute of Technology. This security system is based on the concept of *principals*. Applications, users, and even machines are given principal names. Each principal is also given a password for authenticating itself. This

means that when a machine attaches to the network, it must prove its identity before DCE gives it any attention. Applications and users that call all sorts of servers may also be required to provide a password for gaining access.

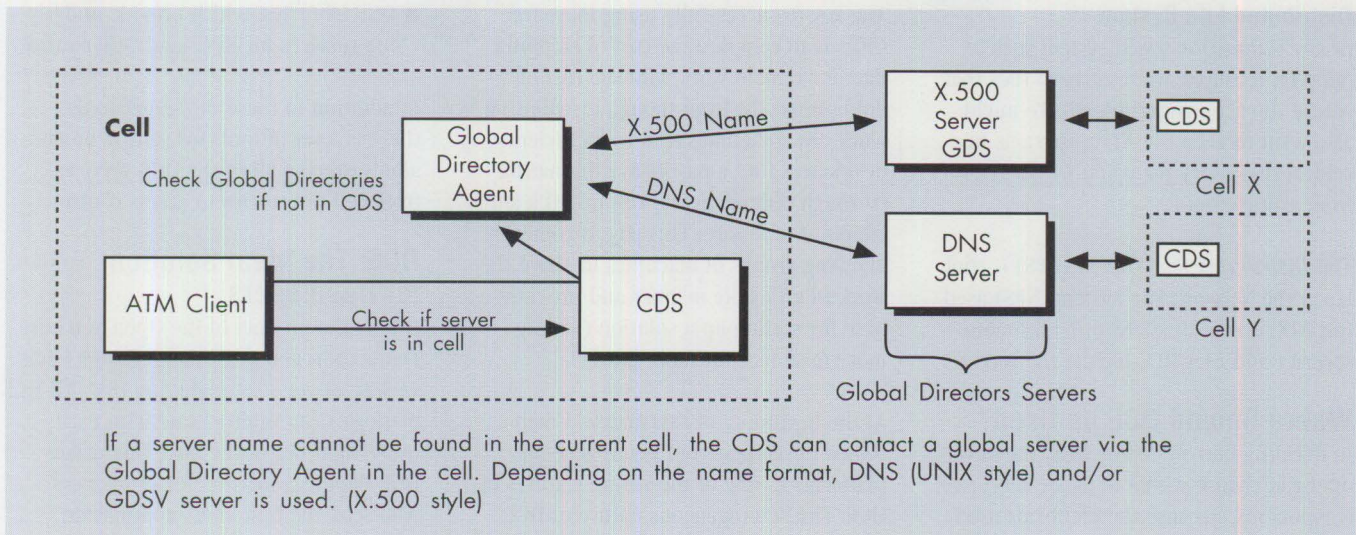


Figure 7. Global Directory Service

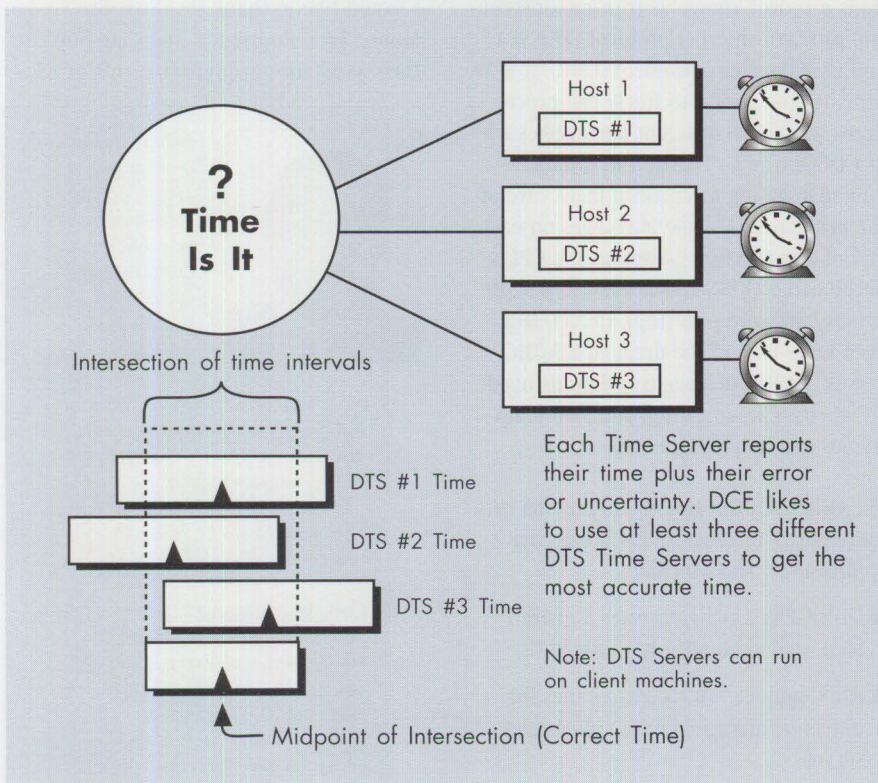


Figure 8. Distributed Time Servers

By allowing the server developer to set the appropriate level of protection based on a principal's identity, DCE provides a great deal of flexibility in security.

### Finding Servers - Cell Directory

Imagine now that a customer inserts a credit card into the ATM instead of the expected bank debit card. Since the default host or server at the bank does not handle credit-card transactions,

another server must be found to deal with the transaction.

To help organize the ATMs and servers for security and administration, we create a grouping, called a *cell*, to which all of the clients and servers belong (see Figure 6).

To find the proper server for the type of bank card tendered, we use the DCE *cell directory server* (CDS). The cell directory server, shown in Figure 6,

contains the names and network locations of all the machines on the network as well as the different credit-card servers and their services.

### Non-Local Bank Card - Global Directory

Suppose someone using the teller machine inserts a credit card from another country. None of the servers within the cell is set up to deal with a foreign card. However, there are servers outside of our bank cell that are available to all banks for this type of transaction (Figure 7). To look up this server's network address, we use a special server known as the *global directory server*.

Why use a special server to come up with other servers' network addresses rather than hard-code the addresses? Because the special server gives the banking system's administrator the ability to dynamically add more servers (to spread the load), as well as to move assets around (if it becomes necessary to temporarily replace one machine with another).

### Time Services

To handle our foreign card user, suppose we had to go to a server in another country. Since we want to keep our local database of transactions in synch with the foreign bank's server, we need some sort of accurate common time standard, or *distributed time server*, to keep things in synch. To accomplish this, we could designate a single server, or a group of servers, to keep the time reference (as in Figure 8).

## Distributed File System

In any transaction-based system such as our ATM example, we must keep files that record the actions performed. We might also want to keep the ATM programs in a central repository for updating the ATMs from a single point.

The *Distributed File System* (DFS) is similar to the Network File System (NFS) used in UNIX, except that DFS provides more security and uses RPC for the file access.

## Where Should DCE be Used?

In deciding how to deploy DCE, the developer can choose where to locate server components. Because the platforms used for servers are multitasking with thread services, a host machine can run multiple server processes simultaneously. This feature allows a single machine to run many of the cell services such as Time Server, Cell Directory Server, and Security Server simultaneously.

The DCE cell administrator decides where to locate each of these DCE system processes. Breaking them up onto different machines might increase performance or reliability. DCE can replicate the Cell Directory database to multiple server machines for load balancing and reliability. In a future version of DCE (the current OS/2 version is 1.0.2), the single cell security server's database will be able to be replicated.

When designing a distributed application using DCE, developers value the ability to distribute the workload for tuning overall performance. Because most machines on the DCE network can function as both clients *and* servers, the developer can now exploit unused CPUs on the network.

The key to successfully using these free CPUs is to compare the cost of shipping data for remote work with the cost of doing it locally. Even though distributing work can sometimes lead to inefficiency, developers find a payoff by eliminating (through redundancy of resources) a single point of failure. This payoff comes by allowing groups of machines to share the work in a flexible manner and provides a base for scaling up a solution (adding more machines for more power).

At the beginning of this article, I mentioned development of heterogeneous platforms as one of the strong features of DCE. The heterogeneous nature of DCE stems from a common network protocol as well as machine and language independence. DCE uses a common communications protocol known as transmission control protocol/internet protocol (TCP/IP) and user datagram protocol (UDP). It also provides the ability to freely call procedures written in different languages such as COBOL and C. The DCE RPC parameter's marshaling mechanism takes care of the peculiarities of the languages on each end of a connection as well as the CPU's peculiarities (RISC versus CISC and MSB versus LSB word ordering). In fact, DCE even takes care of the dreaded EBCDIC versus ASCII code-page translations needed when communicating with IBM mainframes and non-IBM machines.

The DCE tools for developers provide this universality through tools for creating applications. Tools include:

- UIDGEN, which generates a guaranteed unique identification for RPCs
- IDL Compiler, which creates files for universal linkage between client and server

- DCE Libraries, which provide underlying services for RPCs and DCE control

In addition to these developer tools, there is a set of tools for setting up and administering all of the DCE services, from cell directories to access control.

## DCE: The Ideal Solution for Developers

DCE is the product of the Open Software Foundation and is available from a wide variety of vendors, including IBM. All DCE platforms can interoperate with each other right out of the box. This makes DCE the ideal solution for developers who seek the best, most appropriate technologies available.

## Acknowledgments

I would like to thank Dr. Chris Brown, Ian Moore, Fred Rodriguez, and Clay Boyd for their assistance in preparing this article.



**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.

Lieberman and Associates  
221 N. Robertson Blvd., Suite C  
Beverly Hills, CA 90211  
Voice (310) 550-8575, Fax (310) 550-1152, BBS (310) 550-5890  
CompuServe 76426,363, IBMLink:  
DEV2203, PRODIGY: PKFJ82A

# YOUR *single* SOURCE FOR OS/2® APPLICATIONS

Choose from a wide selection of the latest 32-bit applications at discounted prices

**ORDERS:**  
**1-800-776-8284**  
FAX: (919) 878-7479  
INQUIRIES: (919) 878-9700

Authorized IBM Reseller

OS/2 and IBM are registered trademarks of IBM Corporation



PRODUCTIVITY SOFTWARE  
GRAPHICS & CAD  
COMMUNICATIONS  
NETWORKING  
WORD PROCESSING

UTILITIES  
BOOKS & VIDEOS  
OS/2 T-SHIRTS  
BACKUP SOFTWARE  
DEVELOPMENT TOOLS

**FULL COLOR  
CATALOG  
NOW AVAILABLE**

3209 Gresham Lake Road Suite 135 Raleigh, NC 27615

# Distributed Performance Characteristics of IBM DCE for OS/2

*This article presents distributed performance characteristics of the IBM Distributed Computing Environment for OS/2 Software Development Kit Version 1.0 (hereafter referred to as DCE for OS/2). The studies presented here use hardware and software options to distribute the application workload within a heterogeneous, single-cell environment: DCE for OS/2 and IBM AIX DCE/6000 Version 1.2 (hereafter called DCE/6000). Application performance and random access memory/direct access storage device (RAM/DASD) resources were also observed while the DCE Security registry was increased to several hundred entries.*

*An implementation of the powerful DCE performance/resource-balancing tool, the nested Remote Procedure Call (RPC), is examined.*

---

**Benetta Perry and Bob Russell**  
IBM LAN Systems Performance  
Austin, Texas

---

**F**or this study, we used entry-level IBM PS/2 and IBM RISC System/6000 servers to (1) establish minimum performance expectations and (2) identify bottlenecks that, because of the increased processing power available in other models on these hardware platforms, may not be evident. To gather performance measurements and

determine the distributed performance characteristics of DCE for OS/2, we selected five server topology models, shown in Figure 1. These models represent environments ranging from a small, single-server DCE cell to a moderately sized distributed enterprise. The five server topology models are:

1. **Single-server environment.** This model represents a small enterprise, with one server providing both the DCE core services and DCE applications.
2. **Two-server environment.** In this model, one server is the dedicated DCE server, and the second server is the application server. In this study, we evaluated PS/2 and RISC System/6000 servers in both roles.
3. **Multiple application servers.** This model is probably closer to real life; it has one DCE server and multiple application servers. Using this model, we



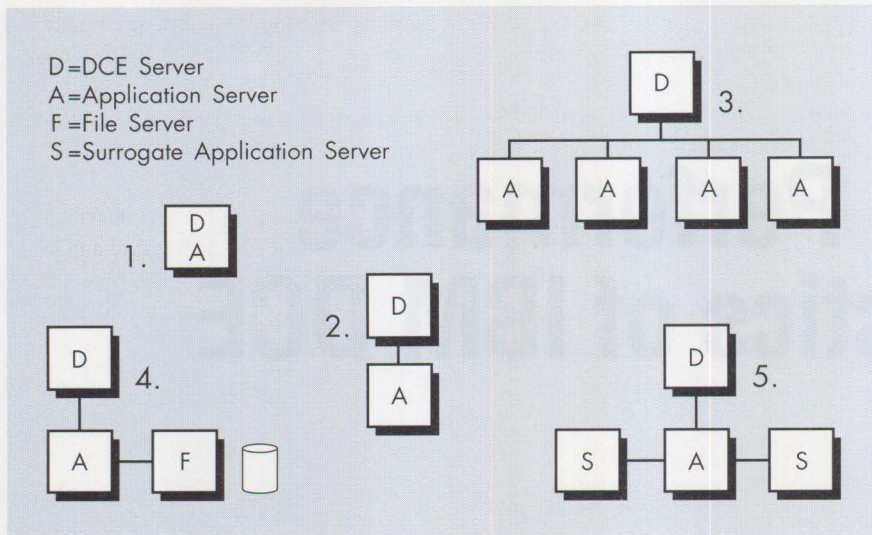


Figure 1. Five Server Topology Models

**RPC API calls to establish a new binding with a PoS application server:**

```

rpc_ns_binding_import_begin
rpc_ns_binding_import_next
rpc_ns_binding_import_done
rpc_ep_resolve_binding
rpc_binding_to_string_binding
rpc_binding_set_auth_info (Call-level authentication)
  
```

**PoS RPC transactions:**

- CATALOG: Get the image file of a catalog page.
- CUSTOMER: Look up the customer's account information by using the customer's phone number.
- PRICE: Look up the price and inventory information of 1 to 4 (average of 2.5) items selected randomly from the catalog page.
- HISTORY: Write a sales history record to the history file.

**RPC API call to terminate the binding with the PoS application server:**

```

rpc_binding_free
  
```

Figure 2. Components of a PoS Customer Sale

- looked at the DCE server throughput while the amount of DCE application activity was incrementally increased to more than 500 clients.
4. **File distribution options.** A variety of off-the-shelf file distribution options are available for OS/2 and AIX. In these tests, we compared the IBM LAN Server 3.0 Advanced and Entry, and IBM Transmission Control Protocol/Internet Protocol (TCP/IP) Network File System for OS/2 and AIX.
  5. **DCE application distribution.** In this model, we distributed discrete parts of the DCE server application to surrogate servers by using nested DCE Remote Procedure Calls (RPCs). This model incorporates some of the advantages of models 3 and 4 while exploiting the cooperative processing strength of RPC.
- In our tests, we used PS/2 servers—a Model 8580-A31 (25 MHz) and a Model

8595-0KD (33 MHz)—and a RISC server—a RISC System/6000 Model 520. These entry-level OS/2 and AIX servers provide similar performance horsepower and are thus appropriate for cross-platform evaluation. Both platforms now have faster CPU and I/O hardware options that provide significantly better performance. However, the goal of these studies was to establish minimum performance expectations.

The DCE client workload for these studies was provided by 25 PS/2 systems. We used the Point-of-Sale (PoS) benchmark developed in our LAN Systems Performance Lab in Austin as the application workload.

An enterprise's environment may not be exclusively DCE but will more likely be an interactive environment of several components and applications. Individual departments may be structured around a workgroup LAN, while the more pervasive enterprise applications might be implemented on DCE because of its advantages of location transparency and platform transparency.

Although the purpose of these tests was to evaluate DCE performance, other OS/2 and AIX products were running and in use during the tests. Our test environment was not the typical benchmarking setup but more representative of a real-world environment. IBM LAN Server 3.0 provided file-sharing of the benchmark programs and exchange of performance data. OS/2 Extended Services 1.0 Database Manager was started on both the OS/2 clients and the OS/2 application servers. The Network File System (NFS) client and NFS server were started on both the OS/2 and the AIX servers. A variety of performance monitors were running on the OS/2 clients and the OS/2 servers. The IBM DCE/6000 Distributed File System server was running on the AIX server.

**The Point-of-Sale Benchmark**

In our Point-of-Sale benchmark, each PoS customer sale consists of the RPC transactions and application programming interface (API) calls listed in Figure 2.

In the Distributed Computing Environment, a *server* is a software entity (such as a daemon), rather than hardware. Multiple instances of a server can be on one computer system. In our PoS example,

each instance of a PoS server exports its unique identity to DCE Cell Directory Services (CDS).

Every customer sale begins with the client establishing a new RPC connection with a PoS application server by using binding information imported from CDS. When more than one server has exported an instance of the same application to CDS, a client's request to import binding information is randomly selected from the list of exported servers. This random selection automatically distributes the workload as additional servers are taken offline or brought online to balance peak-hour loads.

The DCE Security authentication level of the new binding is then set to the call level, meaning that the client's security credentials are reauthenticated by the DCE Security server for every DCE RPC transaction.

When each sale is completed, the binding to the PoS application server is terminated.

More information about the PoS benchmark can be found in the article titled "IBM DCE for OS/2 Multiuser Application Performance" in the January/February 1994 issue of *Personal Systems* magazine.

### Server Topology Studies

The performance studies in this article are grouped according to the server topology models shown in Figure 1. We present results beginning with simple environments and progressing to more complex environments.

Here are some of the terms used in these studies:

- **Throughput** is expressed as Customer Sales Per Minute (CSPM). Each customer sale consists of all the PoS transactions and RPC API calls listed in Figure 2.
- **Simulated workload.** The PoS workload is based on the assumption that one terminal will process one complete customer sale per minute.
- **Arrival rate.** The exponentially distributed think time between PoS RPC calls was progressively decreased in order to increase the rate of arrival at

the application server in increments of 30 CSPM. As one or more system resources became bottlenecks, the clients continued to try to achieve the specified throughput until the queuing reached a maximum depth.

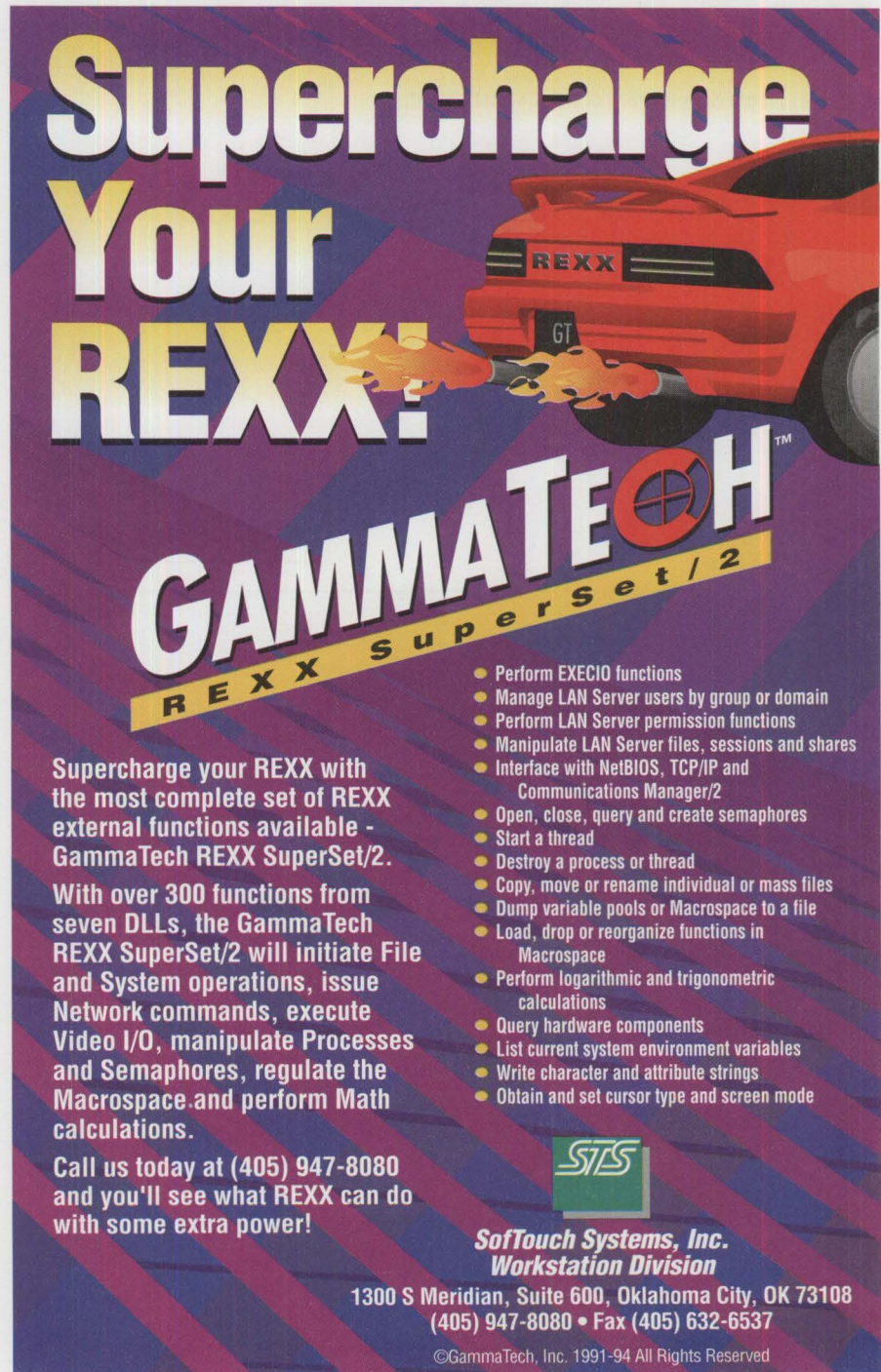
### Single-Server Environment

In the simplest environment, a single server provides all the DCE and application needs of an entire DCE cell. Three tests were performed for the single-server cell, beginning with a single PS/2 Model 95 with one hard disk, then

adding a second hard disk, and finally moving all the DCE server and PoS application server functions to the RISC System/6000 server. The results of these tests are shown in Figure 3.

The PoS application depends heavily on the performance of the physical and logical configurations of the file system and hard disks.

Our first two tests compared the throughput of a single server having only one hard disk to the throughput of a single



# Supercharge Your REXX!

## GAMMATECH™


### REXX SuperSet/2

- 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

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!



**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

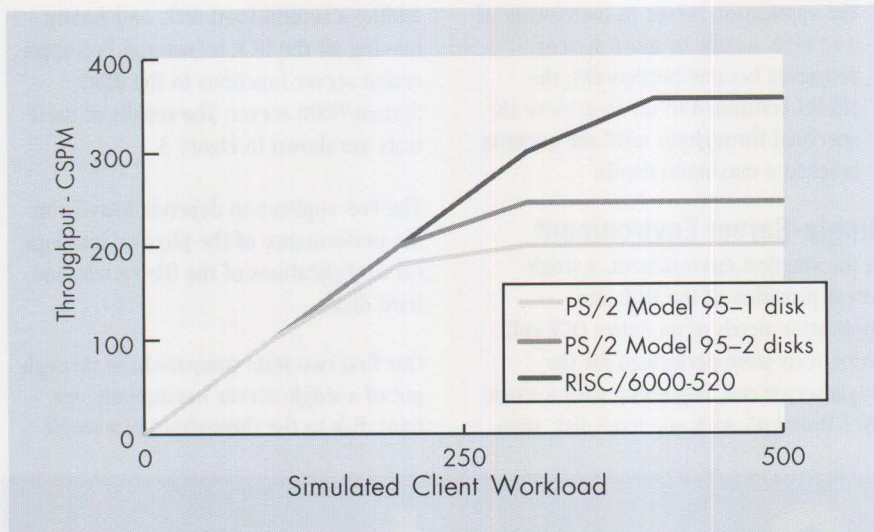


Figure 3. Single Server (PS/2 or RISC System/6000)

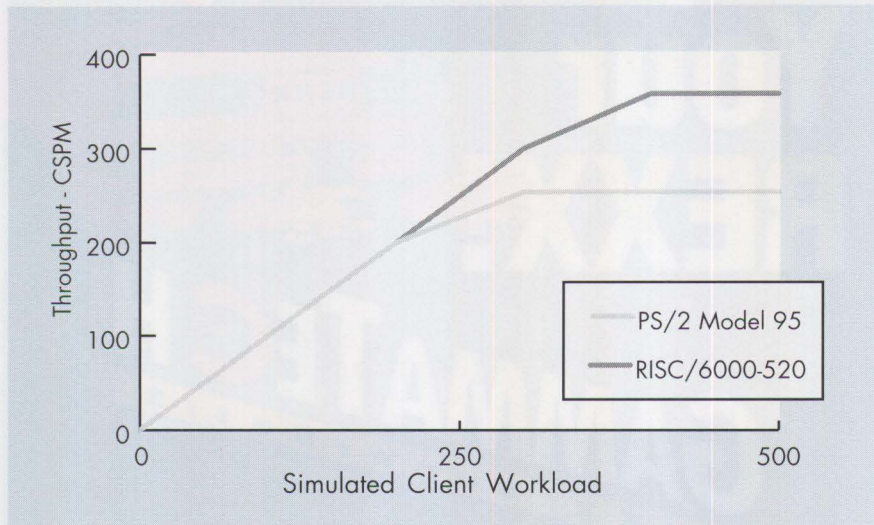


Figure 4. Comparing Two Application Servers (PS/2, RISC System/6000)

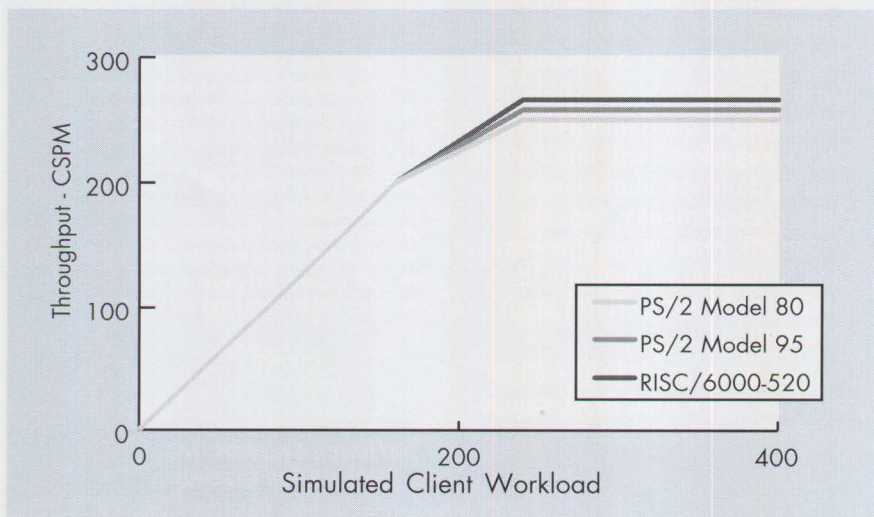


Figure 5. Comparing Three DCE Servers (PS/2 Model 80, PS/2 Model 95, RISC System/6000)

server having two hard disks. Figure 3 shows that the throughput of the two-disk server was 246 CSPM, which is 20 percent faster than the one-disk server's throughput of 200 CSPM. OS/2 is able to take advantage of the mechanical time that the hard disk uses to move the read/write heads and rotate the media. The greatest gain in performance occurred when going from one to two disks. Any change in performance for three or more disks is very small. Although a disk array is viewed as one logical drive, it exhibits the same characteristics as multiple physical disks.

The third single-server test compared DCE for OS/2 on a PS/2 Model 8595-0KD to DCE/6000 on a RISC System/6000 Model 520. Figure 3 shows that the RISC single-server throughput of 356 CSPM was 45 percent faster than that of the PS/2 single server at 246 CSPM. The large gain in throughput achieved by the AIX server can be attributed more to the PoS application than to the DCE core services being on the faster hardware platform. The speed of the DCE server turned out to be less important than the speed of the application server in improving overall throughput. (See the next section, "Two-Server Environment," for more explanation.)

We concluded from studying the single-server environment that a single server having both DCE services and DCE applications could be a viable configuration in a lightly loaded cell of 200 to 300 DCE clients. As mentioned above, these tests were conducted on entry-level servers; faster hardware for either the PS/2 server or the RISC System/6000 server would increase the cell throughput capacity.

## Two-Server Environment

The next logical step in expanding the enterprise environment was to add another server. In these tests, one server was the dedicated DCE server, and the second was the dedicated PoS application server.

In this configuration, two groups of tests were interesting. The first set of tests compared the PS/2 Model 95 to the RISC System/6000 as the PoS application server while keeping the DCE server constant on a PS/2 Model 95. Conversely, the second set of tests kept the PoS application server constant on a PS/2 Model 95 while changing the DCE server among



(1) a PS/2 Model 8580-A31 25 MHz, (2) a PS/2 Model 8595-0KD 33 MHz, and (3) a RISC System/6000 Model 520. The expectation in these tests was that the overall throughput would be influenced more by the speed of the PoS application server than by the speed of the DCE server.

Figure 4 shows the results of switching application servers, while Figure 5 shows the results of switching DCE servers.

In Figure 4, the PoS application is moved between a PS/2 Model 95 server and a RISC System/6000 server. The PS/2 Model 95's throughput was 254 CSPM, while the RISC System/6000's throughput was 29 percent greater at 358 CSPM.

Figure 5 illustrates the performance of three different DCE servers that provide the Security, Directory, and Time services. These tests were intended to demonstrate the sensitivity of the PoS application's performance to the speed of the DCE server. However, the results showed almost no difference in performance: 250 CSPM on the PS/2 Model 80, 258 CSPM on the PS/2 Model 95 (5 percent faster than the PS/2 Model 80) and 266 CSPM on the RISC System/6000-520 (5 percent faster than the PS/2 Model 95).

The speed of the PoS application server had much more significant effect on the application's performance than the speed of the DCE server. Therefore, when considering hardware for the DCE server and the PoS application server, keep in mind that the throughput changes much more when switching application servers than DCE servers.

In performing these tests, we noticed that the PS/2 Model 80 DCE server was pushing its performance and resource boundaries, with its CPU and RAM utilization approaching 100 percent. The PoS application was run at DCE Security call-level authentication. At higher authentication levels, the performance of the PS/2 Model 80 would degrade more rapidly than the performance of the PS/2 Model 95 or RISC System/6000 DCE servers.

### Multiple Application Servers

The next series of tests was intended to begin pushing the limits of the system. In the real world, many different

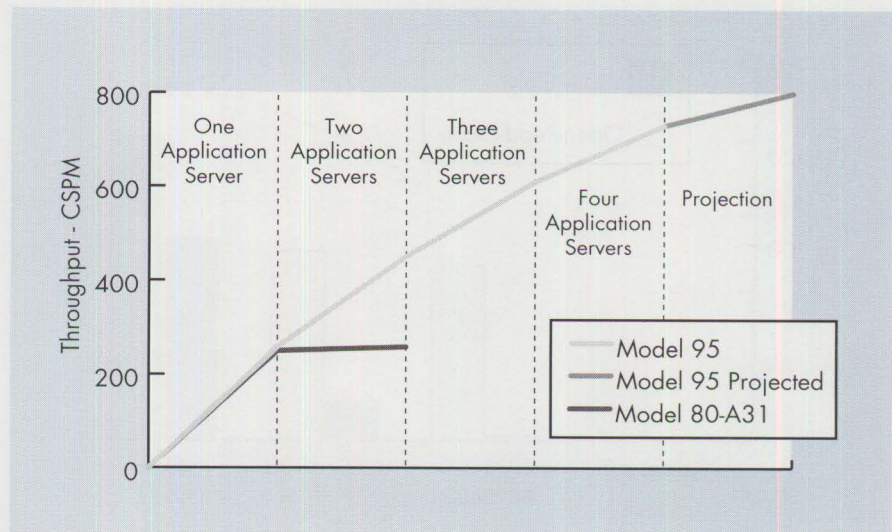


Figure 6. PS/2 Model 95 DCE Server with Multiple Application Servers

| File Name | Entities Added | File Size Growth Delta |
|-----------|----------------|------------------------|
| person    | 1,000          | 320,000 bytes          |
| org       | 1              | 4,740 bytes            |
| group     | 2              | 4,748 bytes            |
| acct      | 1,000          | 236,000 bytes          |
| acl       | -              | 400,748 bytes          |

acl = Access Control List

Figure 7. DCE Security Registry Database Directory File Growth

applications may be running on several servers at the same time. Because different applications vary greatly in their thirst for resources, it would be difficult to predict precisely the resource demands made by a specific set of applications.

To simulate this real-world load, three more PS/2 Model 95 PoS application servers were added to the cell to act as multiple unique applications. In theory, the throughput should scale up geometrically to 250, 500, 750, and so on as each additional application server is added, until the capacity of the DCE server is reached and throughput begins to flatten. In reality, we expected less than a geometric improvement because our current client hardware population is not large enough and fast enough to fully load the system at the higher arrival rates required for this test.

Several changes were made to the multiple application server tests to further increase the throughput load on the DCE server:

- The DCE Security registry was populated with 1,000 additional principals and accounts before running these tests. We expected that the additional DCE Security registry entities would affect RAM, DASD, and performance.
- The screen I/O on the PoS client benchmark was turned off, and the PoS client was started in two or more OS/2 sessions on each physical client. Higher-horsepower PS/2 486 clients were also added to the cell to achieve the desired load on the DCE server.
- Three additional PS/2 Model 95 PoS application servers were added to the cell. We expected that, as each additional PoS application server was brought online, the throughput should increase geometrically.

Figure 6 shows the results of these tests. In Figure 6, the PS/2 Model 95 DCE server remains constant while application servers are added one at a time. Because the results in Figure 5 for the OS/2 DCE

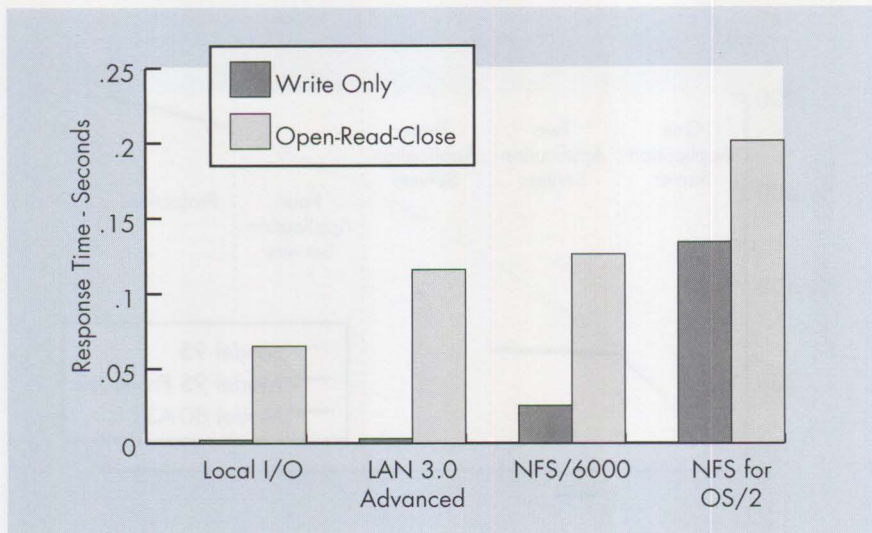


Figure 8. Remote File Systems Single-Client Response Times

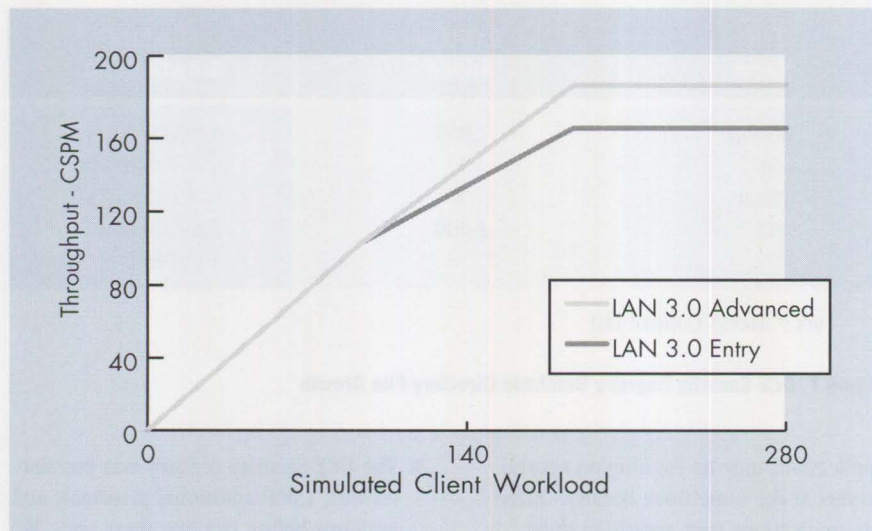


Figure 9. File Redirection Using IBM LAN Server 3.0

server and for the AIX DCE server were almost identical, in Figure 6 we show only the OS/2 DCE server. However, we did run tests using the AIX DCE server and found that, at all arrival rates, the throughput ratio between the OS/2 and AIX DCE servers remained consistent with the results in Figure 5.

Looking at the slope of the curve in Figure 6, we concluded that the DCE server was not yet working hard when there were four application servers processing a simulated workload of 730 clients.

The CPU utilization on the OS/2 DCE server was about 85 percent during the four-server test. Based on CPU characteristics in the tests with the PS/2 Model 80 DCE server, the throughput of the PS/2

Model 95 DCE server can be projected to flatten somewhere near 900 CSPM.

Using the hardware available in our lab, we were unable to generate enough workload to saturate either the PS/2 Model 95 or the RISC System/6000 DCE servers.

The PS/2 Model 80 may not be the best choice for the primary DCE server in a large multiple-server environment. However, price/performance considerations might lead to breaking up the DCE core services (Security, Time, and Directory) among three smaller servers.

In these tests, the CDS Server daemon has the highest CPU consumption. In an initial look at separating the CDS, Security, and

Time servers, we were able to drive from 70 to 90 percent more workload—almost 500 CSPM—through three PS/2 Model 80 DCE servers.

The PS/2 Model 95 and RISC System/6000 Model 520 both provide viable DCE core services performance for the multiple application DCE environment. Using the faster hardware available on both platforms would certainly improve throughput.

### Scaleup Tests

Before running the Multiple Application Server tests, we increased the DCE Security registry population by 1,000 principals, 1,000 accounts, two groups, and one organization. We observed the RAM and DASD usage for the registry while the 1,000 accounts were added. We observed no measurable growth in the memory working set due to the additional security registry population. The DASD required for the registry database grew by about 956 bytes for each account added, or a total of less than 1 MB for 1,000 accounts. Figure 7 shows the details of the file growth in the registry directory (d:\opt\dcelocal\var\security\rgy\_data).

The next test was to log in 360 unique accounts. We accomplished this by opening 15 OS/2 sessions on each of 24 DCE clients and executing a DCELOGIN with a unique account name in each OS/2 session. We observed no DASD growth. The memory working set on the DCE server grew by less than 1 KB per account logged in. This would result in less than 1 MB of RAM per 1,000 accounts concurrently logged in to DCE security.

Some performance impacts from the higher population of the DCE security registry were:

- The PoS application throughput for the single application server case degraded by up to 3 percent at higher arrival rates. DCE security authentication levels higher than call level should be expected to degrade as well.
- The average DCELOGIN response time was not measurably affected for 1, 24, or 50 simultaneous DCELOGINS.
- The average response time for 24 clients to simultaneously execute a first-time

rpc\_ns\_binding\_import\_next, where the client was forced to refresh its local CDS cache, was not measurably affected. The average time that the client took to execute the seven APIs in the PoS scenario to acquire a new binding remained about 0.7 seconds.

We concluded that scaling up the DCE Security registry has a tolerable impact on RAM, DASD, and performance.

### File Distribution Options

Several products available for the OS/2 and AIX platforms make it possible to share network files and data repositories. These products are useful in the DCE environment to balance disk capacity and to allow multiple instances of an application to access a common set of the application's files and programs.

From a performance perspective, accessing network files using LAN Server 3.0 or NFS typically adds path-length to the I/O functions, which tends to degrade response time. Disk I/O requests are somewhat serialized by the OS/2 queuing mechanism and device drivers. One conclusion that might be drawn from the one- and two-disk tests in Figure 3 is that the efficiency of OS/2 disk I/O is about 1.2 to 1. Therefore, even though two I/O requests may be serviced at the same time, the I/O throughput is less than 2 to 1. Therefore, disk I/O represents a bottleneck because fewer than two concurrent I/O requests can flow through it.

Adding the extra response time for remote file access into the I/O queuing algorithm would be expected to degrade both throughput and response time. In the next section, "Distributing the DCE Application," we explain how the nested DCE Remote Procedure Call can be used to increase, rather than degrade, system throughput while accessing remote files.

The PS/2 Model 95 application server was the file system client for each of the following file-sharing options tested:

- IBM LAN Server 3.0 is installable with two High-Performance File System (HPFS) options. The Advanced LAN Server HPFS386 option runs at a privileged OS/2 level and is tuned for remote file-sharing. The Entry LAN Server option runs at a more general OS/2 privilege level and uses the

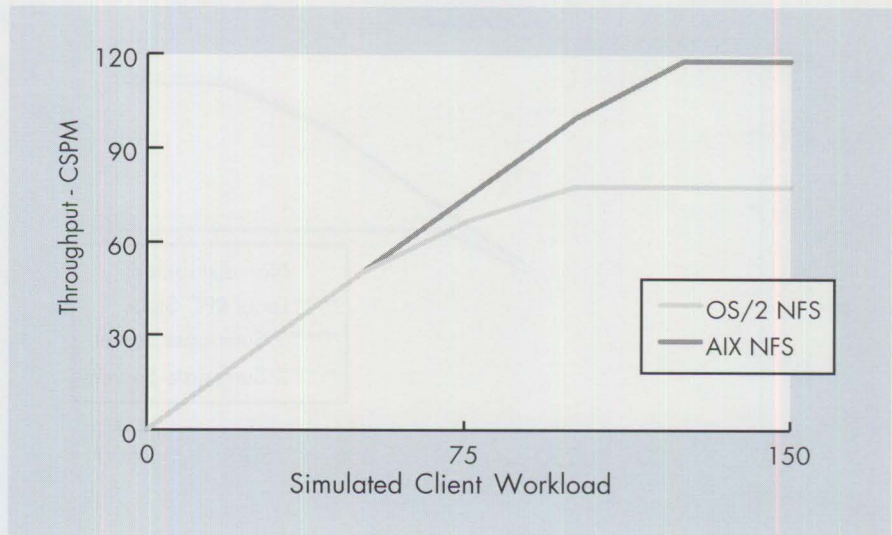


Figure 10. File Redirection Using the Network File System

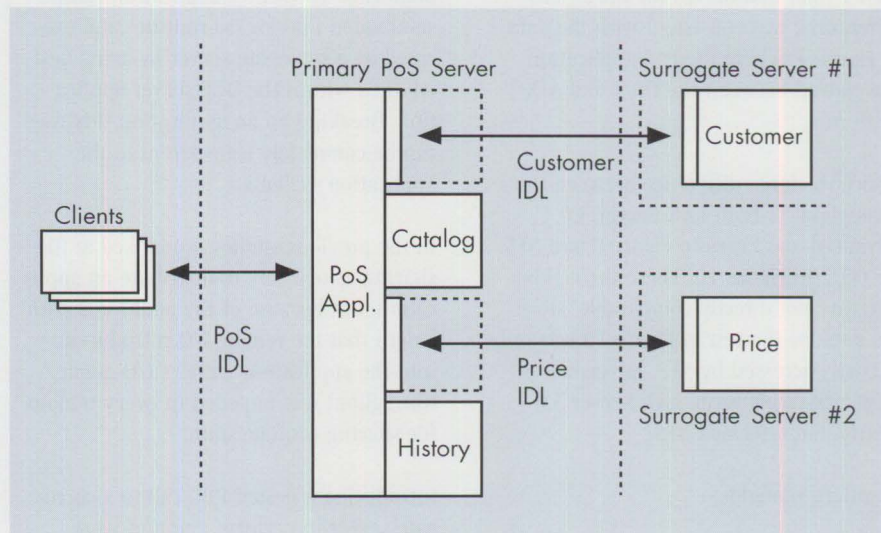


Figure 11. Distributing the PoS Application by Using Nested RPC Calls

default OS/2 I/O device drivers. The Entry LAN Server typically provides better I/O performance for local file accesses.

- The Network File System is available for both OS/2 and AIX and is an apples-to-apples comparison across these two platforms.
- Local I/O is the normal PoS operating mode and is shown as the point of reference for these comparisons.

Figure 8 illustrates the remote file system response times for two of the I/O transactions within the PoS application. The open-read-close (PoS Catalog) transaction must locate one of 5,000 unique image files stored in five subdirectories, open the file, read 16 KB into the application's

buffer, and close the file. The write-only (PoS History) transaction must temporarily lock the previously opened history file and write a 284-byte history record.

Figure 9 shows the PoS throughput with all of the PoS data repositories redirected across the OS/2 network using LAN Server 3.0 Advanced (HPFS386) and LAN Server 3.0 Entry (HPFS). Clearly, on the OS/2 platform, the HPFS386 server provides higher remote I/O throughput than the HPFS server.

The NFS servers for OS/2 and AIX are compared in Figure 10. In this test, only the PoS Catalog (open-read-close) and PoS History (write-only) data were accessed across the network. The throughput of NFS/6000 was about 30 percent

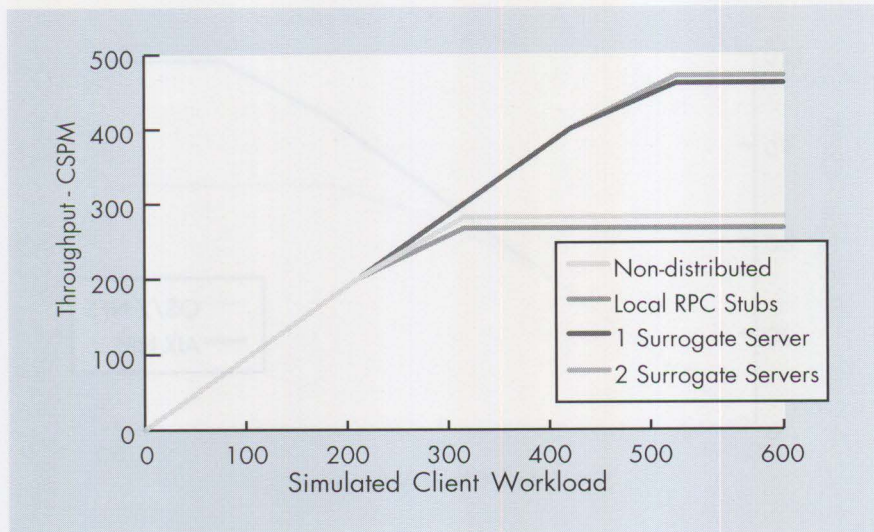


Figure 12. Distributed Performance Using Nested RPC Calls

higher than that of NFS for OS/2. This differential was consistent with the data in Figure 4, where the PoS application was moved between the OS/2 and AIX platforms.

Notice the large difference between the throughput of both LAN Server 3.0 Advanced and Entry in Figure 9 and NFS for OS/2 in Figure 10. These throughput rates are not directly comparable, since the data for the test in Figure 10 was only partially accessed by NFS. Nevertheless, on the OS/2 platform, LAN Server 3.0 clearly outperformed NFS.

Corollary thoughts:

- Currently, NFS is the only off-the-shelf IBM file-sharing offering between OS/2 and AIX. A home-grown RPC or sockets solution might be considered.
- LAN Server 3.0 Entry, rather than LAN Server 3.0 Advanced, can provide slightly better local I/O performance.

### Distributing the DCE Application

How do we break the PoS application server I/O bottleneck? Many applications depend on large data repositories, and like PoS, those applications encounter a performance bottleneck in the disk I/O subsystems. Other applications are CPU-intensive, and they find that their bottleneck is the processor in the application server.

The DCE Remote Procedure Call is a powerful tool for cooperatively distributing work among multiple servers and across

platforms. The server application can be subdivided into logical routines and executed by a surrogate server by using nested RPCs within the DCE server application. Breaking up an application this way can be completely transparent to the application's clients.

In the previous studies, we looked at file-sharing as one way to distribute an application. But because of the additional path-length that the remote I/O introduced into the application server's I/O queue, throughput was impeded in every remote file-sharing configuration.

Introducing a nested RPC call to a surrogate server to perform a unit of work allows the calling thread to go to sleep and wait for the nested RPC call to return. In effect, this moves the queuing to a non-obstructive state in the operating system queue. While the thread waits for the surrogate call to return, other local I/O and CPU processing can continue unblocked.

A secondary benefit of this method is that it enables multiple instances of an application to share a single copy of a data repository in a safe and controlled manner.

To demonstrate the performance characteristics of nested RPCs, we chose two of the PoS transactions as candidates to distribute to a surrogate server. The PoS Price and PoS Customer transactions each contain a few hundred lines of C-language code, and they perform a three-level index search and two or three disk reads. The RPC server stub of the PoS

application was modified to invoke the nested RPCs and to simply pass input and output data between the client and the surrogate servers.

Figure 11 illustrates how the PoS Customer and the PoS Price routines were moved to surrogate servers. A new RPC Interface Definition Language (IDL) was created for each new surrogate routine. The PoS IDL presented to the client was not changed.

The expectations for this experiment were that (1) the response times for these two transactions would increase by up to 500 milliseconds due to the additional RPC call, and (2) the maximum throughput for the application server would improve.

The throughput improvement shown in Figure 12 exceeded our expectations. The I/O bottleneck in the PoS application server was relieved, permitting the CPU to approach full utilization. Distributing the PoS application in this manner enabled cooperative processing between multiple servers and resulted in a 54 percent increase in throughput. The increase in response time for the two transactions was about 460 milliseconds per surrogate RPC. The aggregate end-user response time for a complete customer sale having 5.5 RPC transactions increased from 4.6 to 6.2 seconds at the higher arrival rates.

Figure 12 shows three permutations of our distributed application experiment:

1. Two surrogate servers (for PoS Price and PoS Customer) on separate physical servers
2. One surrogate server (again for PoS Price and PoS Customer) on one physical server in separate OS/2 sessions
3. The primary PoS server and the two surrogate RPC routines all running on one physical server in three OS/2 sessions

The intent of the third permutation was to evaluate the additional performance overhead of calling RPC routines locally.

Figure 12 also includes the non-distributed PoS test results as a basis for comparison.

The results of the local RPC test (permutation 3 above) surprised us. We expected

the throughput with the primary PoS server and the two surrogate routines all running on the same physical server to be substantially lower than the throughput of the non-distributed version. The PoS benchmark is I/O-bound on a single application server. Although the CPU utilization was much higher during this test, the bottleneck was still the I/O. A CPU-intensive application would indeed have resulted in much lower throughput in this configuration. Based on this test, you might consider initially creating an application with nested RPCs, even though they might not be physically moved to a surrogate server until a later time when system capacity warrants the additional hardware.

From this experiment, we concluded that:

- Exploitation of nested RPC calls to balance performance and resources has proven to be a powerful system-tuning tool.
- The location and platform transparency of DCE offers a wide variety of cooperative processing options to distribute an application.
- Surrogate RPC routines allow multiple instances of an application to share common data repositories under the security and control of DCE.

## OS/2-to-AIX Porting Considerations

The PoS benchmark was initially developed for the OS/2 DCE platform with only a fleeting thought about the differences to be encountered in AIX/6000 and DCE/6000 on the IBM RISC System/6000. Fortunately, the pitfalls were few, self-inflicted, and easily remedied:

- Byte reversal. Numeric data is stored differently on the two platforms. The first OS/2 version of PoS was written to pass some of its structures in the RPC Interface Definition Language as an array of bytes rather than a structure. This worked well for OS/2-to-OS/2 and AIX-to-AIX, but the first time we tried OS/2-to-AIX, the numeric values were incorrect. The remedy was simple: incorporate the structure definition in the IDL. RPC automatically handles the numeric byte reversal correctly.
- IBM OS/2 Toolkit. The Toolkit provides many handy interfaces into OS/2 functions. Many of the Toolkit interfaces

can be accomplished using IBM C Set/2 library calls, which require little or no change between OS/2 and AIX. When ease of portability is a concern, operating system-specific APIs should be avoided.

After completing our port of PoS from DCE for OS/2 to DCE/6000, the American National Standards Institute (ANSI)-compliant source code can now be compiled on either platform without code changes.

## Principal Results

- A single server providing both DCE services and application support is a viable entry point (Figures 3, 4, 5). Moving the PoS applications to another server as the workload increases is a simple procedure that is transparent to the clients and requires no application or configuration changes.
- When a large DCE cell environment is planned, initially configuring three separate DCE servers (one each for Security, Directory, and Time) will prevent having to reconfigure all the DCE clients if these services are later broken up. Initial measurements with dedicated Directory, Security, and Time servers indicate much better throughput at high concurrency workloads.

■ Several off-the-shelf file-sharing options are available for the OS/2 and AIX environments. We recommend the following based on the options we have tested (Figures 8, 9, 10):

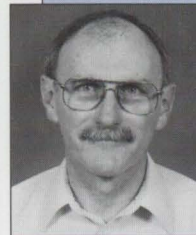
- In the OS/2-to-OS/2 environment, IBM LAN Server 3.0 Advanced offers better performance than NFS for OS/2.
- Currently, NFS for OS/2 and NFS/6000 for AIX are the only cross-platform file-sharing packages we have evaluated.
- Using nested RPCs to cooperatively distribute the application workload among multiple servers was the highlight of these studies. Allowing the primary application server to offload I/O- or CPU-intensive routines to surrogate servers resulted in a 54 percent throughput improvement in our implementation (Figure 12).

## Acknowledgments

We acknowledge the contributions of Tom Boes for porting the PoS application to the AIX platform and for his assistance in setting up and running the many cross-platform tests reflected in this article, and Laura Adams for her help in putting together this article.



**Benetta N. Perry** is a Senior Associate Programmer. In her 10 years of employment with IBM, she has assured for quality, tested for functional accuracy, and analyzed performance, RAM usage, and DASD usage of the OS/2 Operating System, OS/2 Communications Manager, OS/2 Database Manager, OS/2 Query Manager, and Distributed Computing Environment for OS/2. Benetta has a BS degree in computer science from Grambling University.



**Bob Russell** is an Advisory Programmer and team leader of the Workgroup Environment Team in LAN Systems Performance. Since joining IBM in 1963, Bob's

assignments have included Electric Typewriter Division Customer Engineer; Office Products Division Administrator; Systems Analyst for the Installed Machine Inventory System; and Performance Analyst for DisplayWrite, OS/2 Database Manager, and DCE for OS/2.

Perry and Russell can be contacted via voice mail at (512) 838-0844 or fax (512) 838-0666. You can also direct your comments and questions about this article to [psts@vnet.ibm.com](mailto:psts@vnet.ibm.com).

# Architecture Soup: Understanding Modern IBM PC Architecture

*“Alphabet soup” is perhaps a fitting metaphor for the acronym puzzle that confronts the computer user when selecting the proper system. The computer designer, too, now has hundreds of standards from which to select the optimal recipe to satisfy the user’s application needs. No single recipe is proper for all users or all applications.*

*The purpose of this article is to define the proper combination of ingredients. The article focuses on local buses, I/O buses, file interfaces, and parity in RAM. It also recommends components for the various price/performance design points.*

To arrive where PC architecture is today, let us first review the various PC architectures since IBM introduced the original Personal Computer in 1981.

The first PC had one bus and a few peripheral interfaces, as shown in Figure 1. The PC’s memory shared the same bus as the display and all the input/output (I/O) cards. Everything in this simple system was “local” to the processor.

---

**Chet Heath**  
IBM Corporation  
Boca Raton, Florida

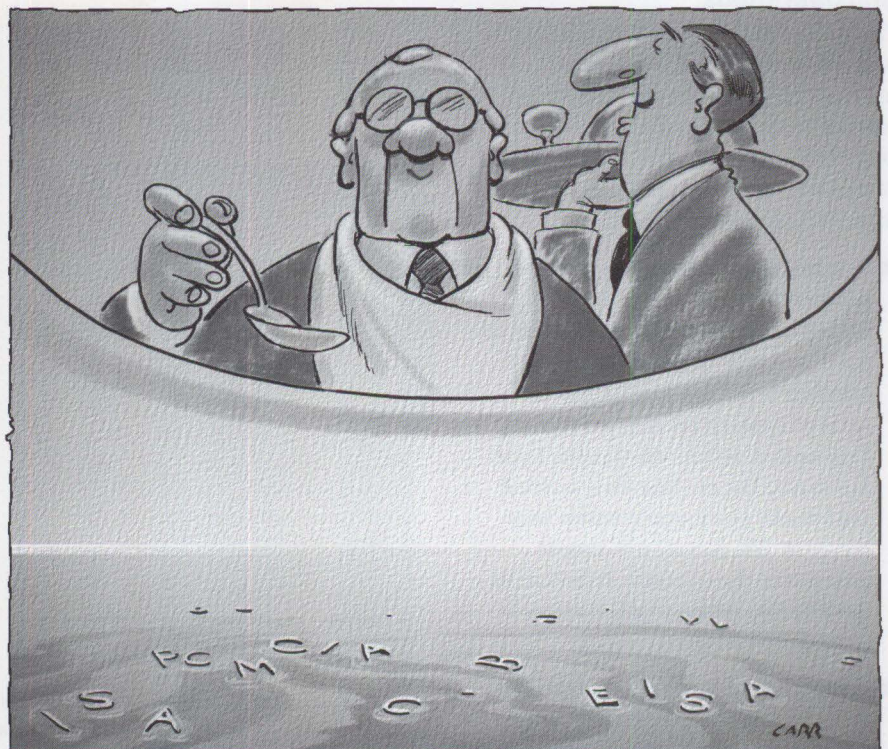
---

Even the first Personal Computer AT systems were designed with just one bus for everything. However, as PC processors became faster and wider than the AT’s 16-bit I/O bus, memory was moved to a local bus (see Figure 2) to allow maximum throughput for instruction fetch.

To save money with large memory systems, a cache of fast RAM could contain the most frequently used instructions, with slower, less expensive RAM holding the remaining, less frequently used instructions.

As the difference in speed and width between the processor’s local memory bus and the I/O bus increased, the processor and memory would wait for the devices on the I/O bus to respond. In single-tasking systems, the mechanical and external interface delays of most devices masked any delays in processing.

Where efficient data-transfer was required, Micro Channel’s streaming data



transfer cycles could move data blocks rapidly to memory, with delays only in the first transfer of a long block. However, delays were noticeable in display operation because the wait periods, or wait states, occurred with every transfer. Moving the display adapter closer to the processor—onto the processor's local bus as depicted in Figure 3—solved this problem. The local bus treats the display adapter as though it is system memory.

### Standardized Local Bus

Local buses come in three basic flavors: internal, VESA-VL, and PCI. These are the basic choices in most PC systems today.

An internal local bus display adapter is typically custom designed for each system. Although a custom design is usually disadvantageous, it has the advantage of being optimized for a given platform.

VESA-VL is the product of the Video Electronics Standards Association. It combines the industry standard architecture (ISA) I/O bus and the 80486 processor bus operating at up to 40 MHz. VL stands for Video Local bus; it was the first to define a standardized socket to insert standardized display adapter cards. VL is defined primarily for display functions, although it has been used for other purposes such as attaching fast file adapters.

Peripheral Component Interconnect (PCI) is a specification initially defined by Intel and previously called the Local Glueless Bus, or LGB (but LGB turned out to be the trademark of a German toy company). Although PCI was originally proposed as an internal local bus, a standardized connector was defined, allowing it to attach a wider variety of system adapters.

Both VESA-VL and PCI have industry organizations of participating manufacturers. VL is well established, but PCI has its strongest support among the major system developers. It is not clear that either definition will dominate the industry; rather, combinations of various local bus and I/O bus standards will likely prevail for several years. The need for performance will also ensure a place for internal local bus design.

### VESA-VL Bus

The VESA-VL bus architecture is shown in Figure 4. The VL connector contains both

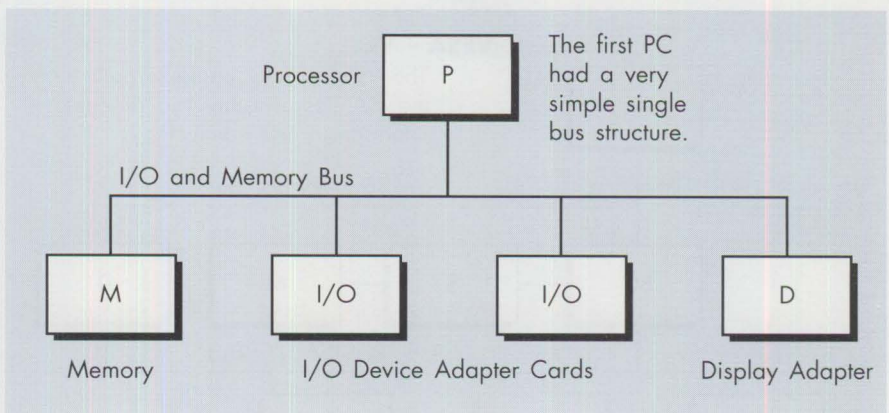


Figure 1. Simple Architecture of the First PCs

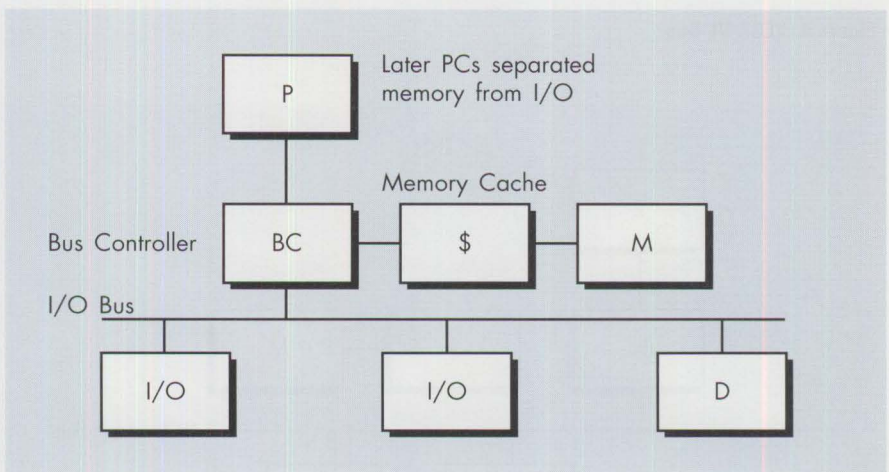


Figure 2. Memory on a Local Bus

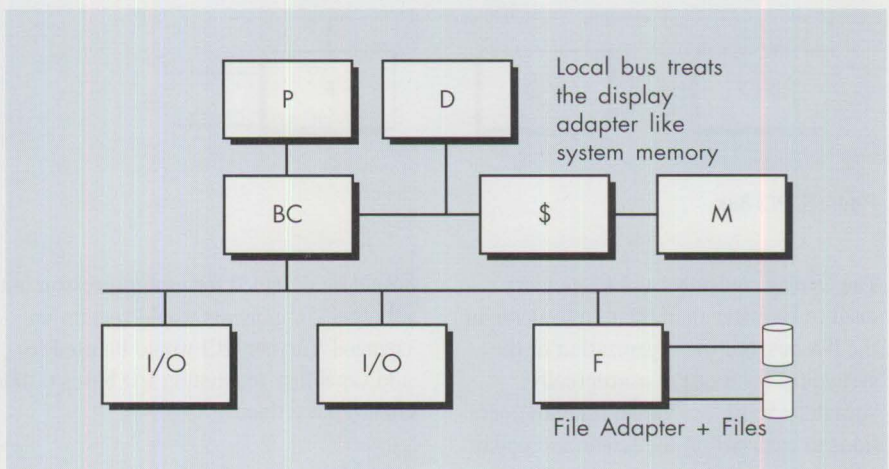


Figure 3. Display Adapter on a Local Bus

the ISA bus and the 486 processor interface in a single connection.

In 486 systems where the ISA bus is already present, VL is very inexpensive to add. The 486 processor bus, plus a few more control signals, is added to one or

two ISA connectors. The display card then plugs into the connector(s), where it receives slow and infrequent control information from the ISA bus, as well as fast and frequent display data information directly from the processor.

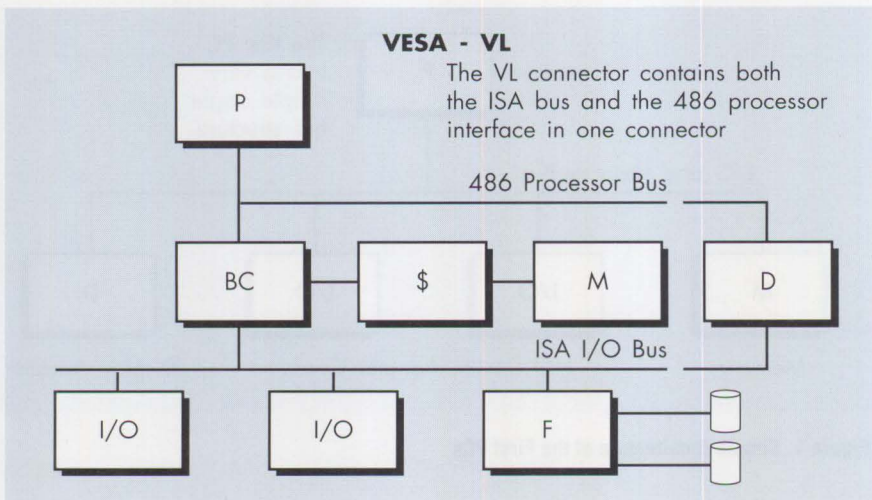


Figure 4. VESA-VL Bus

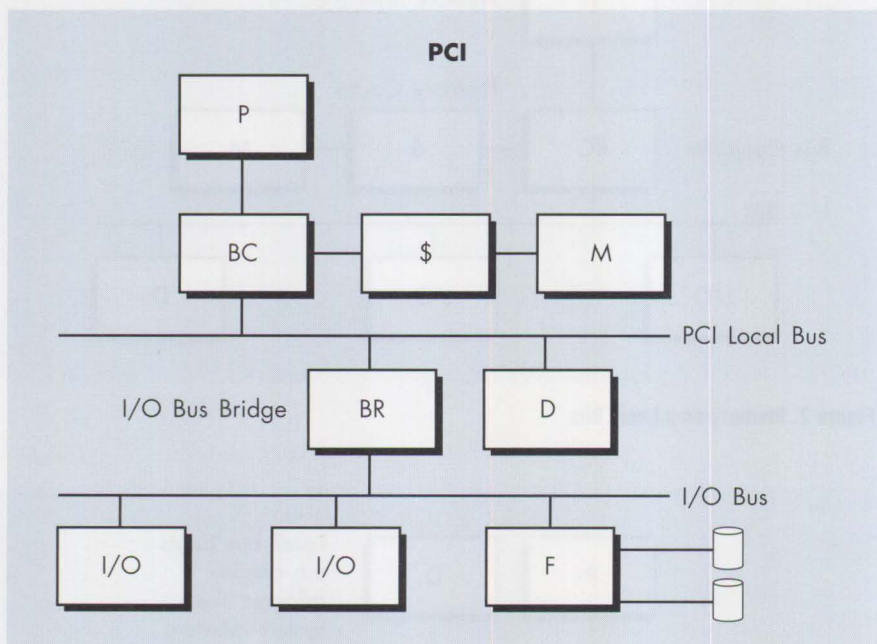


Figure 5. PCI Bus

The display controller and its memory need to be faster than their equivalents in the ISA bus, but the organization of the system (in Figure 4) is conceptually straightforward. Individual random operations to each pel on the screen are updated in random fashion, without the wait states of a bus-attached display.

### PCI Bus

PCI, shown in Figure 5, was defined to be a standardizing influence on PC system design. Intel saw PCI as a means of avoiding expensive repetitive redesigning of processor support components that came with each new generation of processors. One component, the PCI bus controller,

would be designed for each processor, but all other components would remain unchanged. Further, PCI would be used to add capability to existing I/O buses rather than replace them.

In truth, however, PCI by itself cannot provide a complete solution. PCI typically needs a traditional I/O bus such as ISA, EISA, or Micro Channel to produce a PC- and DOS-compatible design.

PCI does not define Direct Memory Access (DMA), upon which almost all systems depend to attach the diskette adapter, common sound boards, tape backup, LAN, and even hard-disk

interfaces, nor does it define the 11 PC AT interrupt request signals critical to PC compatibility.

Many DOS applications access the interrupt controller directly, in response to adapter card requests for attention. These applications expect the requests to be associated with specific request lines into the system. In turn, these request lines are associated with specific register addresses in the PC—addresses where the processor can begin to process the requests. PCI does not provide this one-to-one correspondence between the interrupt request line and the starting address of interrupt service for these existing DOS applications.

Any PC I/O bus can be combined with PCI to complete the requirements for DOS compatibility. However, depending on the application, the choice of I/O bus can strengthen or weaken the suitability of the complete design to satisfy its intended applications and users. In non-DOS-compatible systems, PCI on its own can function as a fast interface to I/O. Therefore, PCI can augment the functionality of any I/O bus in PC DOS systems or act on its own where PC DOS compatibility is not required.

As a central element in a system, PCI can connect to traditional I/O buses through a “bus bridge” component. (I/O bus bridge components exist for ISA, EISA, Micro Channel, Personal Computer Memory Card International Association [PCMCIA], and Apple NuBus.) This component converts the PCI interface to the I/O bus interface, stores data temporarily in transit, and controls the I/O bus operations.

The bridge component permits the designer to define a common converged element of the basic PC processor and memory complex and to define modular personalization of I/O bus, file, LAN, print, and communication interfaces (much like the basic chassis of a car accepts many transmissions and drive-train components to define a complete line of automobiles).

PCI's disadvantage is its bridge component cost, which is not required in a 486-based VL design. Because of PCI's flexibility, the cost can be justified in most high-end implementations.



## Mother Nature Intervenes

Mother Nature limits the number of supported PCI (or VL) connectors to a practical limit of three in most systems, because the capacitance (the electrical analog of elasticity) combines among the components to limit the speed at which a bus can move data—the more components connected, the more limited the data transfer speed. While PCI defines 10 components that may be connected, these are components that can only be internally connected to the system board. Also, cards with connectors count as two loads each. With a PCI bus controller, a PCI I/O bus bridge, and local I/O, each consuming only one load, three card sockets can typically be defined. However, one socket usually houses the display adapter, so only two sockets remain available. For these reasons, the I/O bus must provide an adequate number of adapter slots as well.

In essence, then, the system designer has a choice: (1) more slots, more capacitance, and lower speed; or (2) fewer slots, less capacitance, and more speed. The system designer's selection of the I/O that combines with PCI (or VL) can create dramatic limitations on the system's ability to actually use PCI (or VL).

## Internal Local Bus

If the local bus is not delivered to a connector, the capacitance is minimized so the local bus' operating speed can be increased. If a display adapter is then designed to take advantage of the faster throughput, display responsiveness significantly improves.

For example, a Pentium processor has a 64-bit interface that operates today on a 66 MHz cycle. These numbers result in a data transfer rate of 528 MB per second (see Figure 6). This internal local bus design provides roughly four times the throughput of either VL or PCI, because it operates at twice the speed and twice the width of the typical 32-bit, 33 MHz local bus plus connectors.

To take advantage of the faster data rate, a display adapter with a 128-bit internal architecture and refresh memory interface might be required. This display could then use the direct 64-bit connection to the Pentium processor at four times the performance that one might expect from a

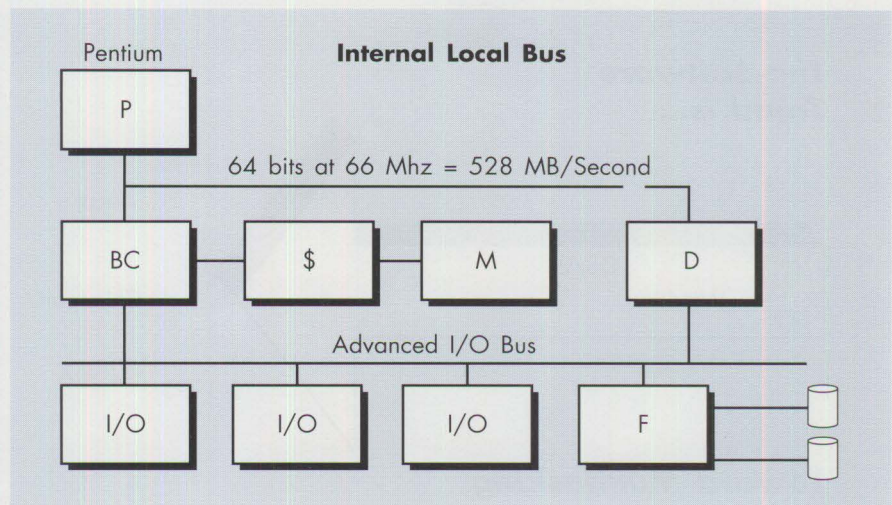


Figure 6. Internal Local Bus

standard 32-bit local bus. While resembling VL, internal local bus interfaces are wider and faster than VL. Today, such a system is likely to be limited to very high-end PCs, engineering workstations, and perhaps wealthy game enthusiasts, because its cost is easily comparable to the cost of the rest of the PC system.

The internal local bus structure would also require an advanced bus for connecting all I/O adapters. PC systems define all I/O devices except the display as logical files. This means that printers, communication interfaces, hard disks, LANs, and all I/O devices other than the display are logical files. In a logical file, all data is organized as a block of sequentially ordered data. An advanced bus, such as Micro Channel, can move block-oriented data with roughly the same performance as a local bus. A local bus is then required only for the display adapter, because it is the only adapter where data is transferred in a random fashion. Indeed, it may be desirable to place the file and display on separate buses to avoid contention, especially in higher video definition multimedia systems.

A modern PC design requires both a local bus, typically for video display performance, and an I/O bus for providing a larger number of DOS-compatible sockets. The local bus and the I/O bus are not comparable elements. *As in a good marriage, each party brings strengths and weaknesses, and when the two parties each bring strengths that compensate for the other's weaknesses, synergy results.*

## Video Media Extension Interfaces

The advent of multimedia is introducing buses that provide intercommunication between the auxiliary functions and the display controller. Examples are video decompression or compression, video capture, and conversion to broadcast-standard outputs. Figure 6 shows where the video media extension interface fits in the system.

There are two standards for this interface:

- VESA Advanced Feature Connector (VAFC). This standard is derived from the historical interface for video extension provided on VESA standard display adapter cards. It can support only one auxiliary device and it is not upgradeable.
- VESA Media Channel (VMC). This newer interface is a dedicated multimedia channel that will eventually replace VAFC. VMC has been accepted by VESA to support multiple auxiliary video functions. It is a more sophisticated architecture that supports higher speeds and concurrent video device support. Like the small computer system interface (SCSI) for file devices, it is modular; that is, it enables flexible expansion by the incremental addition of auxiliary video functions as the user's needs grow. VMC is also a comprehensive hardware and software specification.

## System Requirements Drive Decisions

Computer hardware can be thought of as simply a tool for carrying out the

## The Software Trend is...

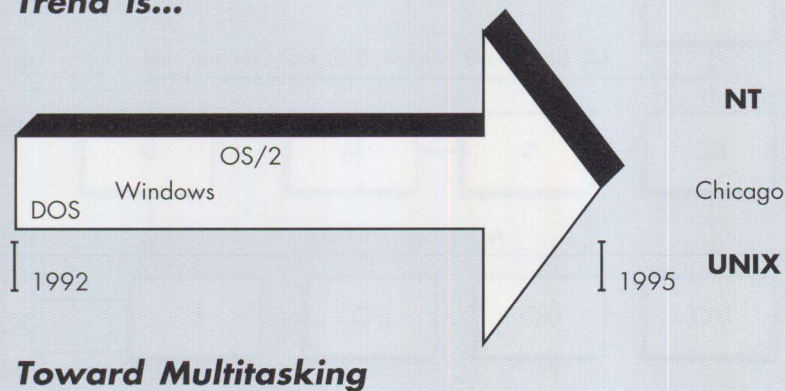


Figure 7. Software Trend Is Toward Multitasking

### Single-Tasking

- I/O is operated sequentially.
- Up to 100 percent processor or other resource capacity can be used by I/O.
- Programmed I/O adapters are economical and sufficient as long as they do not need more than 100 percent of any resource.
- Bus master and DMA slaves are only advantageous if they are more responsive.

### Multitasking

- I/O is operated concurrently.
- All resources, including the CPU, must be shared among tasks.
- Programmed I/O adapters are parasitic on the processor as a common resource.
- Bus masters and DMA slaves conserve processor bandwidth for multiple tasks.

Figure 8. Single-Tasking Versus Multitasking

instructions contained in operating systems and applications. When deciding on basic computer design, the designer looks first at the system application, including the operating system environment.

Within the next year, computer systems will be required to be optimized for multitasking for new operating systems. A computer system's suitability for multitasking will become a major issue as major operating systems convert to multitasking structures capable of operating multiple I/O devices concurrently (see Figure 7). At the same time, these systems must also be compatible with older PCs to run DOS/Windows legacy applications. PS/2 Micro Channel systems already have both of these attributes.

### Single-Tasking Systems

In the simple structure of DOS- and Windows-based systems, the I/O devices operate one at a time, in sequence. In this simple sequential environment, it is permissible to economize an adapter's design and to use up to 100 percent of the processor's capacity to control devices and move data.

When multiple I/O devices operate sequentially in single-tasking computers, the designer must define maximum bus throughput for the worst case, which is just the fastest device—typically the file interface.

### Multitasking Systems

In true multitasking operating systems (such as UNIX, Windows NT, and OS/2),

multiple I/O devices operate at the same time. This concurrent I/O operation yields a radically different set of system design requirements for local bus, I/O bus, file interface, adapters, and even diagnostics and setup.

If an adapter monopolizes the processor's attention, it adversely affects a multitasking system's ability to execute other tasks. Therefore, the designer's objective is to minimize an adapter's dependency on the processor. (This is precisely the opposite of the design expense objective for adapters in single-tasking systems.) All resources, including the processor, must be shared among multiple tasks.

Indeed, many attributes normally associated with premium desktop and server solutions actually came about because users required I/O devices to operate concurrently. This concurrent I/O device operation is usually coordinated by a multitasking operating system or by a network operating system on a server machine.

Figure 8 summarizes the major considerations when designing systems for single-tasking and multitasking.

### Design Requirements for Multitasking

When multiple devices operate concurrently, the worst-case throughput is the *sum* of the throughput of all I/O devices in the system. When all devices operate concurrently, the frequency of interrupts increases, and multiple interrupts can potentially occur simultaneously.

Because a multitasking system is more active, the probability of error is increased. Consequently, both the design requirement to catch errors and the requirement for diagnostic routines to catch error-causing failures increase. Finally, concurrent tasks (user applications) cannot simultaneously direct data to the same I/O device. Therefore, as more tasks run concurrently, more I/O devices are needed. For example, two printing tasks operating at the same instant require two printers, especially if together they generate more data than one printer can handle.

When more I/O devices are connected to handle multitasking and multimedia operations (which also perform concurrent

I/O), the system's setup and configuration become quite complicated.

Multitasking, multi-user, and multimedia are increasing requirements for higher-throughput I/O buses, improved interrupt systems on buses, and adapter design mechanisms (such as bus masters and DMA slaves) that offload the processor. Premium multitasking systems require more sophisticated error detection and recovery procedures than their single-tasking counterparts.

Perhaps most noticeable of all in multitasking systems is the increased difficulty of setting up the hardware and software drivers. To offer relief, systems are being designed to accommodate plug-and-play setup.

### Plug-and-Play

Plug-and-play significantly reduces the ordeal of system configuration. In just a few milliseconds, the PC automatically configures all the adapter hardware, selects the proper software drivers for the hardware, configures the software drivers, installs them, and arbitrates all the potential setup conflicts—all without human intervention. Plug-and-play eliminates CONFIG.SYS files, AUTOEXEC.BAT files, reference diskettes, and Control Panel setup. All the user has to do is plug in the card, turn on the power, and watch the system configure itself.

### Don't Forget Legacy Applications

While caught up in the euphoria of designing multitasking and multimedia systems, the designer must not forget the basic requirement to support legacy single-tasking applications, which constitute the majority of today's applications. Therefore, any new multitasking machine must be a good DOS/Windows system as well.

Figure 9 compares several I/O buses. Computer systems typically need one local bus and one I/O bus.

### System Requirements Based on Application

Consider these four types of computer systems:

- Type 1—Low-cost, primarily single-tasking

# What is Micro Channel Architecture?

Computers that concurrently support many users and tasks have been the core of IBM's system focus since the introduction of System/360 in 1964. But the phenomenal success of the single-tasking, single-user, DOS-based personal computer has created another culture that has focused solely on the simple DOS/Windows environment.

Quite simply, DOS- and DOS/Windows-driven PCs do only one thing at a time—each operation is performed in sequence. With few exceptions, each DOS operation can count on using the full capabilities of the processor, memory, and other support elements, because only one I/O device is operated at a time.

This requirement is simple and inexpensive to implement in a computer system, but it does not use time efficiently. The delays inherent in performing sequential steps extend the total time the computer needs to complete the task.

To become more productive, the computer should be designed to support multiple operations at one time, filling the delays in one operation with useful progress in another. This "concurrency" is fundamental in systems that support multiple tasks or multiple users.

When I/O operations are performed concurrently, the computer can no longer dedicate all of its resources to a single operation; therefore, sharing resources is a requirement. It is the responsibility of a channel architecture to carry out this resource sharing. At the same time, the channel must maximize efficiency and prevent the errors that may happen if equitable sharing does not occur.

Micro Channel Architecture, introduced with the advent of multitasking, multi-user operating environments, is simply concurrent mainframe architecture made compatible and cost-effective for microcomputer systems.

- Type 2—Single-tasking, with occasional multitasking
- Type 3—Single-tasking and multitasking, networked
- Type 4—Portable

IBM product lines correspond to these four system types: PS/1, ValuePoint, PS/2, and mobile, respectively. Each of these four system types, driven by application requirements, is discussed below.

#### Type 1—Low-Cost, Primarily Single-Tasking

This kind of system is optimized for low cost and single-tasking performance. It runs DOS/Windows regularly. (It can run OS/2 or another multitasking operating system with just one I/O device active.) Tasks are forced to operate I/O interfaces sequentially, but in this

system's single-task environment, sequential I/O is satisfactory and yields adequate performance. For this type of system, typical applications are word processing, modem/fax communication, spreadsheets, and the spectrum of general-purpose PC applications.

For cost purposes, ISA architecture is an adequate choice. Given the presence of ISA, a VL bus for the display adapter will provide performance equivalent to PCI, and a VL bus is cheaper because it does not implement the bridge component that PCI requires.

IDE files, which require sequentially operating I/O devices, are typically selected for their low cost per storage byte. IDE devices are usually attached to programmed I/O slave adapters that

| Bus Comparison   | ISA               | EISA                | Micro Channel            | PCMCIA               | PCI                   | VL                   |
|--|-------------------|---------------------|--------------------------|----------------------|-----------------------|----------------------|
| Throughput (width)<br>(@ speed)                                    | 16-bit<br>@ 8 MHz | 32-bit<br>@ 8.3 MHz | 16/32/64-bit<br>@ 20 MHz | 8/16-bit<br>@ 10 MHz | 32/64-bit<br>@ 33 MHz | 32-bit<br>@ 40 MHz   |
| MBytes per Second<br>Random<br>Block                               | 5.3<br>5.3        | 10.6 *<br>33 *      | 10/20<br>160             | < 10<br>< 10         | 132<br>264            | 80<br>80             |
| Automatic<br>Configuration   | ISA-PNP<br>Only   | Y/N *               | All                      | All                  | All                   | Potential<br>ISA-PNP |
| Percentage of<br>Portable Cards<br>with Plug-and-<br>Play Hardware | < 1%              | < 10%               | 100%                     | 100%                 | 100%                  | 0%                   |
| Multitasking Support<br>for Concurrent I/O                         | No                | Yes                 | Yes                      | No                   | Yes                   | No                   |
| Processor<br>Alternatives  | DMA               | Master/<br>DMA      | Master/<br>DMA           | None                 | Master                | DMA<br>(ISA)         |
| Bus Master Support   | 0 **              | 7 max               | 15 max                   | 0                    | 1/slot                | 0 **                 |
| DMA Slave Support  | Yes               | Yes                 | Yes                      | No                   | No                    | Yes                  |
| Number Defined   | 7                 | 7                   | 15 max                   | 0                    | 0                     | 7                    |
| Number 8237 /<br>DOS-Compatible                                    | 4                 | 7                   | 15 max                   | 0                    | 0                     | 4                    |
| Number 8237 /<br>DOS-Available                                     | 0                 | 7                   | 15 max                   | 0                    | 0                     | 0                    |
| PC DOS-Compatible  | Yes               | Yes                 | Yes                      | No                   | No                    | Yes                  |
| Device Interrupts  | 11                | 11                  | 11 x 256                 | 1/slot               | 4/slot                | 11 (ISA)             |
| Maximum Number of<br>Interrupting I/O                              | 11                | >= 11 *             | 2816                     | # slots              | <= 16<br>/ system     | # slots              |
| Sharable?  | No                | Y/ N *              | Yes                      | Yes                  | Yes                   | No                   |
| Number Sharable  | 0                 | 11 (ISA)*           | 11                       | 1/slot<br>/ system   | <= 16                 | 0                    |
| Slots per Bus  | 8                 | 8                   | 8                        | > 2                  | < 5 ***               | 3                    |
| Relative Size  | 1.0               | 1.0                 | 0.8 - 1.0                | 0.15                 | 1.0                   | 1.0                  |
| Relative Power   | 1.0               | 1.5                 | 0.5                      | 0.1                  | 0.5 - 1.5             | 1.0                  |
| Power-on Insertion   | No                | No                  | No                       | Yes                  | No                    | No                   |
| Relative Cost/Card   | 1.0               | 1.3 - 2             | 1.3 - 2                  | 2 - 3                | 1.3 - 2               | 1.1                  |
| Approximate Number<br>of Cards Available                           | 4000<br>stable    | < 300<br>stable     | 1200<br>stable           | 100<br>growing       | < 200<br>growing      | < 200<br>stable      |
| Parity Support   | No                | No<br>(byte)        | Yes                      | No<br>(transfer)     | Yes                   | No                   |

\* EISA function can be disabled if ISA cards are installed. Automatic configuration is available only if all cards are EISA cards or if all cards are ISA-PNP in newer systems. EISA interrupts used by ISA cards defeat sharing in EISA systems. EISA throughput can be limited by ISA cards.

\*\* Although a specification exists for ISA and VL bus masters, both lack critical preemption protocols, which prevent general implementation.

\*\*\* PCI allows 10 loads maximum; cards in slots are two loads each. System typically uses three loads, yielding a practical limit of three card slots per PCI bus. More PCI slots can be defined if performance is reduced. PCI requires a standard I/O bus for PC/DOS compatibility.

Figure 9. Comparison of Major Buses

# Parity in RAM and Its Effect on Data Integrity

**A**lmost every computer transaction involves memory as either a source or destination. Data can come from a LAN, diskette, communication port, or even the keyboard. It can travel from memory to almost every I/O device.

If an undetected error occurs while the data is temporarily resident in RAM, the error can be propagated to other users or files and can potentially corrupt the integrity of other computer systems. Especially when a client system is connected to a central database or to a mainframe, a significant potential is present for degrading the data quality. Many applications such as games and word processing can forgive a single-bit failure. Other applications such as accounting, spreadsheets, process control, and computer-aided design are very critical of a single-bit failure.

For several decades parity checking in RAM has been the design standard for mainframes, minicomputers, workstations, and quality microcomputer systems—all are intended for business-critical use.

Memory parity can catch errors that affect a single bit, as well as errors in which the entire 8-bit byte is lost.

Although parity checking cannot detect two errors in the same byte, the probability that a second parity error will occur in the same byte—out of the many millions of RAM bytes—is very small, even infinitesimal.

What, exactly, is parity checking? In a computer, a byte consists of eight bits, each containing either a 1 or a 0. Parity-checking memory contains a ninth *parity* bit, also either 1 or 0, depending on whether the number of bits in the eight regular bits set to 1 is odd or even. By counting the number of bits set to 1, then checking the parity bit, the computer will know whether the data in the bit is valid or erroneous.

The following example illustrates the importance of memory parity in desktop systems.

A sales agent at an insurance company develops a rate estimator to quickly estimate bids in the field. A formal bid, with rates calculated by the home office, is subsequently generated. The agent's rate-estimator program is so successful in generating additional business that it is copied from user to user throughout the insurance company's network.

The original copy of the program yields acceptable estimate data, but some of the copies distributed elsewhere in the

company develop errors, which are not detected for some time.

Eventually, an agent gives a customer a quote that is significantly better than quotes from competing companies. The customer ignores all the other quotes and focuses on the attractive one. However, when the formal proposal arrives from the home office, it is much higher than the original estimate. The customer is irritated, and the insurance company loses the business.

Worse, thousands of copies of the defective estimating program have now been innocently copied throughout the company, setting the stage for many more misquotes and dissatisfied customers. The detective work necessary to recover the defective programs and to prevent their further use consumes considerable time and money.

This example illustrates that the information inside a computer is the most valuable component in the system, and as such, it should be protected to the highest degree possible. As you can imagine, the point in this example is the lack of memory parity in most inexpensive business PCs.

The error occurred in a single byte of a table used as the basis for

## Parity is the *Context* Around Data

If you were to hear that a dog took a bite out of my foot, it can be assumed that "bite" is the correct spelling of the word.

If you heard that I lost a byte of data from a diskette, you could assume that "byte" was the correct spelling.

If you heard that the dog took a bite (or byte?) out of a diskette, you can't be sure of the correct spelling—the sentence lacks context.

In a computer, the context for each byte of data can be stored as a ninth bit. If the number of 1's in a byte is even or odd, then the parity bit is either a 1 or a 0. By counting the number of 1's in a byte of data and checking the parity bit, the computer can tell if the data is valid or not. This is called *data parity checking*. Storing a ninth bit means that the memory system will cost 1/8 more than if parity checking were not defined.

Parity is critical in a business computer.

## Parity in RAM and its Effect on Data Integrity (continued from page 43)

estimating. Using network activity logs, the error was traced to the computer from which all the defective copies originated.

A single-bit error had occurred spontaneously in that PC's RAM, but the PC did not have memory parity checking in RAM. The error occurred after the program file was received over the LAN and loaded correctly into RAM, but before it was stored on the system's hard disk. When the program file was stored, the error was also stored. Without parity checking, the defective bit was not detected when it was saved from RAM to disk. After that, the program file containing the error was passed as good data to users of other client systems on the network.

A client PC that had no parity in RAM was the weak link in the entire network and caused all sorts of havoc.

Since it is obvious that parity checking is crucial, why do some PC manufacturers deliberately omit parity checking in their systems?

Manufacturers of inexpensive computers commonly explain omitting parity in RAM as an effort to reduce unnecessary system halts. They assert that the parity-checking circuitry itself can fail and lead to false alarms that halt the system unnecessarily.

Although no logic design is perfect, the parity-check circuitry's complexity is in the range of a few hundred gates, whereas the memory system has tens of millions of gates.

Therefore, the probability of parity circuitry failure is much smaller than the probability of failure in the memory system.

Let's cut to the chase: The real reason that manufacturers of low-cost systems do not implement parity in memory is that its absence saves them 1/8 of their system's memory cost!

Here is a simple analogy: It is possible to get false alarms from fire systems in buildings, but that possibility does not dissuade building codes from requiring smoke detectors and sprinkler systems in schools and high-rise buildings.

When it comes to parity checking in RAM, prudent engineering design errs on the side of caution!

depend on the processor to move the data to memory. The IDE interface is inexpensively derived from ISA by using only a few gates of logic.

Effectively, then, ISA, VL, and IDE constitute the structure of most low-end systems. The ISA bus is adequate for supporting data transfer to and from the fastest device—the IDE device—when no other devices are active, and VL supports 32-bit video at the processor's speed.

A large number of inexpensive ISA systems are stand-alone configurations, not attached to any network. Both VESA-VL and ISA offer a significant number of adapter cards at low cost. In these systems, economies concerning the integrity of data, such as elimination of memory parity, are commonplace.

### **Type 2—Single-Tasking, Occasional Multitasking**

This type of system is primarily single-tasking, so the choice of an IDE file with sequential file access operation performs sufficiently for most applications.

Intended for business applications with larger files and a network client

attachment, a type 2 machine also supports occasional multitasking operations.

A type 2 system can require file adapters with 32-bit I/O that can attach to PCI. Because fewer than 11 additional I/O devices are typically required, ISA is acceptable for the remaining 16-bit I/O devices. Where necessary, a PCI bus can also support the addition of SCSI files, which permit file devices to operate concurrently in a multitasking system.

### **Type 3—Single-Tasking and Multitasking, Networked**

This type of system is intended for power users who frequently use multitasking. Because a type 3 system often acts as a client system on a network, it requires high data integrity in RAM and the other critical I/O interfaces.

#### **Cache Memory**

These requirements imply parity RAM and a high-integrity cache. Premium systems not only need parity in RAM, but their cache systems must be designed to protect against loss of data as well.

A *cache* is a small, but very fast memory containing more frequently used portions

of data or a program (or programs in multitasking) from the system's memory. Because programs tend to execute in some order, the next most likely required contents in system memory can be managed by logic circuitry. In this way, processor requests for memory contents are "hits," where the contents are in the fast cache, rather than "misses," where the processor must wait for the memory contents to be fetched into the cache from the much slower general system memory called *backing store*. Within limits, increasing the size of the cache increases the probability of a hit and thereby increases performance.

A cache can work in several ways. If it is a *write-through* cache, its data is always written to backing-store memory before the processor is released to the next operation. If it is a *write-back* cache, the processor is released immediately, and the data is written to the backing store at the next convenient opportunity. In both cases, if the data is in the cache, processor reads are immediate on a hit and delayed on a miss.

Without the wait to put its data into backing store, a write-back cache is

somewhat faster. However, it can be difficult to maintain data integrity with bus masters (either Micro Channel or PCI) without periodically halting the operation and negating the advantage of high-speed transfer.

In low-end desktop systems, bus masters are infrequent. Integrity with bus masters is less of an issue, so write-back is typically used without problems. In some premium systems, write-through is the typical prescription because bus masters are far more typical. In such cases, performance takes a back seat to data integrity when a design choice must be made.

However, the newest IBM PS/2 systems employ a unique buffered write-through design that solves the integrity or performance problem of write-back with a high-speed protected buffer in the Micro Channel bus controller/memory controller module. This form of write-through cache performs within a few percentage points of write-back designs and yields high throughput by allowing streaming-data-mode files and LAN adapters to move data into memory at full speed while optimizing for data integrity.

#### Network Administration

Corporate client systems typically need a fully administrable system for complex networks that can be monitored accurately from a remote point. In this type of system, I/O will often operate concurrently, implying a need for the overlapped file operations of SCSI file devices and high-throughput data transfer.

#### File Interfaces

In servers, multiple file devices operating concurrently will increase the need for a SCSI interface to multiple file devices in the file system. The processor will coordinate the multiple concurrent I/O operations, so the ability to offload data-transfer responsibilities to bus masters and DMA slave devices is very desirable.

Although some users may select a type 3 system for single-tasking, such systems generally have bus master file adapters, DMA for each parallel and serial port, DMA fax, and DMA or bus master LAN. Every interface should be as wide as possible for the greatest efficiency in data transfer, especially in server systems.

| File Interfaces and Adapters | Single-Tasking Sequential Seek | Multi-Tasking Overlapped Seek            |
|------------------------------|--------------------------------|--|
| IDE (typically PIO slave)    | With one adapter and device    | With multiple adapters only              |
| SCSI                         | With one adapter and device    | With one adapter and one or more devices |
| Programmed I/O slave adapter | Adequate for single-tasking    | Can be parasitic to other tasks          |
| Bus master adapter           | Same performance as PIO slave  | Frees processor for other tasks          |
| DMA slave adapter            | Same performance as PIO slave  | Frees processor somewhat                 |

Figure 10. Comparison of File Interfaces and Adapter Types

#### Automatic Configuration

Type 3 systems typically have very complex configurations and can be supported by a wide variety of complex software configurations. Every interface requires automatic configuration, because setup expenses might otherwise absorb a considerable percentage of the initial purchase. For this reason, all buses in the system must support plug-and-play protocols.

Multimedia for advanced systems typically implies a full complex of auxiliary functions. Consequently, the Video Media Channel is the prescribed interface for supporting multiple concurrent multimedia operations.

#### File Interfaces, Bus Masters, and DMA Slaves

In the single-tasking environment, programmed I/O slave adapters are economical and sufficient, as long as they do not need more than 100 percent of any resource. Also, bus master and DMA slaves are advantageous only if they are more responsive than programmed I/O adapters.

However, in a multitasking environment, programmed I/O adapters act as parasites on the processor, with each adapter using the processor as a common resource. On the other hand, the use of bus masters and DMA slaves conserves processor bandwidth. Figure 10 summarizes and compares file interfaces and adapter types.

#### Type 4—Portable

Type 4 systems typically place a premium on small size, low weight, and low power consumption. File systems are often miniaturized IDE types, because multitasking is infrequent. Configurations are typically simple, with only a few I/O devices per system, but the configuration can vary spontaneously as adapters are dynamically added or removed, systems are plugged into docking stations, or I/O devices are installed or removed among mobile systems for data exchange.

The requirement for plug-and-play in these systems is driven not by complexity, but rather by dynamic variation of the configuration. Reconfiguration occurs more frequently, so it must be made less difficult.

The very nature of a portable system also necessitates automatic configuration of networks that it may join and leave. Therefore, features that support network citizenship are very desirable. Remote support of diagnostics is also desirable for a system that travels and is a user's primary "lifeboat" system when away from the office.

The fixed configuration is typically DOS-compatible and not bus-dependent. Screen and keyboard quality, overall weight, and battery life typically are more important than bus functionality, with the exception of automatic plug-and-play configuration.

| System Requirements                   | Type 1      | Type 2       | Type 3           | Type 4                  |
|---------------------------------------|-------------|--------------|------------------|-------------------------|
| Throughput (32-bit I/O)               | Low         | Medium       | High             | Low                     |
| Automatic plug-and-play configuration | Low         | Some devices | High             | High                    |
| Cost restriction                      | High        | Medium       | Medium           | Lower for same function |
| Multitasking I/O support              | Low         | Minimal      | High             | Low                     |
| Number of I/O devices attached        | 11 adequate | > 11         | High             | Low                     |
| Size restriction                      | Low         | Low          | Low              | High                    |
| Weight restriction                    | Low         | Low          | Low              | High                    |
| Power restriction                     | Low         | Medium       | Medium           | High                    |
| Data integrity                        | Low         | Some         | High devices     | Medium                  |
| Network affinity                      | Low         | Medium       | High             | High                    |
| Design choice:<br>Local bus type      | VL          | VL/PCI       | Internal/<br>PCI | Internal                |
| I/O bus type                          | ISA         | ISA          | Micro<br>Channel | PCMCIA                  |
| File interface                        | IDE         | IDE          | IDE/SCSI         | IDE<br>(miniature)      |

Figure 11. Local Bus, I/O Bus, and File Interface by System Type

### Systems Requirements Matrix

The requirements for the four types of systems discussed above are summarized in Figure 11. Note that the requirements between system types may overlap. Low-end premium systems may be used as single-tasking network clients or as multi-tasking systems or servers. Type 2 systems may have some models with VL local bus and other models with requirements for 32-bit I/O adapters that lead to a choice of a PCI local bus.

### System Design Requirements for All Types

Each system design, no matter which type, must include choices for local bus, I/O bus, and file adapter interface. Data integrity and network administration benefits may weigh heavily in corporate

environments. The ability to run multi-tasking and server operating systems efficiently can have a major effect on system design, bus selection, and file adapter interface selection.

### Price Points, Requirements, and Components

This section matches system costs and requirements listed in Figure 11 with the components listed in Figures 9 and 10 that best fit.

#### Lowest-Cost System

In the lowest-cost system, designs can limit the supported I/O to a total of 11 adapters, meaning the sum of the number of sockets and system-board I/O adapters does not exceed 11. The ISA bus plus inexpensive adapter cards is the solution here.

Given the presence of ISA, a VESA-VL local-bus video display adapter is easily and inexpensively added. Single-tasking IDE files are often chosen, because the cost of implementing the IDE interface from the ISA bus signals is minimal.

Requirements for data integrity are less stringent than in more expensive designs, so parity error checking is often removed from system memory in the least expensive machines in this category, thereby saving 1/8 of the system memory cost.

Automatic configuration is also limited in these systems, due to an absence of VL plug-and-play designs and a near absence of ISA plug-and-play cards.

The PC industry produces machines in high volume around this basic design point, much like the radio industry of the 1950s defined a standardized five-tube design. In both cases, the design point was minimal price for minimal function.

#### Medium-Cost System

At a medium-price design point, system designers can add a more sophisticated local bus design such as PCI, theoretically permitting more I/O devices than the 11 ISA adapters. PCI, with a limited number of sockets, can define 32-bit I/O devices and bus master capability for a portion of the I/O adapters defined in these systems.

Given the presence of PCI with 32-bit adapters, EISA, with its limited data integrity, has little added value over ISA.

Parity memory becomes a requirement. These systems are more likely to share data with other systems and with network servers, so they need better protection against data corruption as files pass through the system to other network citizens.

Although automatic plug-and-play configuration is possible on PCI bus designs and can be inexpensively added to the ISA system design, automatic configuration awaits the availability of ISA plug-and-play adapters that will replace a satisfactory number of the 4,000 non-plug-and-play designs presently in the market.

#### Premium-Cost System

The premium design point assumes a requirement for data integrity, data parity



on all buses, and parity on all system memory to protect critical corporate or financial data that is exchanged on a network. Therefore, only two local bus types can be considered: internal and PCI. Where socketed adapters are considered desirable, PCI becomes the optimal bus choice, because PCI is the only local bus standard that defines parity. Similarly, there is only one I/O bus type that defines parity: Micro Channel.

In the most technically demanding market segment—premium desktop and server systems—the synergy of the marriage between Micro Channel and PCI is most apparent. PCI is limited in the number of sockets that can be provided on a local bus to devices that need random-access control, such as displays. Micro Channel can provide a large number of sockets to support PCI throughput levels to the remaining block-oriented devices, such as files, communications, printing, and LAN.

Micro Channel supports plug-and-play on every card design—as does PCI—so the entire system, not just a portion, can configure automatically. However, PCI is limited in its ability to provide PC DOS-compatible interrupts and Direct Memory Access, whereas Micro Channel provides these functions generously.

By providing shareable interrupts at each of the 11 AT-compatible signals, Micro Channel delivers more potential adapters per system and more potential configurations to the user than any other I/O bus in combination with PCI.

Micro Channel also supports up to 15 DMA slave devices as inexpensive multitasking processor alternatives for moving data. No other I/O bus can either define this many slaves or make this many available to adapter cards.

### Premium File Systems

File systems in premium computers can support single-tasking network client scenarios using IDE-type file adapters. IDE is also easily derived from Micro Channel.

However, SCSI adapters become more desirable as multitasking requirements increase. Newer premium Micro Channel systems allow the user to begin with

inexpensive programmed I/O slave SCSI file adapters and upgrade to a bus master design at a later time when making the transition to multitasking. By initially selecting SCSI as the file interface, the investment in the file device itself can be retained during the bus master SCSI adapter upgrade. Further, a hard-disk device from a previous PS/2 can be installed for data migration and later more file devices can be added to the original controller (unlike IDE) to add hard-disk space. Therefore, in premium systems, servers, and predominantly multitasking scenarios, SCSI file devices are highly recommended.

### Mobile System

In the mobile segment, size, power, and weight combine to limit the I/O bus choice to PCMCIA. The local bus is typically internal and customized. Externally, the mobile system can dock to an expansion unit or a network to provide more I/O.

### Decision Flowchart

To help match system components to requirements, Figure 12 presents a decision flowchart specifying the most likely components in various situations.

### The User's Decision

No single standard for local bus, I/O bus, or file interface is optimal for all applications. Instead, when purchasing a computer system, users are presented with choices. Prudent users will anticipate their operating system and application requirements and will choose the appropriate computer system architecture accordingly. Prudent users will also realize that purchasing a system based on the needs of the moment, or simply along price/performance lines, can quickly lead to obsolescence (a system that cannot be upgraded).

Other system considerations—multitasking operating system optimization, size, weight, power, plug-and-play support, system integrity, expandability, and flexibility—may weigh more heavily over the system's entire life.

Only the user can decide which factors are most important in purchasing a computer system and which letters to spoon out of the alphabet soup of standards.

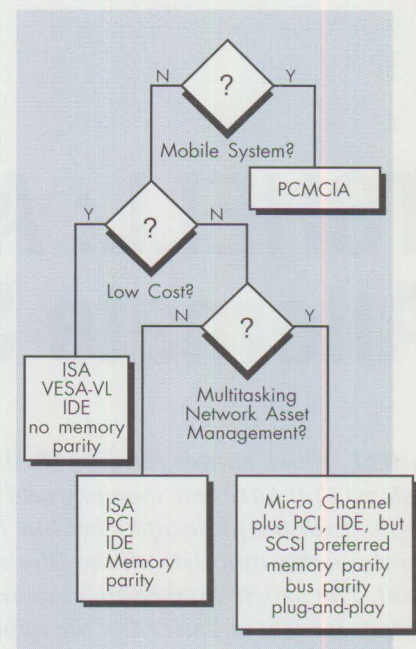


Figure 12. System Component Decision Flowchart



**Chet Heath** is a Senior Technical Staff Member and an engineer in the IBM Personal Computer Company in Boca Raton, Florida. Chet holds a BS degree in electrical engineer-

ing from the New Jersey Institute of Technology and an MS degree in electrical engineering from the IBM LSI Institute at the University of Vermont. He refers to himself as "the oldest living survivor of Micro Channel architecture," because he was involved in its definition since inception and gave the architecture its name. Chet has received IBM Quality, Outstanding Technical Achievement, Outstanding Innovation, and Corporate Technical Achievement Awards. He has also attained the eighth level of invention awards. Chet joined IBM in 1970.

# TSHELL: A Text-Based Alternate Shell for OS/2

*TSHELL, which stands for Text Shell, allows OS/2 to run on computer systems that have low memory and hard-disk resources. This article begins with the philosophy behind TSHELL. It then discusses system requirements and lists all the files needed in the OS/2-TSHELL environment. The article then changes course by providing case studies of customers for whom TSHELL is the appropriate user interface. Next, the article illustrates how to use TSHELL, concluding with the author's testing methodology.*

**G**raphical User Interface (GUI) and Common User Access (CUA) technologies have had a revolutionary impact on the computer industry. These facilities don't come cheap; they put an ever-increasing demand on processor power, memory utilization, and hard-disk capacities. Although the decreasing cost of increasing hardware power somewhat offsets the burden of this GUI/CUA technology, it does not relieve it entirely.

I am not an anti-GUI sentimentalist—in fact, I am quite the reverse—but I recognize that there are diverse interests to be served in the computing world, and some of those interests have no need or desire to use GUI and CUA facilities. As an independent consultant, I need to be in a position to service all of my clients' needs, and the combination of OS/2 and TSHELL plays a key role.

---

**Bill O'Connor**  
Executive Computer Systems  
Toms River, New Jersey

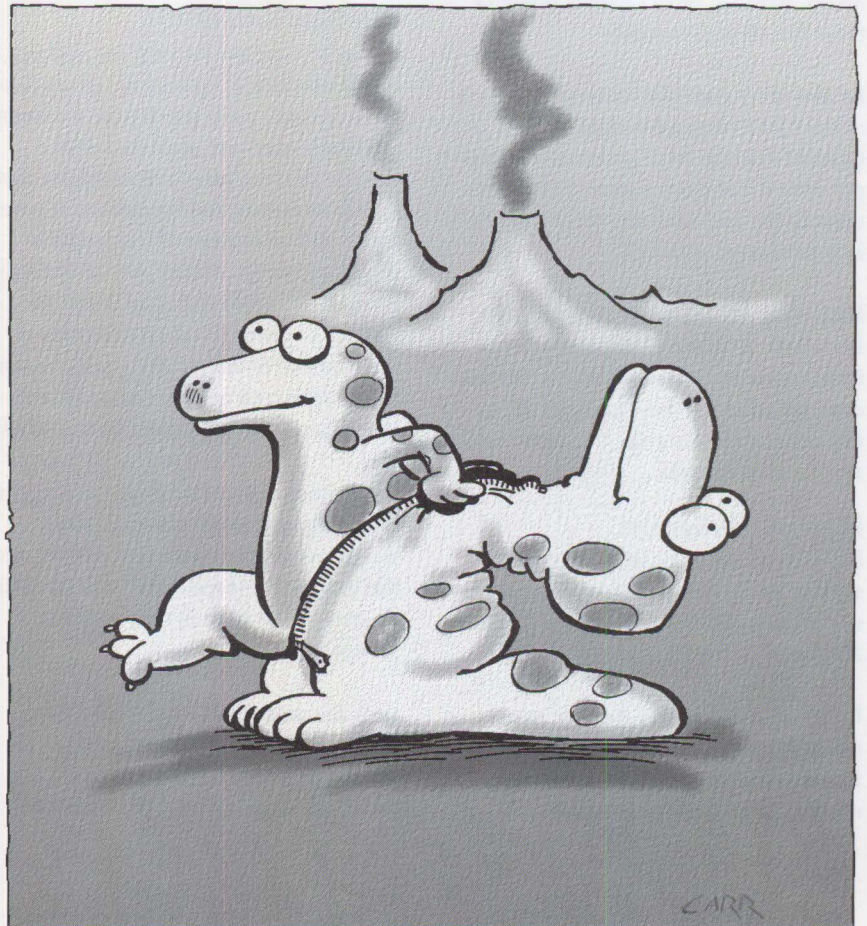
---

It is important to put the OS/2-TSHELL combination into perspective. This combination should not be construed as conflicting with the ultimate goals of OS/2. Rather, it should be viewed as a tool that enhances

the usability, and therefore the acceptability, of OS/2 in the marketplace.

OS/2 is an extremely powerful and diverse operating system, and as such there are many alternate ways to use it. This built-in diversity was a significant design decision for OS/2 developers. OS/2 was designed to be a truly "democratic" operating system; it provides facilities for effective use by DOS, Windows, 32-bit OS/2, 16-bit OS/2, GUI, and non-GUI users. OS/2 welcomes all these users and does a fine job of facilitating all of their needs.

I have been talking about and using OS/2 in a non-GUI environment for years, yet I



run into resistance from many people who refuse to consider OS/2 in any way other than a GUI/CUA system. I keep wondering: Why would anyone want to categorize an operating system in such a way? I assume that these people simply don't understand the diversity of the marketplace.

## OS/2-TSHELL Hardware Requirements

The current version of the OS/2 2.1-TSHELL combination operates effectively with the following minimum hardware requirements:

- 16 MHz 386SX processor, or higher.
- 3 MB of RAM.
- Less than 2 MB of hard-disk storage.
- A monochrome monitor. Obviously, a graphical monitor is required for any graphical work, but a monochrome monitor is all that is needed here.

I do not claim that this is the "recommended" or "ideal" configuration. I am simply saying that this is the minimum configuration that OS/2 2.1 requires with TSHELL.

Can OS/2 run effectively with this minimum configuration? Absolutely—I do it every day! I don't have to use the OS/2 2.1-TSHELL combination, because my hardware exceeds the minimum requirements. Rather, I use this combination because some of my customers use it, and I need to keep current.

The minimum configuration outlined above enables you to run multiple 32-bit OS/2, 16-bit OS/2, and virtual DOS machine (VDM) sessions simultaneously. With a graphical monitor, you can also run Windows 3.1 software (within WIN-OS/2 sessions) with this minimum configuration.

The 2 MB of hard-disk storage does not include the space required for the OS/2 swap file or the space required for your Windows 3.1 software if you choose to use that software.

The last half of this article lists the specific files and the contents of various configuration files that you will need in order to use some or all of the facilities in the OS/2-TSHELL combination.

## TSHELL Accesses the Full OS/2 Kernel

The minimum configuration stated above gives you the full capabilities of the OS/2 kernel, which is the heart of OS/2. Here is a short list of those capabilities:

- True 32-bit processing
- Protected-mode operation
- Dynamic memory management using the 32-bit flat memory model
- Priority-based, preemptive multitasking of DOS, Windows, 32-bit OS/2, and 16-bit OS/2 tasks

No other operating system accomplishes these things so efficiently, and the OS/2-TSHELL combination accomplishes them best for users who have a low-resource environment. Low-resource users can use TSHELL to begin taking advantage of the power of OS/2. Later, they can add to their resources as their needs dictate and as time permits. This migration path is an attractive option for many OS/2 users.

## TSHELL Combines with a Graphical Interface

It is important to understand that you can use both the OS/2-TSHELL combination and the full Workplace Shell/PMSHELL interface. The two interfaces are not mutually exclusive systems, so there is no reason why they cannot both be used. However, you cannot use both interfaces at the same time on the same computer. A simple switch of the CONFIG.SYS file is all that is necessary to alternate between the two environments. DOS users have been using this switching mechanism for many years, and there is no reason why OS/2 users should not be able to enjoy the same type of facility.

## Inside the OS/2-TSHELL Combination

TSHELL has several components that enable the effective use of OS/2. The following components are some highlights.

### TSHELL.EXE

The TSHELL.EXE program is relatively small (about 22 KB) and provides you with a textual menu interface consisting of two parts: the Start Group or task selection list at the left side of the screen and the Running Group or task running list at the right. Menu items can be selected using the arrow keys on the keyboard.

# How to Get TSHELL

TSHELL is free and is available on all the major bulletin board systems:

- Internet
- Fidonet
- CompuServe
- IBMLink

The TSHELL.ZIP package contains documentation and examples plus the PGMSHELL.EXE file. The optional STARTDOS.EXE file is available separately as STARTDOS.ZIP and it is included with the MSHELL.ZIP package. Be sure to read about TSHELL's restrictions (known bugs) in the README file.

The TSHELL textual menu interface appears quite similar to the original OS/2 1.0 shell and the DOS Shell. All of these shells came from the same source—the OS/2 1.0 shell—which lends credence to the adage "The more things change, the more they stay the same."

The familiar Alt+Esc and Ctrl+Esc key sequences are used to navigate through the sessions as well as to display the master menu and task list.

### PGMSHELL.EXE

The PGMSHELL.EXE program occupies about 50 KB and lets you build and maintain custom user menus for use with TSHELL.

This optional facility works hand-in-glove with the REXX facility that is an integral part of OS/2. PGMSHELL's documentation provides simple, clearly defined examples that show how to set up a customized menu for TSHELL. The PGMSHELL.EXE facility may be used effectively even by users who have only a casual knowledge of the REXX system.

### STARTDOS.EXE

The STARTDOS.EXE program requires

|                                   |               |                                      |
|-----------------------------------|---------------|--------------------------------------|
| A:\                               |               |                                      |
| os2boot                           | 1,099 bytes   |                                      |
| autoexec.bat                      | 169 bytes     | for use with DOS                     |
| config.dos                        | 1,258 bytes   | boot from A: with DOS on C:          |
| config.a                          | 705 bytes     | boot from A: no DOS                  |
| os2krnl                           | 734,312 bytes |                                      |
| os2ldr                            | 28,160 bytes  |                                      |
| os2ldr.msg                        | 8,516 bytes   |                                      |
| vtbl850.dcp                       | 10,478 bytes  |                                      |
| config.sys                        | 2,292 bytes   | currently active Menu Master from C: |
| config.001                        | 3,068 bytes   | full WPS from C:                     |
| config.002                        | 1,793 bytes   | TSHELL from C:                       |
| 791,850 bytes in 11 files         |               |                                      |
| A:\OS2                            |               |                                      |
| chkdsk.com                        | 68,656 bytes  |                                      |
| ckmem.exe                         | 19,412 bytes  | my memory analyzer                   |
| clock01.sys                       | 3,632 bytes   |                                      |
| cmd.exe                           | 90,624 bytes  |                                      |
| country.sys                       | 25,610 bytes  |                                      |
| tshell.exe                        | 17,547 bytes  |                                      |
| harderr.exe                       | 14,824 bytes  |                                      |
| ibm1flpy.add                      | 25,102 bytes  |                                      |
| ibm1s506.add                      | 22,767 bytes  |                                      |
| kbd01.sys                         | 29,525 bytes  |                                      |
| os2dasd.dmd                       | 33,236 bytes  |                                      |
| screen01.sys                      | 5,590 bytes   |                                      |
| sl.exe                            | 9,755 bytes   | TINYED                               |
| 366,280 bytes in 13 files         |               |                                      |
| A:\OS2\DLL                        |               |                                      |
| ansicall.dll                      | 438 bytes     |                                      |
| bkscalls.dll                      | 401 bytes     |                                      |
| bmscalls.dll                      | 398 bytes     |                                      |
| bvhinit.dll                       | 9,443 bytes   |                                      |
| bvhvga.dll                        | 40,892 bytes  | only needed when using DOS           |
| bvscalls.dll                      | 454 bytes     |                                      |
| doscall1.dll                      | 90,854 bytes  |                                      |
| kbdcalls.dll                      | 858 bytes     |                                      |
| moucalls.dll                      | 1,010 bytes   |                                      |
| msg.dll                           | 508 bytes     |                                      |
| nampipes.dll                      | 1,024 bytes   |                                      |
| nls.dll                           | 466 bytes     |                                      |
| npxmltr.dll                       | 25,280 bytes  |                                      |
| os2char.dll                       | 56,000 bytes  |                                      |
| quecalls.dll                      | 15,250 bytes  |                                      |
| sesmgr.dll                        | 32,806 bytes  |                                      |
| viocalls.dll                      | 1,825 bytes   |                                      |
| 277,907 bytes in 17 files         |               |                                      |
| 1,436,037 total bytes in 41 files |               |                                      |

Figure 1. Contents of Single OS/2 Boot Diskette

about 50 KB and allows you to customize DOS VDM processing.

This facility not only duplicates some things that PGMSHELL.EXE does, but it also does a lot more. STARTDOS.EXE also works with the built-in REXX facility; its primary purpose is to provide freestanding DOS VDM processing. It also provides all of the DOS settings available to users of the full Workplace Shell/PMSHELL interface.

Like PGMSHELL.EXE, the STARTDOS.EXE facility is purely optional. However, users who do a lot of DOS VDM processing will find these two facilities indispensable.

The elegant simplicity of the TSHELL, PGMSHELL, and STARTDOS facilities underlies their extreme effectiveness.

All programs mentioned thus far are IBM employee-written software (EWS), and as such they are provided at no charge to any OS/2 user. They may be obtained from IBM electronic bulletin boards as well as from most of the popular OS/2 bulletin boards.

## Disk and Memory Usage with TSHELL

The files listed in this section are the ones that I currently use with my OS/2 test system.

Figure 1 lists all the files that are on my single boot diskette. These files are the same ones that you need on your hard disk to enable multiple OS/2 sessions to run using TSHELL. To be able to run multiple DOS and Windows sessions using TSHELL, you have to ensure that the \OS2\MDOS and \OS2\MDOS\WINOS2 subdirectories are on your hard disk and that the CONFIG.SYS pointers are set up to reflect that.

To use the single boot diskette to install OS/2 on your hard disk, insert the diskette into drive A: and do the following:

1. Type: SYSINSTX x: (where x: is the hard-disk drive that you want to boot from).
2. Type: XCOPY A: X: /S
3. Remove the diskette, then reboot. You are now running this basic OS/2 system from your hard disk.

The list of files necessary for DOS VDM processing is shown in Figure 2.

A quick tally of the files in Figures 1 and 2 shows that the total disk space required is under 2 MB to run 32-bit OS/2, 16-bit OS/2, and DOS VDM sessions in a text-only environment. (In fact, some of the files listed are not really necessary—I included them to do some memory analysis.)

I do not include Windows files because they vary from system to system.

Figure 3 lists the contents of the AUTOEXEC.BAT file to be used when running with DOS and Windows.

Figure 4 lists the contents of a file called CONFIG.A. This is the basic configuration file to use when booting from diskette and running multiple OS/2 sessions that use TSHELL.

Figure 5 contains the CONFIG.DOS file that you would use to boot from diskette and to run multiple OS/2, DOS, and Windows sessions.

Finally, Figure 6 lists the contents of CONFIG.002, the configuration file that takes advantage of all of the facilities of the TSHELL.

Any files listed in Figures 1 through 6 that are outside of the \OS2 directory are for use by my own applications.

Astute readers will notice that the CONFIG files in Figures 4, 5, and 6 are set up to use a VGA monitor. These CONFIG files represent my test system. However, TSHELL may indeed be run using a monochrome monitor.

## Candidates for Using the OS/2-TSHELL Combination

What kinds of users are the most likely candidates for using the OS/2-TSHELL combination? Here are several examples.

### Small Business

One of my clients is a supplier of a proprietary software package for small, local area businesses. This software package is a menu-driven, text-only accounting system, and its users are extremely happy with it. My client supplies a complete turnkey system, including the hardware

|                           |              |                 |
|---------------------------|--------------|-----------------|
| C:\OS2\MDOS               |              |                 |
| vmdisk.exe                | 56,634 bytes |                 |
| vvga.sys                  | 52,795 bytes |                 |
| command.com               | 52,686 bytes |                 |
| mem.exe                   | 39,834 bytes | memory analyzer |
| fsaccess.exe              | 29,488 bytes |                 |
| doskrnl                   | 29,437 bytes |                 |
| vdpx.sys                  | 24,017 bytes |                 |
| vwin.sys                  | 22,800 bytes |                 |
| vkbd.sys                  | 22,247 bytes |                 |
| vdpmi.sys                 | 21,873 bytes |                 |
| vemm.sys                  | 19,008 bytes |                 |
| vmouse.sys                | 14,688 bytes |                 |
| fsfilter.sys              | 11,948 bytes |                 |
| vdma.sys                  | 10,281 bytes |                 |
| vtimer.sys                | 9,168 bytes  |                 |
| vdisk.sys                 | 8,896 bytes  |                 |
| vpic.sys                  | 8,678 bytes  |                 |
| vxms.sys                  | 8,608 bytes  |                 |
| vlpt.sys                  | 8,513 bytes  |                 |
| vcom.sys                  | 8,112 bytes  |                 |
| append.exe                | 7,650 bytes  |                 |
| vbios.sys                 | 7,584 bytes  |                 |
| ansi.sys                  | 7,471 bytes  |                 |
| doskey.com                | 6,199 bytes  |                 |
| vflpy.sys                 | 4,896 bytes  |                 |
| subst.exe                 | 4,520 bytes  |                 |
| join.exe                  | 4,520 bytes  |                 |
| vesa.exe                  | 3,824 bytes  |                 |
| assign.com                | 2,782 bytes  |                 |
| ega.sys                   | 2,111 bytes  |                 |
| vnpix.sys                 | 2,016 bytes  |                 |
| vcmos.sys                 | 848 bytes    |                 |
| vtouch.com                | 647 bytes    |                 |
| vapm.sys                  | 592 bytes    |                 |
| emm386.sys                | 522 bytes    |                 |
| lptdd.sys                 | 499 bytes    |                 |
| comdd.sys                 | 493 bytes    |                 |
| mouse.com                 | 460 bytes    |                 |
| himem.sys                 | 455 bytes    |                 |
| exit_vdm.com              | 7 bytes      |                 |
| 517,807 bytes in 40 files |              |                 |

Figure 2. Files Necessary for DOS VDM Processing

```
@echo off
prompt $p$g
path c:\os2;c:\os2\mdos;c:\os2\mdos\winos2;c:\userdir\dos6;
doskey
set tmp=c:\
set dircmd=/o
set temp=c:\os2\mdos\winos2\temp
```

Figure 3. AUTOEXEC.BAT File for Use with DOS and Windows

```

rem config.a file contents
protshell=a:\os2\tshell.exe
set os2_shell=a:\os2\cmd.exe
set comspec=a:\os2\cmd.exe
set runworkplace=a:\os2\tshell.exe
libpath=.;a:\os2\dll;a:\;
set path=.;a:\;a:\os2;
set dpath=a:\OS2;a:\;
rem the following is to receive oso001.msg
  messages
rem set dpath=a:\OS2;a:\;c:\os2\system;
devinfo=scr,ega,vtb1850.dcp
set prompt=[$p]
autofail=yes
priority_disk_io=no
priority=absolute
buffers=10

```

```

iop1=yes
diskcache=2048
maxwait=3
memman=noswap
rem memman=swap,protect
rem swappath=c:\os2\system 1024 1024
break=off
threads=64
country=001,a:\os2\country.sys
set keys=on
protectonly=yes
set dircmd=/o
pauseonerror=yes
basedev=ibm1flpy.add
basedev=ibm1s506.add /v
basedev=os2dasd.dmd
rem end of config.a

```

Figure 4. CONFIG.A File for Running Multiple OS/2 Sessions That Use TSHHELL

```

rem config.dos contents
protshell=a:\os2\tshell.exe
set os2_shell=a:\os2\cmd.exe
set comspec=a:\os2\cmd.exe
libpath=.;a:\os2\dll;a:\;c:\os2\mdos;
set path=.;a:\;a:\os2;
set dpath=a:\os2;a:\;c:\os2\mdos
rem The following is to receive oso001.msg
  messages
rem set dpath=a:\os2;a:\;c:\os2\system;
  c:\os2\mdos;
devinfo=scr,ega,vtb1850.dcp
set prompt=[$p]
autofail=yes
priority_disk_io=no
priority=absolute
buffers=10
iop1=yes
rem diskcache=64
diskcache=2048
maxwait=3
rem memman=noswap
memman=swap,protect
swappath=c:\os2\system 1024 1024
break=off
threads=64
country=001,a:\os2\country.sys
set keys=on

```

```

protectonly=no
set dircmd=/o
pauseonerror=yes
basedev=ibm1flpy.add
basedev=ibm1s506.add /v
basedev=os2dasd.dmd
device=c:\os2\dos.sys
rem DOS stuff
shell=c:\os2\mdos\command.com
  c:\os2\mdos /p
fcbs=16,8
rmsize=640
files=20
dos=low,noumb
break=on
device=c:\os2\mdos\vvgas.sys
device=c:\os2\mdos\vemm.sys
device=c:\os2\mdos\vdpx.sys
device=c:\os2\mdos\vxms.sys /umb
device=c:\os2\mdos\vdpmi.sys
device=c:\os2\mdos\vwini.sys
rem mouse stuff*****
device=c:\os2\pointdd.sys
device=c:\os2\mouse.sys serial=com2
device=c:\os2\mdos\vmouse.sys
rem end of config.dos

```

Figure 5. CONFIG.DOS File for Running Multiple OS/2, DOS, and Windows Sessions

as well as the software, in a completely networked environment. His client base is roughly 400 small businesses, all of which have local area networks. The network sizes range from three to 15 workstations, with an average of eight. When I first became involved a year ago, the total number of workstations on all networks was about 3,200—every one of which was

a monochrome monitor! (Things have changed somewhat since then.)

This client's system was DOS-based, and as such, it was constantly bumping up against the 640 KB barrier that DOS users know so well. Because of this DOS barrier, my client expressed an extremely high interest in OS/2.

We ported his DOS application to OS/2 without difficulty. His dilemma was how to convince all of his customers to upgrade to more powerful processors, more memory, bigger hard-disk systems, and graphical monitors to use his latest offering. He did not want to do that if there was some other way.

Well, there was another way—the OS/2-TSHELL combination. When I first showed it to my client, he was (to say the least) ecstatic. Here was the perfect opportunity for him to introduce his customers to an advanced software system that required only a minimum hardware upgrade. By simply replacing their older motherboards with 386SX boards and 3 MB of RAM, they were off and running. My client and his customers are now in a position to grow their hardware at a reasonable pace and cost as their software needs grow. To my small-business client and his small-business customers, the OS/2-TSHELL combination is a very attractive option.

### Larger Business

Another client of mine is a government engineering center. This agency has roughly 3,000 PCs, most of them 286-based systems. These folks have many diverse needs, but their computing budget has been severely curtailed due to recent cutbacks.

Their current user community spans the gamut of PC usage, from engineering workstations to secretaries' desks to data/transaction-entry operations, each with different operations and computing requirements. In fact, much of what they currently do is text-only and will remain that way for quite some time. This center is currently evaluating the OS/2-TSHELL combination to determine whether it will fit into their current and future computing needs. I am betting that it will.

### Students

I have encountered many students who currently have 386-based computers with 4 MB of RAM. Many of these students do not have a viable upgrade capability, yet they want to be in a position to learn something about the current state of operating system design. The OS/2-TSHELL combination gives them this opportunity for a reasonable resource investment.

### BBS SysOps

I know several BBS operators who absolutely need the protected multitasking capability that OS/2 offers. But, because their systems run unattended for much of the time, they are reluctant to pay the price of having to use a GUI. After all, what good is a graphical user interface when there is no user to interface with?

```
rem config.002 file contents
rem tshell.....230n ***This rem must be at the top*** tshell
protshell=c:\userdir\tshell\tshell.exe
set os2_shell=c:\os2\cmd.exe /k c:\init.cmd
set comspec=c:\os2\cmd.exe
set runworkplace=c:\userdir\tshell.exe
libpath=.;c:\userdir\tshell\dll;c:\userdir\tshell\dll\dll;
c:\os2\mdos;c:\userdir\userdll;c:\c:\userdir\ibmc\dll;
d:\toolkt20\dll;
set path=c:\userdir\usersys;c:\userdir\usercmd;c:\os2;
c:\os2\mdos;c:\os2\system;c:\os2\install;
set dpath=c:\os2;c:\os2\system;c:\c:\os2\mdos;
set prompt=[$p]
autofail=yes
priority_disk_io=no
priority=absolute
buffers=30
iop1=yes
rem diskcache=64,lw,ac:c
diskcache=128,ac:c
maxwait=3
memman=swap,protect
swappath=c:\os2\system 1024 1024
break=off
threads=96
country=001,c:\os2\system\country.sys
set keys=on
protectonly=no
set dircmd=/o
pauseonerror=yes
rem device=c:\os2\vdisk.sys 2048,512,256
basedev=ibm1flpy.add
basedev=ibm1s506.add /v
basedev=os2dasd.dmd
device=c:\os2\com.sys
device=c:\os2\dos.sys
rem device=c:\os2\pmdd.sys
devinfo=scf,vga,c:\os2\viotbl.dcp
set video_devices=vio_vga
set vio_vga=device(bvhvga,bvhvga)
device=c:\userdir\usersys\menusecu.sys
rem DOS stuff
shell=c:\os2\mdos\command.com c:\os2\mdos /p
fcbs=16,8
rmsize=640
files=20
dos=low,noumb
break=on
device=c:\os2\mdos\vvgas.sys
device=c:\os2\mdos\vemm.sys
device=c:\os2\mdos\vdpx.sys
device=c:\os2\mdos\vxms.sys /umb
device=c:\os2\mdos\vdpmi.sys
device=c:\os2\mdos\vwins.sys
rem mouse stuff
device=c:\os2\pointdd.sys
device=c:\os2\mouse.sys serial=com2
device=c:\os2\mdos\vmouse.sys
rem end of config.002
```

Figure 6. CONFIG.002 File for Running All Features of TSHELL

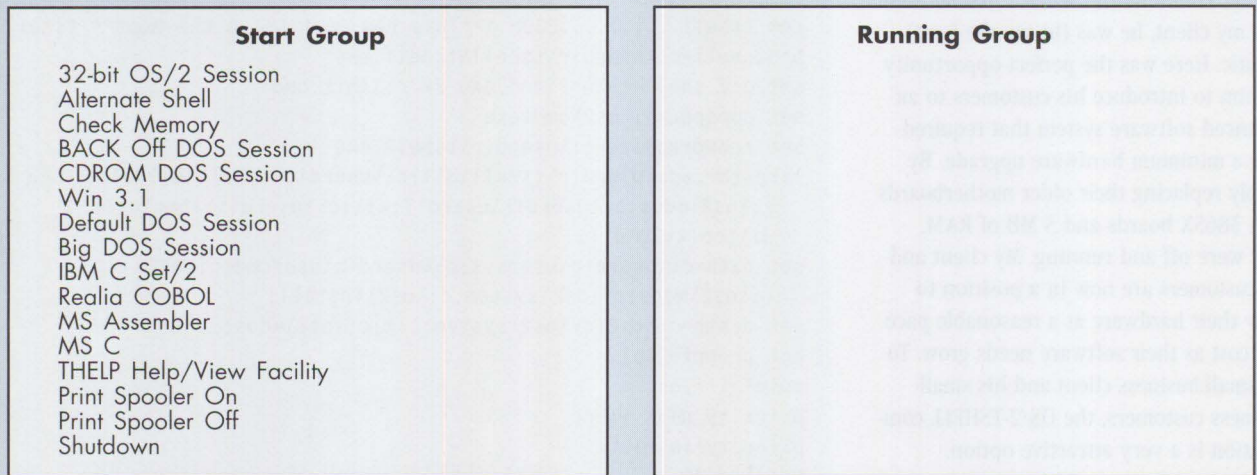


Figure 7. TShell Menu Screen at Startup

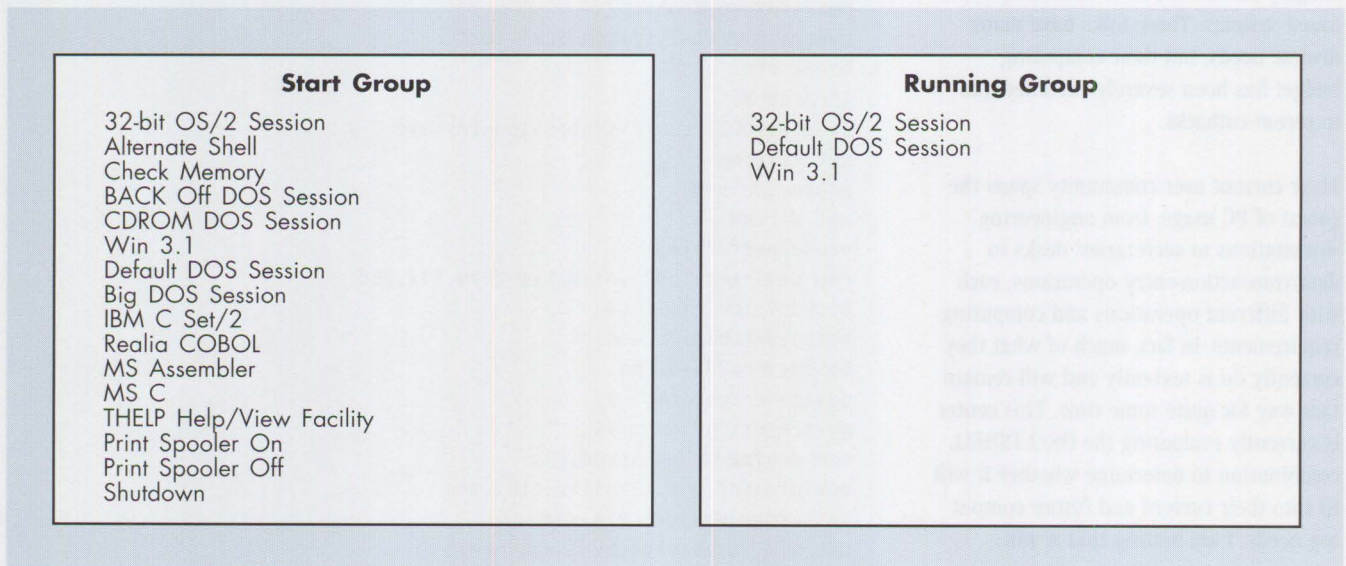


Figure 8. TShell Menu Screen After Starting Three Sessions

```
@echo off
cd \
cd \userdir\tshell
pgmshell xxdos
exit
```

Figure 9. STARTUP.COM FILE

The OS/2-TShell combination offers these BBS operators the best bang for their buck.

#### Developers

As a developer of software, I have many different hardware platforms at my disposal, but my primary test machine is a

20 MHz 386DX with 10 MB of RAM. Many times when I do a lot of compiling and testing, I use the OS/2-TShell combination because it lets me use my extra memory for such things as cache areas and/or virtual disks (VDISKS). I can use my resources in ways that do the most good, when and where I need them.

#### Hardware Resellers

While I have never been a hardware reseller, I can safely say that if I ever were, I would certainly want to show my hardware in the best possible light. The OS/2-TShell combination makes anyone's system shine in comparison to any other currently available software system and affords a competitive edge.

#### Using TShell: A Scenario

Using TShell is simple, as illustrated in this short scenario. The normal OS/2 Ctrl-Esc key sequence brings up the main TShell menu screen, shown in Figure 7. It consists of two halves, Start Group and Running Group. Figure 7 shows the TShell menu screen at startup time when using the sample STARTUP.COM and XXDOS.COM files listed above.

To navigate through and to highlight entries in the Start Group, use the up/down arrow keys; to select a job to run, use the Enter key. This action is equivalent to pointing and clicking on an icon in the startup folder when using



```

/* REXX program to modify TSHELL menu */
    if 'PGMSHELL' <> address() then do
        say 'Expected PGMSHELL environment.'
        say 'Usage: PGMSHELL <cmd filename>'
        return 2
    end

/* title text for the start list */
    rc = SetStartTitle( "Start Group" )

/* title text for the running list */
    rc = SetRunningTitle( "Running Group" )

/* Add OS2 program; arguments: title, startup dir, parameters, exe */
    rc = AddOS2Program( "32-bit OS/2 Session",,, "CMD.EXE" )

/* Add OS2 program; arguments: title, startup dir, parameters, exe */
    rc = AddOS2Program( "Alternate Shell",,, "menuini.exe" )

/* Add OS2 program; arguments: title, startup dir, parameters, exe */
    rc = AddOS2Program( "Check Memory",,, "ckmem.exe" )

/* is configured for DOS? */
    if QueryDOSCapable() then do

/* Add DOS program; arguments: title, startup dir, parameters, settings stem */
        drop settings
        settings.0 = "DOS_BACKGROUND_EXECUTION=OFF"
        settings.1 = "XMS_MEMORY_LIMIT=0"
        settings.2 = "EMS_MEMORY_LIMIT=0"
        settings.3 = "XMS_MINIMUM_HMA=0"
        rc = AddDOSProgram( "BACK Off DOS Session",,, "settings" )

/* Add another DOS program, CDROM DOS Session */
        drop settings
        settings.0 = "DOS_STARTUP_DRIVE=d:\userdir\bootcdrom\bootcdrom"
        rc = AddDOSProgram( "CDROM DOS Session",,, "settings" )

/* Add another DOS program, Win 3.1 */
        rc = AddDOSProgram( "Win 3.1",,, "/c win", "" )

/* Add another DOS program, Default DOS */
        rc = AddDOSProgram( "Default DOS Session",,, "" )

/* Add another DOS program, Big DOS */
        drop settings
        settings.0 = 'DOS_HIGH=1'
        settings.1 = 'DOS_UMB=1'
        settings.2 = "VIDEO_MODE_RESTRICTION=MONO"
        settings.3 = "XMS_MEMORY_LIMIT=4096"
        settings.4 = "EMS_MEMORY_LIMIT=4096"
        rc = AddDOSProgram( "Big DOS Session",,, "", "settings" )

        end /* end DOS configuration examples */

/* Add OS/2 programs; arguments: title, startup dir, parameters, exe. The */
/* following are examples of using a cmd procedure for starting a special */
/* 32-bit .CMD with session. */
    rc = AddOS2Program( "IBM C Set/2 ",,, "/k ibmc2", "cmd.exe" )
    rc = AddOS2Program( "Realia COBOL ",,, "/k kobol", "cmd.exe" )
    rc = AddOS2Program( "MS Assembler ",,, "/k closasm", "cmd.exe" )
    rc = AddOS2Program( "MS C ",,, "/k closnew", "cmd.exe" )
    rc = AddOS2Program( "THELP Help/View Facility",,, "/k thlp", "cmd.exe" )

/* Add OS2 SP00L on prog; arguments: title, startup dir, parameters, exe */
    rc = AddOS2Program( "Print Spooler On ",,, "spulon.exe" )

/* Add OS2 SP00L off prog; arguments: title, startup dir, parameters, exe */
    rc = AddOS2Program( "Print Spooler Off ",,, "spuloff.exe" )

/* Add shutdown option; arguments: title, completion msg */
    rc = AddShutdown( "Shutdown", "Shutdown Complete" )

return 0

```

Figure 10. XXDOS.CMD File

the full Workplace Shell. The names of the jobs selected are placed in the Running Group.

Figure 8 illustrates the TSHELL menu screen after you select three tasks to run.

Navigating through and highlighting entries in the Running Group is also accomplished using the up/down arrow keys.

The Alt-Esc key sequence is used to switch the active focus among the running tasks.

The left/right arrow keys are used to jump between the Start Group and the Running Group.

Figure 9 lists the contents of the STARTUP.COM file in this scenario, and Figure 10 lists the contents of the XXDOS.COM file in the \USERDIR\TSHELL subdirectory.

In Figure 10, CKMEM and MENUMINI are home-grown applications, but all other files are part of the basic OS/2-TSHELL package.

The TSHELL menu items in Figures 7 and 8 are fully configurable using the PGMSHELL facility. This includes the optional renaming of the Start and Running Groups.

All of the functionality of TSHELL, PGMSHELL, and STARTDOS is detailed in the documentation for each of these packages.

## Test Scenario

The bare-bones system configuration that I tested had 3 MB of RAM, and my own memory exerciser/analyzer showed that well over 1 MB of RAM was still available for use after the OS/2-TSHELL combination was loaded.

I booted up OS/2 on a 3 MB RAM, 20 MHz 386DX system, with the swapsize initially

set for 1 MB. I ran my own memory-checker (CKMEM), which shows swapsize and available memory on the fly. I started two OS/2 and two DOS sessions, each one running TINYED. At this point there was no degradation in performance, but the swapsize grew to 2 MB. Next I started two more OS/2 and two more DOS sessions, again all running TINYED; the swapsize remained at 2 MB, still with no degradation in performance.

I had the following tasks running:

- TSHELL
- CKMEM (my memory checker)
- 4 DOS sessions, all running TINYED
- 4 OS/2 sessions, all running TINYED

That's a total of 10 sessions running on a 3 MB system with a swapper size of 2 MB. The system experienced no degradation in performance, and the task switching and navigation was as snappy as you could want.

After my initial tests, I tested TSHELL with the OS/2 for Windows product. TSHELL was also extremely impressive in that environment. The only programs that I exercised were some of the applications that come with Windows—Solitaire, Calendar, Calculator, and so on. I was able to start multiple Windows 3.1 sessions, again with no severe impact on the available memory or swapsize, and task switching was relatively smooth.

## Unparalleled Technical Excellence

I concluded that OS/2 is well suited for computer systems that, by today's standards, are slow and have low memory, and that the performance of the OS/2-TSHELL combination refutes the often-heard contention that OS/2 requires investment in a fast, high-memory system. Indeed, the OS/2-TSHELL combination establishes a new (lower) measure of basic system resources for OS/2.

The OS/2-TSHELL combination provides unparalleled technical excellence for users who want to jump aboard the OS/2 bandwagon but who have limited computer resources. Positioned in this way, the OS/2-TSHELL combination delivers a quality product with the customer's true interests at heart. Users who use the OS/2-TSHELL combination to get started should remember which vendor treated them fairly to begin with and should stick with that vendor as they grow. In this sense, IBM has garnered the respect of the operating system marketplace and should reap the benefits of a continuing, constantly expanding role as the preferred supplier of operating systems.

## Acknowledgments

I offer a note of appreciation to Monte Copeland, IBM Boca Raton, and to all of the other fine IBM folks who were instrumental in helping to bring TSHELL and its facilities to the public. Not enough can be said to do justice to the contributions that Monte and the others have made to the OS/2 community of users. Good job, folks!

**Bill O'Connor** is chief consultant for Executive Computer Systems, Toms River, New Jersey. He has more than 30 years of progressively responsible experience with business machines, including programming Eastern Airlines' first reservation system; 12 years with Burroughs Corp. as a technical representative; and almost 20 years of independent consulting. Since the early 1980s, Bill has consulted on right-sizing from mainframe to networked micro-computer solutions. He has a BS degree in accounting from St. Peter's College, Jersey City, NJ, and can be reached at:

Executive Computer Systems  
290 West Water Street  
Toms River, NJ 08753

Direct your questions and comments about this article to [psts@vnet.ibm.com](mailto:psts@vnet.ibm.com).

# Extended Attributes for Files

*This article discusses hints and tips for using extended attributes in your programming projects. It is an excerpt from Chapter 7 of OS/2 2.1 Application Programmer's Guide, by Jody Kelly, Craig Swearingen, Dawn Bezviner, and Theodore Shrader. The code sample in this article was written by Paul Koenen and Gregory Fox.*

**Y**ou can have a long and happy programming career and never have to bother with extended attributes. But for the intrepid, extended attributes will let you pull off some nifty tricks that will fully repay the trouble of mastering this material.

Using file system application programming interfaces (APIs), an application can create extended attributes that associate information with a file beyond the data that operating systems automatically maintain about files and directories.

---

**Jody Kelly**  
IBM Corporation  
Austin, Texas

---

DOS and all versions of OS/2, for example, maintain Level 1 file information for you. This information includes the name and size of the file as well as the date and time the file was created, most recently

accessed, and most recently written to. Most of this information is displayed when you issue a DIR command on a directory.

A file's extended attributes contain information that can be used by another application, the operating system, or the file system that manages the file. The purpose of extended attributes is one or more of the following items:

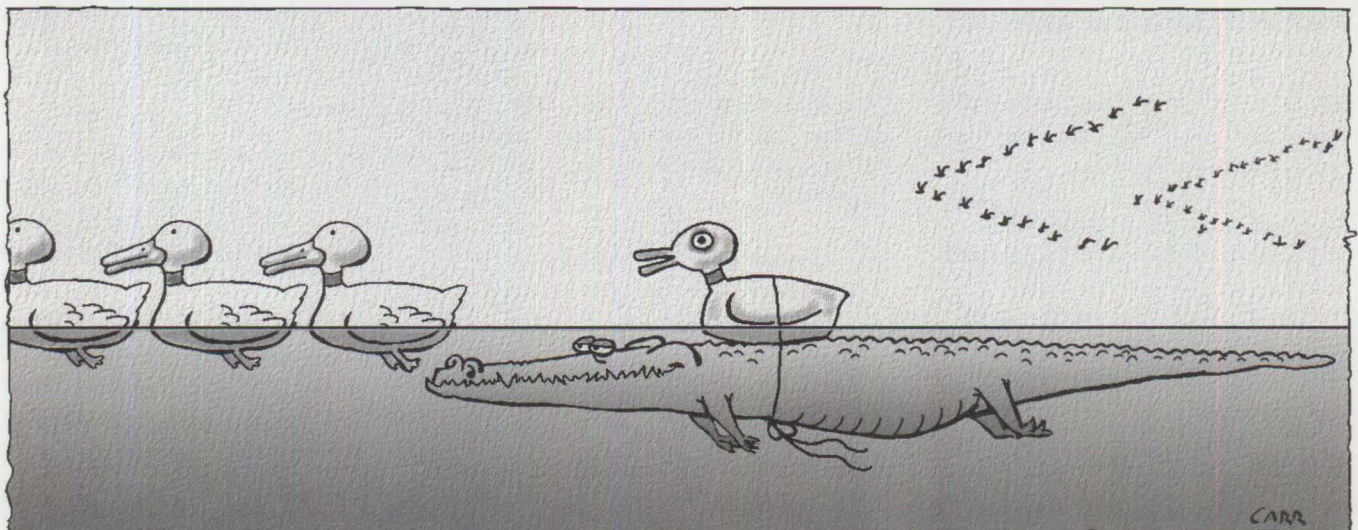
- To make notes about the file, such as the name of its owner or its function.
- To describe the contents of the file, such as text or binary data.

- To explain the data record format of the file.
- To add data to the file; for example, data to be processed only if a certain condition is true.
- To associate an icon with a file.
- To describe the type of the file.
- To associate data files with the applications that create or use the data.

The additional information in extended attributes can include almost anything, such as the file creator's name, the version level, a history log, a contents summary, the icon associated with the file type, a graphic the file uses, and the file's code page.

Extended attributes have been available since OS/2 1.2. However, as long as you used only the File Allocation Table (FAT) file system, the extended attributes for all files in a directory were stored in a separate file with a unique name: EA DATA. SF.

The spaces in the file name and its extension were included to make it harder for someone to erase the file by mistake.



```

Assembler code
BASIC code
Binary data
Bitmap
C code
COBOL code
DOS command file
Dynamic Link Library (DLL)
Executable (EXE)
FORTRAN code
Icon
Metafile (graphics)
Object
Code (compiled code)
OS/2 command file
Pascal code
PIF file
Plain text (ASCII)
Pointer
Printer-specific
Resource file

```

Figure 1. Predefined Values for {name} Associated with .TYPE

```

EAT_ASCII 0025 .TYPE ABC_Co Payroll.C Lab EXE

EAT_ASCII 0025 .TYPE ABC_Co Payroll.C HdQtrs EXE

```

Figure 2. Two Different Entries for {name}

```

ASSOCTABLE assoctable -id
BEGIN
association_name,[extension],[flags],[icon_filename]
.
.
association_name,[extension],[flags],[icon_filename]
.
.
END

```

Figure 3. The Resource File .ASSOCTABLE

The EA DATA. SF file could take up a lot of disk space, and frequent accesses to it could degrade performance.

On partitions formatted for the High-Performance File System (HPFS), extended attributes are still stored separately from the file itself, but each file has its own extended attributes file. HPFS manages the file storage and maintenance automatically.

### Standard Extended Attributes

OS/2 2.1 provides standard extended attributes with pre-defined names for

you to use. These names begin with a period (.). All can contain multiple values (MV), and some can contain multiple types (MT).

The two most often used standard extended attributes (EAT) are .TYPE and .ASSOCTABLE.

#### .TYPE

Format: EAT\_ASCII {length in hex}  
.TYPE {name}

.TYPE is similar to the extension portion of a file name. It indicates the data type

of the file it is associated with and can contain up to 254 characters in HPFS. {name} is case-sensitive and must be unique.

Figure 1 lists the predefined values for {name}. You can also define your own names.

Figure 2 shows how to distinguish between the payroll for laboratory employees and those at company headquarters.

#### .ASSOCTABLE

Format: EAT\_MVMT 0000 0004  
EAT\_ASCII .TYPE {name}  
EAT\_ASCII {file  
extension}  
EAT\_BINARY {flags}  
EAT\_ICON {icon data}

.ASSOCTABLE associates data files with the applications that create or can use these data files. It contains the file type, its extension, its icon, and an ownership flag.

If two applications both use the .ASSOCTABLE extended attribute, the calling application can retrieve this information from the called application. Another use for .ASSOCTABLE is to control, by means of the ownership flag, which application is to run when you double-click on an icon for a particular data file type.

The information that builds .ASSOCTABLE resides in a table in the application's resource file. .ASSOCTABLE can be created by the resource compiler from this table. The resource file .ASSOCTABLE has the format shown in Figure 3, where items in square brackets are optional fields.

In Figure 3:

- The association\_name field is the file type contained in the .TYPE standard extended attribute. The resource compiler must be able to recognize this file type.
- The extension field indicates the file type if a file does not have a .TYPE extended attribute defined.
- The flags field can show which application will be started when the user double-clicks on the icon for a particular data file type.

- If an icon is used to represent this file type, data associated with the icon is included in the `icon_filename` field of `FILEICON`.ICO, the icon file name.

The `flags` field can contain the values specified in the `PMWIN.H` and `PMWIN.INC` files. For example, set the `flags` field to `EAF_DEFAULTOWNER` in the file that should run when the user double-clicks on the icon for a data file type.

For each of the file types that `.TYPE` can contain, only one file should have its flags set to `EAF_DEFAULTOWNER`.

If no specified file type or more than one specified file is set to `EAF_DEFAULTOWNER`, the operating system will not be able to determine which application to run and will display a list from which the user can select.

Another flag setting, `EAF_REUSEICON`, specifies that users employ the same icon as previously defined in the `.ASSOCTABLE` of the resource file.

In Figure 4, the resource file contains the company name, the application name, the file type, the flags, and the icon to uniquely identify the application. As Figure 4 shows, you can join two flags.

Note that if you used a `.TYPE` extended attribute, you would not include the type indicators shown in Figure 4 (“DOC,” “BIN,” “PRT,” “SPR,” and “FOR” in lines 1, 3, 5, 7, and 8).

Figure 4 shows that the payroll application can obtain certain files generated by spreadsheets at the company and the headquarters level—to calculate raises, for example. However, if you clicked on these spreadsheet files, you would not start `Payroll.C Dept_21` because this application is not the default owner of the spreadsheet files.

From the table in the resource file, the `.ASSOCTABLE` standard extended attribute would show which file types the payroll application can recognize and use. Figure 5 lists a sample `.ASSOCTABLE` standard extended attribute.

You can also understand `.ASSOCTABLE` by viewing an Association page on the

```
ASSOCTABLE
BEGIN
"ABC_Co Payroll.C Dept_21 documentation", "DOC",
    EAF_DEFAULTOWNER, Payroll.ICO
"ABC_Co Payroll.C Dept_21 binary data", "BIN",
    EAF_DEFAULTOWNER | EAF_REUSEICON
"ABC_Co Payroll.C Dept_21 print checks", "PRT",
    EAF_DEFAULTOWNER+EAF_REUSEICON
"ABC_Co Co_plan spreadsheet", "SPR", 0
"HDQRTS Corp_plan forecast", "FOR", 0
END
```

Figure 4. Resource File Example

```
EAT_MVMT 0000 0005

EAT_MVMT 0000 0004
EAT_ASCII 0025 ABC_Co Payrl.C Dept_21 documentation
EAT_ASCII 0003 DOC
EAT_BINARY flags
EAT_ASCII icon data

EAT_MVMT 0000 0004
EAT_BINARY 0023 01110001 10101000 11000011 01010001
EAT_ASCII 0003 BIN
EAT_BINARY flags
EAT_ASCII icon data

EAT_MVMT 0000 0004
EAT_ASCII 0025 ABC_Co Payrl.C Dept_21 print checks
EAT_ASCII 0003 PRT
EAT_BINARY flags
EAT_ASCII icon data

EAT_MVMT 0000 0004
EAT_ASCII 001A ABC_Co Co_plan spreadsheet
EAT_ASCII 0003 SPR
EAT_BINARY flags
EAT_ASCII icon data

EAT_MVMT 0000 0004
EAT_ASCII 0019 HDQRTS Corp_plan forecast
EAT_ASCII 0003 FOR
EAT_BINARY flags
EAT_ASCII icon data
```

Figure 5. Sample `.ASSOCTABLE` Standard Extended Attribute

OS/2 2.1 Desktop. Association is one of the items you can select in the Enhanced Editor.

#### Other Extended Attributes

Eight other standard extended attributes are also available.

- `.CODEPAGE` indicates that the code page for the file differs from the code page

for the application or from the system default. The `.CODEPAGE` attribute is usually single-value, single-type.

- `.COMMENTS` stores notes about the file, such as environment restrictions, memory requirements, or other important information. The `.COMMENTS` attribute is usually multi-value and can be any type.

- `.HISTORY` stores changes to the file. The entries must be in a standard format and must use only ASCII characters. To keep the `.HISTORY` attribute from becoming huge, your application can post a message asking the user whether to log this printing of the file.

- `.ICON` specifies the icon that represents the file, such as the icon for the file when it is minimized. Since the `.TYPE` attribute contains an icon field, you could use it to indicate the default icon. However, the `.ICON` attribute is checked first. If it exists, it is used for the physical icon data.

The resource compiler creates the `.ICON` extended attribute by using the keyword `DEFAULTICON` with the file name `{filename.ico}` that contains the associated icon definition. Your application can use either `.ASSOCTABLE` or `DosSetPathInfo` to associate the icon with the file.

- `.KEYPHRASES` associates text with a file that an application or database search routine will look for. Another use for `.KEYPHRASES` is to describe or identify the file. The `.KEYPHRASES` attribute is usually a multi-value, single-type attribute.

- `.LONGNAME` shortens a long name for the purpose of writing the file name to a file system such as FAT that does not support long names. The application should save the long name in the `.LONGNAME` attribute and should notify the user about the new short name.

When an application copies a file from FAT to HPFS, it should check to see if the `.LONGNAME` attribute contains a value. If so, the application should restore the long name and delete the `.LONGNAME` attribute. The `.LONGNAME` attribute is usually single-value.

- `.SUBJECT` summarizes the contents or purpose of the file, up to a maximum of 40 characters. The `.SUBJECT` attribute is usually a single-value ASCII attribute.

- `.VERSION` records the level or version of a file, application, or DLL. The `.VERSION` attribute is usually single-value and can be ASCII or binary.

## The EAOP2 Data Structure

To set or query extended attributes, an application program can call an API,

such as `DosEnumAttribute`, `DosSetFileInfo`, or `DosQueryPathInfo`. An application uses these APIs to fill in the fields in the extended attributes operation data structure (`EAOP2`).

This data structure is required for all operations on extended attributes. `EAOP2` consists of two list structures (`FEA2List` and `GEA2List`) and an error field. The list structures contain one or more name structures (`FEA2` and `GEA2`).

`FEA2List` is a list of full extended attribute data structures. It is used to create or set extended attributes and query them; it contains the total length of the list as well as the list of `FEA2` data structures. Use the `FEA2List` structure with certain APIs to query, delete, change, or add an extended attribute. Fields in `FEA2` are required input to the `DosSetFileInfo` and `DosSetPathInfo` APIs, which create or set extended attributes. These fields are required output from the `DosQueryFileInfo`, `DosQueryPathInfo`, and `DosEnumAttribute` APIs, which query extended attributes.

`FEA2` is the full extended attribute data structure. It comprises one element in a `FEA2List` structure and contains the name and the contents or value of the extended attribute as well as the length of both fields. The name can be any characters legal for the file name. The name field cannot be of zero (0) length. The contents field can be set to 0, which deletes the extended attribute.

`GEA2List` is a list of get extended attribute structures used to return the contents of a set of extended attributes. It contains the total length of the list as well as the list of `GEA2` data structures. Use the `GEA2List` structure with certain APIs to query an extended attribute. Fields in `GEA2List` are required input for APIs that query extended attributes (the APIs are `DosEnumAttribute`, `DosQueryFileInfo`, and `DosQueryPathInfo`).

`GEA2` is the get extended attribute data structure. It comprises one element in a `GEA2List` structure and contains the name and the contents or value of the extended attribute as well as the length of both fields. The name field cannot

be of zero (0) length. The contents field can be set to 0, which deletes the extended attribute.

Syntaxes for `EAOP2`, `FEA2List`, `FEA2`, `GEA2List`, and `GEA2` are shown in Figure 6.

## Defining and Managing Extended Attributes

Use file system APIs to define extended attributes on a file. The following file system APIs operate on extended attributes:

- `DosFindFirst` and `DosFindNext` can search for and return specific extended attributes.
- `DosOpen` can create or open files with specific extended attributes.
- `DosQueryFileInfo`, using a file handle, and `DosQueryPathInfo`, using a file name, can read specific extended attributes. `DosEnumAttribute` can use either the handle or the file name to return information about extended attributes.
- `DosSetFileInfo` and `DosSetPathInfo` can set or change extended attributes.

An application can also define extended attributes for a directory by using the `DosCreateDir` API. You can protect extended attributes so that two or more processes operating on a file at the same time won't leave unexpected results. Two access protection schemes are available:

- Access permission based on a handle is set by the sharing mode of the file associated with the extended attributes. An application can query attributes with `DosQueryFileInfo` if the file is open for reading. An application can set attributes with `DosSetFileInfo` if the file is open for writing.
- Access permission based on a path is set by adding the file to the sharing mode that is established for the length of the access. An application can query attributes with `DosQueryPathInfo` if the file is open for reading and permission is set to `DENY_WRITE`. An application can set extended attributes with `DosSetPathInfo` if the file is open for writing and permission is set to `DENY_READ_WRITE`. An application should use `DosEnumAttribute` only after a file is opened in `DENY_WRITE`.

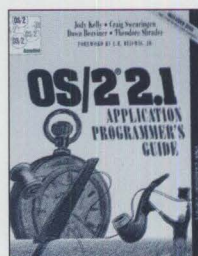
| Structure | Syntax   | Parameters   |
|-----------|--|--|
| EAOP2     | <pre>typedef struct _EAOP2 {     PGEA2LIST pfpGEA2List;     PFEA2LIST pfpFEA2List;     ULONG uloERROR; } EAOP2;</pre>                                | <ul style="list-style-type: none"> <li>■ pfpGEA2List contains the list of GEA2 structures.</li> <li>■ pfpFEA2List contains the list of FEA2 structures.</li> <li>■ uloERROR contains the offset of the FEA error.</li> </ul>   |
| FEA2List  | <pre>typedef struct _FEA2LIST {     ULONG ulcbList;     FEA2 list[1]; } FEA2LIST;</pre>  | <ul style="list-style-type: none"> <li>■ ulcbList contains the total number of bytes in the structure, including the full list.</li> <li>■ list[1] contains the variable-length FEA structures.</li> </ul>   |
| FEA2      | <pre>typedef struct _FEA2 {     ULONG uloNextEntryOffset;     BYTE bfEA;     BYTE bcbName;     USHORT uscbValue;     CHAR chszName[1]; } FEA2;</pre> | <ul style="list-style-type: none"> <li>■ uloNextEntryOffset is the offset to the next entry in the array (FEA2List).</li> <li>■ bfEA contains flags.</li> <li>■ bcbName contains the length of the name, excluding the NULL terminator.</li> <li>■ uscbValue contains the length of the value or content.</li> <li>■ chszName[1] contains the name of the extended attribute.</li> </ul> |
| GEA2List  | <pre>typedef struct _GEA2LIST {     ULONG ulcbList;     GEA2 list[1]; } GEA2LIST;</pre>  | <ul style="list-style-type: none"> <li>■ ulcbList contains the total number of bytes in the structure, including the full list.</li> <li>■ list[1] contains the size of a structure of variable length (GEA2).</li> </ul>  |
| GEA2      | <pre>typedef struct _GEA2 {     ULONG uloNextEntryOffset;     BYTE bcbName;     CHAR chszName[1]; } GEA2;</pre>                                      | <ul style="list-style-type: none"> <li>■ uloNextEntryOffset contains the offset in bytes to the next entry.</li> <li>■ bcbName contains the length of the name, excluding the NULL terminator.</li> <li>■ chszName[1] contains the name of the extended attribute.</li> </ul>  |

Figure 6. Elements of the EAOP2 Data Structure

If another application that has conflicting sharing rights is accessing the extended attributes before your application opens the file, your application will return a non-zero return code.

### Viewing Extended Attributes

The program in Figure 7 lets you view attributes in an ASCII file. The remainder of the program in Figure 7, and a complete program showing how to set



extended attributes on a file, are included in Chapter 7 of *OS/2 2.1 Application Programmer's Guide*, published by Van Nostrand Reinhold, New

York, NY (ISBN 0-442-01736-7). The chapter also shows you how to define your own extended attributes.

**Jody Kelly** is a staff information developer. Currently, she is team lead for Distributed Computing Environment (DCE) publications for AIX, OS/2, and Windows. She holds a PhD in English from Duke University and took 21 hours of math and computer science at the University of Southwestern Louisiana before joining IBM Austin 10 years ago.

```

/*****
/* eaview.c
/* A routine for viewing extended attributes.
/* Prints out the extended attribute name,
/* the size of the EA value,
/* and, with certain attributes, it prints
/* out the value itself given that the
/* attribute is of type EAT_ASCII
/*****

#define INCL_DOS
#define INCL_NOPM
#include <os2.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/* Function prototypes. */
BOOL readEA(char *path);
VOID allocMem(PVOID *pvMessage,
              ULONG ulSize);
VOID walkdir(CHAR *name);
VOID GetEAsFromFile(CHAR *szFilename,
                   DENA2 *eaList,
                   ULONG ulEnumCount);
VOID PrintData(USHORT *pEAData);

/* Defines. */

#define MAX_GEA 500L

/* Begin program. */

int main (int argc, CHAR **argv)
{
    if (argc == 1)          /* Default: current directory. */
        walkdir("");
    else
        while (--argc > 0)
            walkdir(++argv);
    return 0;
}
VOID walkdir(CHAR *name)
{
    APIRET retVal;
    FILEFINDBUF3 buf;
    ULONG srchcnt = 1;
    HDIR hdir=HDIR_SYSTEM;

/*****
/* Use DosFindFirst to find the first file in the
/* directory. Print the filename of the file and call
/* readEA to print the EAs if it is found. Otherwise
/* print an error message.
/*****
if (!(retVal = DosFindFirst(name,
                            &hdir,
                            FILE_NORMAL + FILE_HIDDEN
                            + FILE_SYSTEM + FILE_READONLY,
                            &buf,
                            sizeof(buf),
                            &srchcnt, 1L))) {
    printf("%15s ",buf.achName);
    readEA(buf.achName);

/*****
/* Use DosFindNext to get the rest of the
/* filenames in the directory. Call readEA to
/* get the EAs for that file.
/* Otherwise, print an error message.
/*****
while(!(retVal = DosFindNext(hdir,
                             &buf,
                             sizeof(buf),
                             &srchcnt))) {
    printf("%15s ",buf.achName);
    readEA(buf.achName);
}
} else {
    printf("DosFindFirst failed. rc=%d\n",retVal);
}
return;
}

```

Figure 7. Viewing Extended Attributes



# Developing Lotus Notes Applications

*Chances are you have heard about Lotus Notes recently, either in the press or in your own organization. Notes is a tool that allows people to more effectively share, distribute, and collect information across a computer network. Lotus Notes application programs range from simple to extremely advanced and can be developed by consultants, purchased off the shelf, or written by you or your programming staff.*

*What is the secret to developing your own programs in a Lotus Notes 3.0 environment? Where do you begin? These questions and more are answered in this article, and a sample application is developed in 10 easy steps.*

If you ask five people to define Lotus Notes, you may get more than five different answers—and they will probably all be correct! Notes is very flexible and is unlike any previously existing computer application. There is no reference with which to compare Lotus Notes.

Here are some answers to the question “What is Lotus Notes?” Notes is . . .

- A tool that allows you to share, distribute, and collect information across a computer network. You use it primarily on local area networks (LANs) and wide area networks (WANs).
- A self-contained application development environment. You can write applications customized to your particular requirements.
- A multi-user database system that manages compound documents. Notes supports multiple clients in a client/server environment. The server manages the databases. Inside these databases, you store information in documents, which may contain traditional database fields as well as other rich items such as spreadsheets, dynamic links (dynamic data exchange [DDE] and object linking and embedding [OLE]), images, graphics, and so on.

---

**Doug Northup**  
IBM Corporation  
Raleigh, North Carolina

---

- Electronic mail.
- A forms-routing application.
- A document library.
- A business application.

All these answers are accurate!

## A True Client/Server Application

Notes' architecture includes a true client/server application with two distinct components: client code and server code. On the client workstation, the client code runs the graphical user interface to Notes. The server handles requests from clients, which are received over the LAN, and controls all aspects of the Notes network.

Multiple tasks run concurrently on the server to perform various functions such as database read/write operations, automatic index creation, security control, network communications, statistics monitoring, intelligent agents launching, mail routing, and so on.

The Notes server runs on a native OS/2 computer system; no LAN Server code is required. The server code takes advantage of OS/2's preemptive multitasking, which helps give Notes its excellent performance characteristics, considering the number of responsibilities it assumes.

One way to distribute information and applications across LANs is to use a feature called *replication*. Server-to-server replication occurs automatically when you must run the same application in various locations, such as different buildings, cities, states, or even countries. Figure 1 shows a Notes network with clients and servers distributed across several locations.

Let's say you are running a Notes application at your desk in New York City. Your client calls you with a request, and you update your Notes database with details of the request. How will your peers, located in Dallas, see this new information on their Notes system?

The answer is replication! The New York Notes server can automatically contact the Dallas Notes server, either across the LAN (bridged over a WAN) or by dialing over a modem. At this time, all of the New York updates, changes, and deletions are transmitted to Dallas (and vice versa). The servers are now synchronized with each other so that Dallas and New York each have the same information.

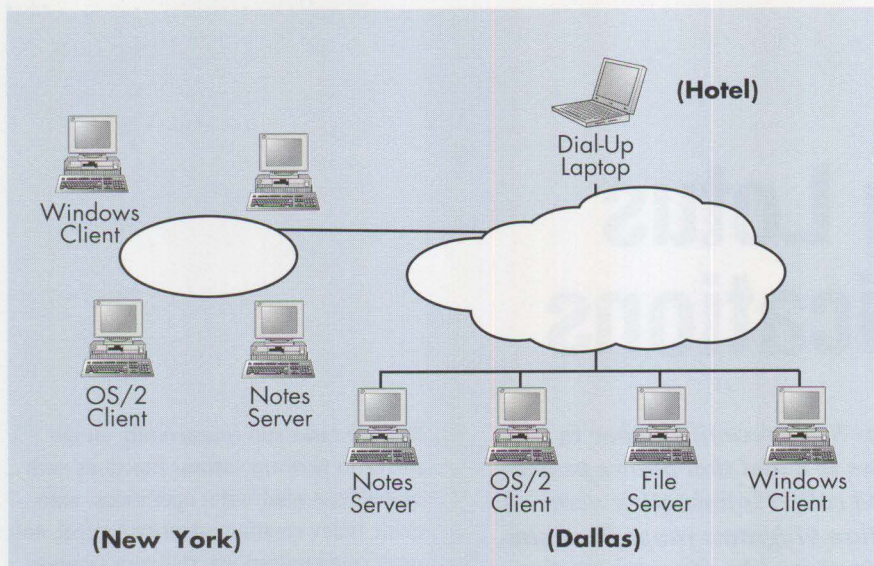


Figure 1. A Lotus Notes Distributed Network

You can also run applications and store data locally on your client computer without using a server. This feature is useful when you are developing new applications, using Notes from home, or using a laptop computer while traveling. If you are traveling, you can update your client information directly on your laptop. Later, just dial in to your New York server from your hotel room and replicate with the server. Now your laptop is synchronized with New York, and your updates flow automatically to Dallas as well. (There is much more to discuss about replication, but it is beyond the scope of this article.)

Lotus Notes contains an integrated word processor. Clients use it to input data. You also use it to develop Notes applications. This gives you the ability to create high-quality documents with effective use of fonts, colors, and formatting.

### Database = Application

Because of the internal architecture of a Notes application, the terms *database* and *application* are synonymous within Notes. When you develop a Notes application, you create all components and store them in a single file with an .NSF extension. This file makes it easy to transport or distribute the entire application.

A Notes database (or application) contains several pieces, or building blocks, that you create as a developer:

- **Form:** A template you create that contains database fields. It usually has fixed text and variable input fields.

You create new database fields inside a form. These fields have a variable name and a type (such as text, numeric, etc.) but do not have a defined length. Imagine a blank 1040 federal income tax form. It contains preprinted text throughout the form and fields for you to input your own data.

- **Document:** A form that you have filled in with data. Continuing the tax form example, this would be your own 1040 form that you have completed with your name, income, deductions, taxes, etc. When you display or edit this document, Notes shows the actual form and data merged together, just as if you were looking at your completed tax return. A document also corresponds to one data record in the database.
- **View:** A list of documents (or records) in the database. A view of 1040 documents could be sequenced alphabetically by name. This view could show several columns of fields on one screen, such as name, social security number, city, state, and income. You can create multiple views within a database, so that you can display information sorted by other fields (such as state or social security number). You can also create views to show other columns (such as exemptions and charitable contributions) or a subset of documents in the database (for example, only those documents with incomes greater than \$100,000). In terms of flexibility, Notes views are much like

views in relational database systems. However, Notes is not a relational database, does not use the SQL language, and is not intended for one-time *ad hoc* queries.

- **Access control list:** Controls security authorization for your application. It allows certain users (or groups of users) to read or edit data in the database. Notes automatically creates a default access control list for you, so you do not have to be concerned about it initially. But this list is a vital component of Notes security, and you should update it before you start sharing your application with others on the server.
- **Macro:** Gives you a programming language to automate keystrokes or execute other tasks inside or outside the Notes environment. This is an advanced feature of Notes programming and acts like a spreadsheet macro.

An important feature in Notes programming allows you to add new database fields at any time. This makes it easy to get started since you can experiment with small databases at first, develop prototypes, and then make additions once you have honed your skills.

### Where to Start Writing Notes Applications

As with most applications, it is important to start with a good design. Many application developers spend more than 50 percent of their time doing design work. The important elements in designing a Notes application are defining database fields, developing the look of the forms, documenting the system, and mapping the flow of information as it gets processed.

The following example is a step-by-step process to create a personal phone book from scratch. For simplicity's sake, the form (or document/record) has two fields:

- **Name:** The person's full name. This is a text field called *Name*. Upper and lower case are respected in Notes, so be consistent when using the field. (If you want to enhance this application, consider designing two fields for first name and last name.)
- **Phone Number:** The person's phone number. This is also a text field called *Phone*. We could choose to make it a numeric field, but the hyphen would

cause problems. (To enhance this, you could separate the area code from the rest of the number.)

Other enhancements might include adding the address, city, state, zip code, and title (Mr., Ms., Dr., etc.).

1. **Create the application.** Remember that a Notes application is equivalent to a database, so to create a new application, you actually create a new database, as shown in Figure 2. Fill in the blanks as indicated. Choosing the server as Local puts your database on your client workstation instead of on the server, which is useful for application development. Then click on the New button. You will be presented with an empty view, because there are no documents in the database yet.

2. **Create the form.** Choose Design - Forms, and click on New. This puts you into the Notes word processor with an empty WYSIWYG (what you see is what you get) form. Type the words "Telephone Listing" at the top, and press Enter twice to provide some white space. If you want to highlight "Telephone Listing," use the mouse as though you were doing a clipboard copy. Then choose Text - Fonts to make this area bold with a different font. You can also choose other colors for this text. Click on OK to return to your new form. Return your cursor to the beginning of the last blank line. Figure 3 shows you how to define this new form.

3. **Create the fields.** On the last blank line type the word "Name:" (including a space), and then choose Design - New Field. (Press OK for "Create field to be used only within this form.") Fill in the field name as "Name" and help description as "Enter person's full name." Leave the type as "Text" and press OK. A box will appear on your form, indicating an input field called Name. Place your cursor to the right of the box, and press Enter to force a new line.

On the next blank line, type the word "Phone Number:" (including a space), and then choose Design - New Field. (Press OK for "Create field to be used only within this form.") Fill in the field name as "Phone" and the help description as "Enter person's

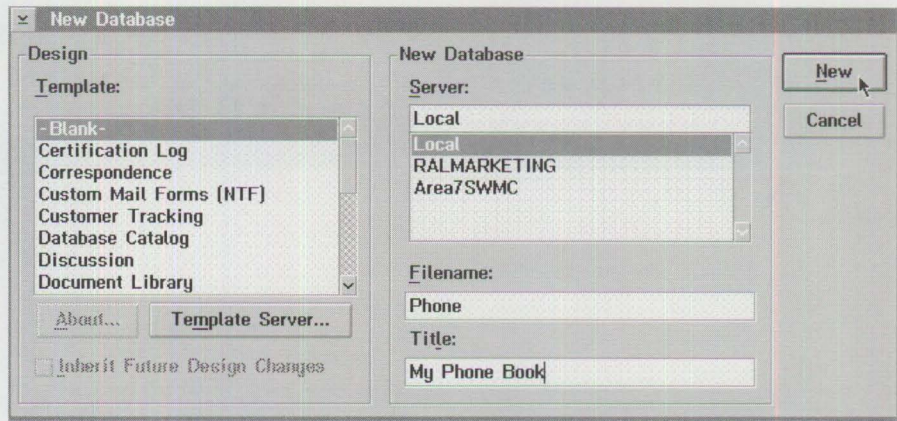


Figure 2. Creating a New Database

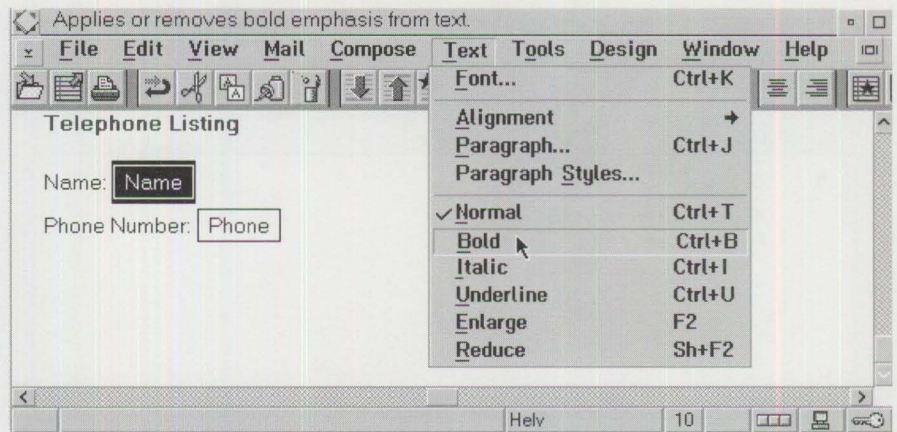


Figure 3. Creating a New Notes Form

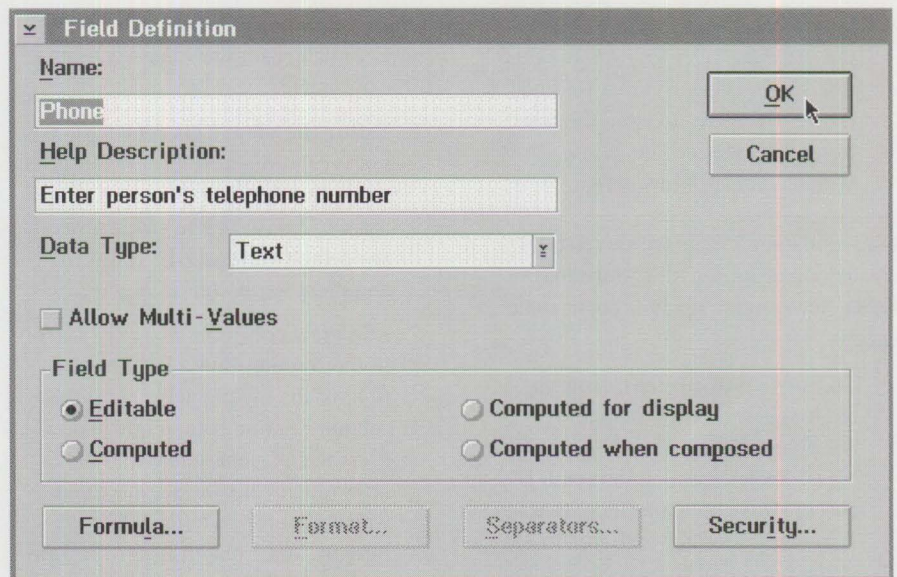


Figure 4. Creating a Telephone Listing

telephone number." Leave the type as "Text" and press OK. A box will appear on your form, indicating an input field called Phone. Figure 4

shows you how to define the phone number.

4. **Save the form.** Before you save the form, you must give it a title. Choose

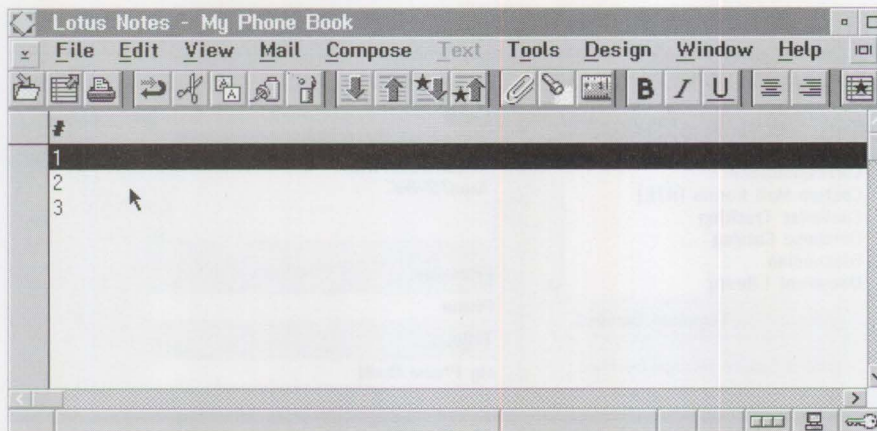


Figure 5. Default View Containing Three Documents

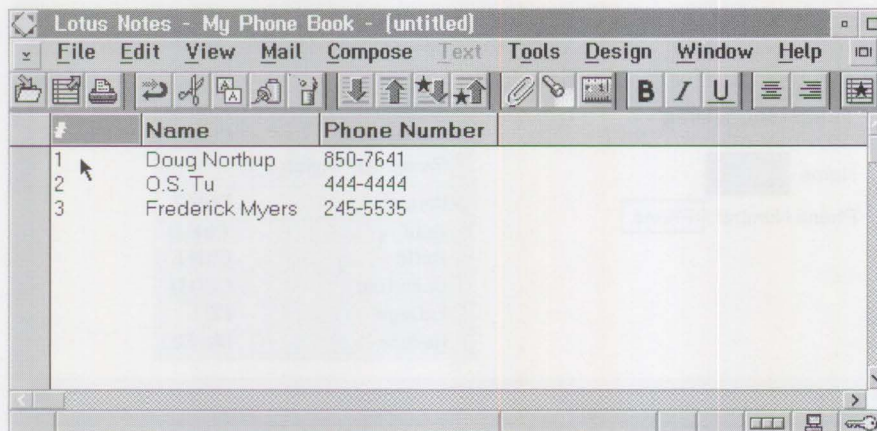


Figure 6. Customized View of Telephone Listings

Design - Form Attributes and enter the name as "Telephone Listing" and type as "Document." Press OK and choose File - Save to store the new form. Next, choose File - Close Window to exit forms design.

Congratulations! You have successfully created a new form with two database fields. Now you are ready to input some data.

5. **Compose a document.** From the empty view, choose Compose - Telephone Listing. You should also set View - Show Field Help on (with a check). This automatically puts your help description for each input field at the bottom of the screen. Fill in your name and telephone number, then choose File - Close Window.

You now have a document in the database, and your view shows document number 1. At this point, compose one or two more Telephone Listing documents with other people's names and phone

numbers (as in step 5 above). Figure 5 shows you a default view that contains three documents.

6. **Tailor the view.** This default view is not very useful as it stands. You need to modify the view to make it more usable. Choose Design - Views, select the default view called "(untitled)," and press Edit to tailor the view.
7. **Add view columns.** Place the cursor in the second column heading to the right of the highlighted # sign (this column heading is currently blank). Press the left mouse button. This action highlights the second column, which is so far undefined. Choose Design - New Column. Enter the Title as "Name." This is the text in the column heading. Enter the formula as "Name" also. This tells Notes to place the field called Name in this column. Press OK to return to your view.
8. **Refresh the view.** Notice the question mark that appears in the upper left corner of the view. This means

the view needs to be refreshed in order to show new data from the changes you made. Click on this question mark with your left mouse button. You now see the names from each of your documents displayed in the second column. Figure 6 shows your customized view at this point.

9. **Add the final column.** Now add the phone number to the view. Do this just as in step 7 above, by placing the cursor in the third column heading to the right of the Name column. This heading is currently blank. Complete the column definition by entering the title as "Phone Number" and the formula as "Phone." Press OK to return to your view. Refresh the view again to see your new phone numbers.
10. **Save the view.** Choose File - Close Window, and Notes will prompt for a name of your view. Enter "Names and Numbers," and press OK. (Refer to Figure 7.)

You have now created a complete Lotus Notes application! This application is a database consisting of forms, documents, and views. An access control list was created automatically for you, and we bypassed any advanced macro processing.

Because Notes databases can be easily expanded after they are created, you may want to develop some more features in your phone book, for example:

- Add address, city, state, and zip code fields
- Create another new view with these additional columns
- Sort your view by name (you'll quickly see why it is advantageous to have separate first and last name fields in the form)

## Are All Applications Suitable for Lotus Notes?

As you progress with any Lotus Notes implementation, be aware of certain characteristics that may influence your selection of Notes applications. Notes is excellent for many applications because it exploits the power of OS/2 and provides unique business solutions for today's environment. Certain types of

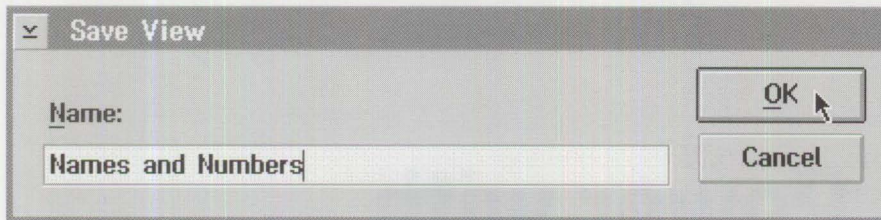


Figure 7. Saving the Customized View

applications, however, do not lend themselves to a distributed LAN application.

Generally, if you run Notes applications and replicate with multiple servers, consider the time delay between replication (or synchronization). For example, what happens if your data in Dallas has been outdated by a recent change in New York and replication has not yet transmitted those changes to your site? Will you make a poor decision based on old data? In most cases, this either will not happen or will not matter. But it could be an issue, depending on the nature of the application. If your application always requires real-time access to all possible data at all locations, then consider writing the application outside of Notes.

Notes is an information sharing system, not a transaction processing system. It relies on the significant horsepower available today on personal computers, but it will not match the speed with which a typical host-based transaction system, such as Customer Information Control System (CICS), can deliver. In other words, if your application requires multiple operators entering data eight hours a day, or if it must process tens of thousands of daily transactions, then consider writing the application outside of Notes.

Also, since Notes is not a transaction-processing system, it does not provide record-level locking. This can be reason for caution if two people update the same document at the same time. Notes

allows this to happen, and one person's changes can be accidentally overwritten. Replication conflicts can also occur if two people in different locations each update the same document between replication events. These chances are slim, and in a properly designed system, these instances can be minimized.

Although Notes databases can potentially become very large, Notes easily handles databases in excess of one gigabyte. You need to think about this when using these applications over dial-up telephone lines. Replicating large databases over a telephone link may take too long, depending on the speed of the connection. A replication feature is available to truncate large documents and remove file attachments, which is particularly helpful for laptop computers.

### Advanced Development Techniques in Notes

Once you become comfortable with Notes, you will want to share your applications with others on the LAN. Be sure to thoroughly test any new applications before sharing them, and add documentation such as help text and a database policy document. Take pride in the presentation of the forms and views by using colors, fonts, and so on, and impress people with your new development skills!

Although Notes is a multi-user system, your applications do not necessarily have to be multi-user. A good example is the personal phone book developed in this article. However, you always have the

option to put your application on the server instead of your local hard disk, and (by adjusting the access control list) open it up to the entire Notes community.

For developers, it is important to understand all application development concepts. There is much more to Notes programming than discussed here. Many advanced techniques can be learned by re-engineering other existing Notes applications. In fact, you rarely create Notes databases from scratch as we did here. Instead, you use an application template that already contains much of the code. Better yet, you can make a copy of an existing application and modify it to suit your needs.

### Reaping the Benefits of Notes

Notes significantly helps you perform your job better and faster than you could before. The benefits of high-quality information are key and often worth the investment in Notes. If you understand how to develop Notes applications and know which ones fit best, chances are you will enjoy it and greatly appreciate this new technology.



**J. Douglas Northup** is an IBM Certified Marketing Specialist in workgroup computing. He has been with IBM for 13 years in Raleigh, North Carolina, and has held positions in sales, systems

engineering, and application development. Doug has specialized in Lotus Notes for more than two years and has a BS degree in Computer Science from Virginia Tech. He can be reached at (919) 850-7641, fax (919) 850-7555, and USB235V at IBMMAIL.

# Conserving Power with Personal System Power Management

*The new generation of portable and desktop systems uses power management techniques to conserve power. This article gives an overview of power management and the implementation of power management support at both the hardware level and operating system level. It also discusses the Advanced Power Management (APM) and the power management support in PC DOS 6.x.*

**P**ower management is essential for extending the battery life of portable (laptop, palmtop, and notebook) systems and for conserving energy in desktop systems.

Power management can be incorporated at the hardware level, BIOS level, or operating system level. These levels of power management can either work independently to produce some power savings, or they can collaborate to produce greater power savings.

## Three Power Management Modes

Power management works by controlling the power to the system based on the system activity.

---

**Pylee Lennil and  
Kamal Patel**  
IBM Corporation  
Boca Raton, Florida

---

In basic power management, a system can run in three different power modes:

- **Full-on.** In this mode, the system works at full power. All devices are powered on, and no power management is done.
- **Standby.** This mode is entered if the system is inactive for a short period. In standby mode, the system runs at low power with some

power savings. Seldom-used peripheral devices, such as the floppy diskette drive or a fax/modem adapter card, are powered off, and the display is powered down (but not off). The processor's clock speed is reduced to conserve additional power. When a system event occurs, such as mouse movement, a keystroke, modem ring, or program execution, power management quickly restores the system to full-on power.

- **Suspend.** This mode suspends the system for further power conservation under the following circumstances:

- The system power manager detects the system being in standby mode for



a certain time, depending on the power management implementation.

- The user hits the suspend button, or the system power manager detects a lid closure from the lid-closing switch.
- The system generates a battery-low signal.

Suspend mode provides maximum power savings by turning off power to the processor, the processor's clock, RAM memory, and all peripheral devices. However, power to the processor's core logic is not turned off. Before entering suspend mode, the system state must be saved; later, when full-on power is restored, power management refers to the system state to restore full power to all suspended components.

A system event—such as a keystroke, lid opening, modem ring, or activity on an input/output channel—restores the system to the full-on state. It may take noticeable time to recover from suspend mode to full-on mode.

### Hardware-Level Power Management

Power management can be incorporated at the hardware level, such as in Intel 386SL and Intel 486SL processors.

Power management at the hardware level can work independently, or it can work in conjunction with power management at the system BIOS and operating system levels to conserve the most power. For example, the Intel 80386SL and 80486SL processors, in conjunction with the Intel 82360SL peripheral controller (shown in Figure 1), efficiently manage the power consumption of personal systems. Power management facilities in the 82360SL include powering down idle peripheral devices, slowing or stopping the clock to the system components, and suspend/resume operations.

An important feature of the SL architecture is a new operating mode called the *system management mode* (SMM). SMM provides a separate execution environment for the power management software, transparent to the operating system and application programs.

A power management event generates a system management interrupt (SMI) that

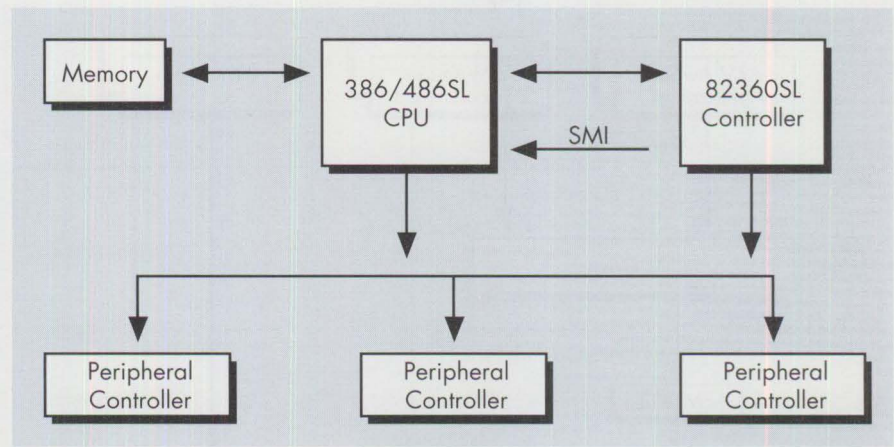


Figure 1. Intel SL Architecture

activates the system management mode. The processor temporarily suspends execution of the operating system or the current application program, then executes the power management software in a dedicated system management RAM address space. After handling the event, the power management software executes a return from system management (RSM) instruction, which restores the processor from SMM to normal mode, then executes the operating system or the application program.

The SL architecture makes it easy to implement the following power management features:

- Detecting idle peripheral devices and powering them down; then powering them up again when they are needed.
- Turning off power to all the peripheral devices and the processor clock; then, when the system is restored to the full-on mode, restoring the clock and the power to the devices without losing the machine state.
- Saving the system state in non-volatile memory and turning off power to all system devices, including the processor and the 82360SL peripheral controller; then restoring the clock and the machine state when the system is restored to full-on mode.

These features enable robust, highly flexible power management. The power management software can run independently of the BIOS and operating systems, and it can dynamically control the processor's clock speed and the power to different components such as memory and

peripherals, based on a particular power management level.

### Five Levels of Power Management

The SL processors and the 82360SL provide five different levels of power management support:

- Full-on
- Local standby
- Global standby
- 5-volt (or 3.3-volt) suspend
- 0-volt suspend

The two standby levels and the two suspend levels are discussed below.

#### Local Standby

In local standby mode, the 82360SL peripheral controller uses timers and device monitors to manage the power for up to six peripheral devices, powering them down if they remain idle for pre-selected periods of time. Seldom-used peripherals, such as a floppy diskette drive, will be powered down, affording some power savings.

Prior to turning off the power to a device, the power management software saves the state of that device by reading the contents of its I/O ports and saving those contents in memory. In the case of write-only I/O ports, the last values written to the ports are saved in the 53 shadow registers provided by the system.

The processor's clock is also stopped for further energy conservation. However, the power to the core system is still on

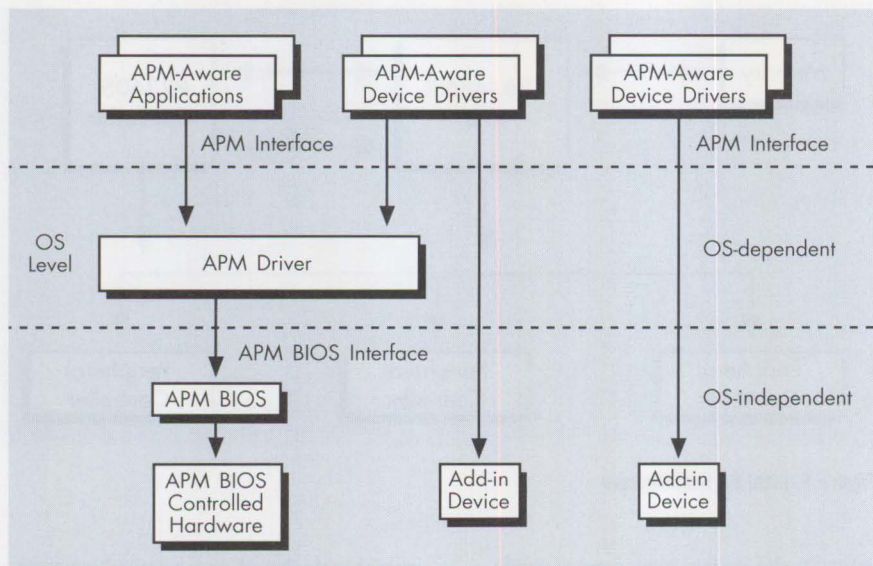


Figure 2. Advanced Power Management System Architecture

and normal operation resumes quickly when a system event such as a keystroke, device interrupt, modem ring, or activity on an I/O channel occurs.

### Global Standby

In global standby mode, a system idling for a long period is powered down even more for further power conservation. This idle-power mode happens when the global standby timer counter expires. (This counter is reset by system events.) When the time-out occurs, it generates a system management interrupt that places the processor in system management mode and executes the power management software. This software turns off the peripheral devices and stops the processor's clock. The power management software then stops executing. Global standby mode provides more power savings than local standby mode.

A system event, such as a device interrupt, modem ring, or activity on an I/O channel, restarts the clock. The execution of power management code resumes, restoring power to the peripheral devices. Finally, the processor executes the RSM instruction to exit SMM and return to normal mode.

### 0- and 5-Volt Suspend

Suspend mode takes place when the system has been inactive for a long period. Suspend mode puts everything in the system, including the core system components, into a low-power mode that can extend the battery life even longer.

Suspend mode can achieve considerable power savings.

The 82360SL offers both 5-volt and 0-volt suspend modes for extending battery life. The major difference is that the power to the processor and the 82360SL is turned off during 0-volt suspend.

Suspend mode can be generated by any of these events:

- Auto power off
- Invoking the suspend button
- Battery-low signal
- Software-generated SMI
- Externally-generated SMI

The auto power off occurs after the system is in the global standby mode for a long period, allowing the system to automatically go into suspend mode. The suspend event can also be invoked when the user presses the suspend button, or by the lid-closing switch, or by a built-in hardware feature wherein the system automatically times out after a certain period of time and goes into suspend mode.

### Advanced Power Management

SL architecture facilitates highly efficient power management that is independent of power management in the operating system and BIOS. However, some conditions are difficult to manage without the operating system's assistance:

- The processor state is difficult to detect without the help of the operating system, especially when the processor is performing intensive computation without any user interaction.
- When the system remains in standby mode for a long time and then a real-time clock interrupt occurs, the system time cannot get updated without the operating system's assistance.

To resolve these drawbacks, Intel and Microsoft jointly defined the Advanced Power Management (APM) interface specification. The APM specification establishes that power management is performed by the operating system in conjunction with the power management hardware and the APM-aware BIOS. The APM software interface specification is a layered environment in which applications, operating systems, device drivers, and the APM BIOS work together to produce a robust, highly flexible power management architecture, as shown in Figure 2.

The APM's objective, which is supported by several personal computer manufacturers, is to control the system's power usage based on system activity. As system activity decreases, APM reduces power to unused system resources until the system is brought into a suspend mode. Another advantage of APM is that it masks the details of the hardware, allowing higher-level software to use APM without any knowledge of the hardware interface.

### APM Components

The major components of the APM architecture are discussed below.

#### APM BIOS

The APM BIOS is the hardware-specific power management software module located on the motherboard. The APM BIOS, supplied by the system manufacturer and specific to the hardware platform, manages power in the background based on device activity.

The APM BIOS is called by the APM device driver, which uses APM BIOS interface calls. It may provide some degree of stand-alone power management functionality without support from the operating system or application software. The stand-alone power management functionality is enhanced once an APM device driver establishes a cooperative connection with



the APM BIOS. This connection establishes a protocol that allows the APM BIOS to communicate power management events to the APM driver.

### APM Driver

This operating system-dependent module cooperates with the APM BIOS to enforce the power management policy. A connection must be established between the APM BIOS and an APM driver for optimum system power management. The APM driver polls the APM BIOS and the APM-aware applications, and device drivers communicate with the APM driver using the APM interface calls.

The APM driver has three primary power management functions:

- Pass calls and information between the application layer and the APM BIOS layer.
- Arbitrate application power management calls in a multitasking environment.
- Identify power-saving opportunities not apparent at the application or BIOS layer.

### APM-Aware Applications

These applications exchange power management information with the APM driver to monitor and control power management through the APM interface.

### APM-Aware Device Drivers

These device drivers provide the power management software interface for add-in devices.

### APM Power Modes

The APM system has five general system power modes: full-on, APM-enabled, APM-standby, APM-suspend, and off. The main difference among the modes is the time required for the system to return to full power. Power consumption and performance are greatest in the full-on mode and decrease with each subsequent mode.

Full-on mode is the default mode when the system is not doing any power management.

In APM-enabled mode, the system is working normally, but unused devices may not be powered.

APM-standby mode is entered if the system is idle for a short period. Recovery

from APM-standby mode to APM-enabled mode appears to be instantaneous.

APM-suspend mode is entered if the system stays idle for a relatively long period. It takes a relatively long time to recover from the APM-suspend mode to the APM-enabled mode.

### APM Support in IBM DOS 5.02 and PC DOS 6.x

APM 1.0 is an industry standard supported in several personal computer systems. IBM DOS 5.02, PC DOS 6.x, and MS-DOS 5.0 or later have an APM driver (POWER.EXE) that supports APM 1.0. POWER.EXE allows the user to choose the power conservation level—minimum, maximum, or regular.

In the APM environment, the POWER.EXE driver works at the software level, and APM BIOS works at the hardware level. Each of these components can work alone to save some power, or they can work together to save more power. Working alone, POWER.EXE can detect periods of system idle, then issue the halt (HLT) instruction to halt the processor and place the system into a reduced power (standby) mode. In both APM-compliant and non-APM-compliant systems, this conserves some power. In an APM-compliant system, the APM driver POWER.EXE and the APM BIOS can share the power management tasks to save more power.

When specified as a device driver in the CONFIG.SYS file, POWER.EXE loads as a device driver. When it is invoked from the command line, it works as a utility to control the device driver. POWER.EXE command options allow the selection of either full power management (POWER.EXE and APM BIOS working in full cooperation), only APM BIOS power management, or no power management at all.

POWER.EXE provides an APM-aware application, a set of INT 2Fh application programming interface (API) calls that comply with the APM 1.0 BIOS specification. These functions are documented in the *DOS 5.02 Technical Reference* (S52G-9932).

When full power management is enabled, system activity is monitored at two different levels. The POWER.EXE

monitors the frequency of certain interrupts while the APM BIOS monitors hardware activity directly. When POWER.EXE starts up and detects APM BIOS, it establishes bi-directional communication between the two.

Either the POWER.EXE software or the APM BIOS may detect a need or opportunity to reduce power. If the APM BIOS makes this determination, it usually posts a message for the software. When the POWER.EXE next polls and finds the message, it broadcasts a power management event to all of the active APM-aware applications and terminate-and-stay-resident (TSR) applications. Each of these receives the interrupt in turn and can act appropriately. For example, an application's action might be to prepare for the change in mode or to reject the change.

Upon return from the interrupt, POWER.EXE checks to see whether any of the handlers has registered a rejection. If no rejection is registered, POWER.EXE sends a message to the APM BIOS telling it to proceed to move into a low-power mode. If a rejection was registered, no message is sent to the APM BIOS, and the system will not go into the low-power mode that was requested.

When POWER.EXE itself detects an idle condition, it does not broadcast an APM event. Instead, it immediately sends a CPU idle call to the APM BIOS. On non-APM systems, POWER.EXE executes a CPU halt instruction, which puts the processor in standby mode until an external event requires servicing. It is up to the APM BIOS whether to change to a lower-power mode upon receipt of this message. The APM BIOS may also switch modes by itself, without querying POWER.EXE. If the APM driver POWER.EXE is not present or is not connected to the APM BIOS, it will switch modes by itself for certain critical power events like a dangerously low battery level. The software is only notified of such a mode switch upon return to the ready state.

POWER.EXE updates the date and time when coming out of a low-power mode. If the system is APM-compliant, POWER.EXE will update the date and time every time the APM BIOS posts a resume message.

## Effect of Power Management on Performance

The power management system detracts from the overall system performance. For instance, slowing the processor's clock speed not only reduces power consumption but also reduces processor throughput. Running power management software also affects performance, because it uses processor cycles.

The effect of the power management system on overall system performance depends on the implementation used. Consider as an example the DOS APM driver, POWER.EXE, which can reduce power consumption when your applications and devices are idle. The syntax is: POWER [ADV [:MAX | REG | MIN] | STD | OFF] where ADV [:MAX | REG | MIN] conserves power when applications and hardware devices are idle. Performance might be affected if an application is active instead of idle. Use MAX for maximum power conservation. Use REG to balance power conservation with application and device performance. Use MIN if the performance of an application or device is not satisfactory when you specify MAX or REG.

STD is a special intermediate level of power management. In this mode, only the firmware level of idle detection is

active. This saves less power than POWER ADV, because it provides more processor time for background tasks. This is useful for background applications that do not receive enough time with full power management active. OFF turns off power management.

## References

Desmond Yuen, *Intel's SL Architecture: Designing Portable Applications*, McGraw Hill, 1993.

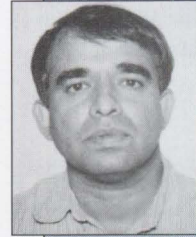
Lorin Ullmann, "Power Management in CRT Displays," *Innovations* (IBM Boca Raton site technical publication), March 1994.

"Technical Update, Part 3: OS/2 2.1 Hardware Support," *Personal Software* magazine, Issue 1, 1994, pages 29 through 31.

*Advanced Power Management BIOS Interface Specification*, revision 1.1, Intel/Microsoft, September 1993.

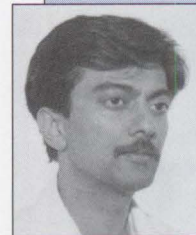
*Advanced Power Management version 1.0: The Next Generation*, Intel/Microsoft.

*DOS 5.02 Technical Reference* (S52G-9932), IBM Corp., 1993.



**Pylee Lennil** is an Advisory Programmer in the IBM Personal Software Products Division in Boca Raton, Florida. He is presently involved in the development of

Workplace OS. Previously, he spent several years in PC DOS, OS/2, and AIX development. Pylee joined IBM in 1983. He earned a BS degree in physics from Kerala University in India and an MS degree in computer engineering from the University of Massachusetts at Lowell.



**Kamal Patel** is a Senior Associate Programmer in the IBM Personal Software Products Division in Boca Raton, Florida. He joined IBM in 1991 and is presently involved in the

development of PC DOS. Kamal received a BS degree in electronics engineering from Bangalore University in India and an MS degree in computer engineering from Florida Atlantic University in Boca Raton. He can be reached via Internet at [kcpatel@vnet.ibm.com](mailto:kcpatel@vnet.ibm.com).

## 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

# Superstor/DS Data Compression in PC DOS 6.x

*The major new feature in IBM PC DOS 6.x is a real-time disk compression capability called Superstor/DS. This article gives an overview of data compression, followed by a detailed description of Superstor/DS.*

**D**isk compression is a technique that increases the storage capacity of disk drives by removing redundant data before it is written to a file. In compression, redundant data bytes are replaced by specially encoded bytes or tokens.

Popular DOS applications such as Stacker and PKZIP perform disk compression as stand-alone programs. In a stand-alone environment, users run these utilities manually to compress and decompress files. PC DOS 6.x disk compression, on the other hand, works in real-time by automatically compressing data while writing to a file and decompressing the data while reading from the file. The PC DOS 6.x disk compression process is transparent to users and application programs.

---

**Pylee Lennil and  
Mark Nosewicz  
IBM Corporation  
Boca Raton, Florida**

---

A compressed file provides two major advantages:

- It occupies less space on the disk, thus increasing the disk's storage capacity.
- It can be more quickly transmitted through a communication network.

For example, a compressed file can be downloaded from an electronic bulletin board faster than a decompressed file.

The efficiency of compression is expressed as a compression ratio, which is the ratio of the number of bits in the decompressed file to the number of bits in the compressed file. The efficiency can also be expressed as the percentage of compression. A 2:1, or 50 percent, compression ratio means the compressed file uses half the disk space it used before compression.

The amount of compression possible depends on the file type. For instance, graphics-image files are easily compressed, achieving a compression ratio of 8:1, while text or database files can be moderately compressed to a ratio of 4:1.

Data can be compressed using two methods, *lossy* and *lossless*. The specific method used depends on the type of data in the file and the compression ratio desired.

## Lossy Compression

As its name implies, with lossy compression, the decompression process cannot fully restore a compressed data stream to its original form. However, lossy compression is faster and provides a higher compression ratio. For instance, it works well with graphics images and audio files because losing a

few pixels or bits may not drastically alter the image's appearance or the sound's quality. But lossy compression is not suitable for data or text files, because even a few missing bits of file information can completely alter the data's meaning or the text's appearance.

## Lossless Compression

Lossless compression ratio is lower than lossy, but the compressed data stream can be restored to its original form by the decompression process.

## Run-Length Encoding

The lossless compression technique can be demonstrated using the run-length encoding (RLE) scheme. RLE works well on files with long strings of repeating bytes. In RLE, blocks of single repeated bit-patterns or bytes are replaced by a single-bit pattern and a count indicating how many times the bit pattern is repeated. During decompression, data is reproduced to its original form by repeating the single-bit pattern the number of times specified by the count.

To demonstrate RLE, assume a 32-byte file with many blocks of repeated single-byte characters (see Figure 1). This file is compressed by replacing repeated single-byte characters by a count and a single byte. Notice that, after compression, the same file contains only eight bytes, which yields a 32:8, or approximately 75 percent, compression ratio.

RLE is most suitable for compressing bit-mapped files, which contain many consecutive occurrences of the single-bit pattern (i.e., many continuous rows of pixels that represent the same color). On the downside, if there is an insufficient number of single-bit patterns, RLE produces a compressed file larger than the

### Uncompressed File

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 44 | 44 | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 | 03 |
| 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 | 05 |

### Compressed File

|    |    |    |    |
|----|----|----|----|
| 07 | 00 | 02 | 44 |
| 10 | 03 | 13 | 05 |

Figure 1. Run-Length Encoding

decompressed version. RLE is not suitable for compressing text or data files, because those files are unlikely to contain many long strings of single repeated character patterns.

### Lempel-Ziv

PC DOS 6.x Superstor/DS and PKZIP use several variations of a dictionary-based lossless compression algorithm known as *Lempel-Ziv* (LZ). LZ works well with text data that has many repeating character patterns, and it provides a compression ratio of approximately 2:1.

LZ is a sliding, dictionary-based algorithm that compresses a file by replacing each repeating pattern with a pointer that refers to the same pattern's previous occurrence and a count specifying the size of the string.

In LZ, the file to be compressed slides through a window. The window consists of a large look-back buffer (the dictionary) containing recently processed text and a smaller look-ahead buffer containing the incoming text not yet processed (see Figure 2). As the input text slides through the window, the look-ahead buffer is examined to see if it contains any character string that matches with any previously processed character string in the look-back buffer. If a match is found, the matching string in the look-ahead buffer is replaced by a token that contains two pieces of information:

- An offset to the identical string in the look-back buffer
- The number of characters in the string

We can illustrate the LZ algorithm by compressing the following quotation by Winston Churchill, which contains the repeating patterns "he farther" and "ward you":

"The farther backward you can look, the farther forward you are likely to see."

As illustrated in Figure 2, the first token 35,11 is generated for the repeated string "he farther" based on a previously processed 11-character identical string "he farther" located 35 bytes to the left. The window then shifts 11 bytes to the right, which is the size of the text just processed, and moves the next 11 bytes to be processed into the look-ahead buffer. This process is repeated for generating subsequent tokens.

During decompression in LZ, tokens are expanded to form the original text. For instance, token 35,11 is replaced by an 11-byte string "he farther" that is located 35 characters to the left. The window is then shifted by 11 bytes to the right, and the process is repeated for expanding the remaining tokens.

Churchill's quotation, which contained 77 bytes in decompressed form, is now reduced to the 61-byte compressed text in Figure 3. Admittedly, this is not very much compression. But, take heed of what the quotation says: "Looking farther backward," and using a larger look-back buffer can yield more compression, because more matching strings are likely to be found in a larger look-back buffer. However, a larger look-back buffer can adversely affect performance, because

many more string comparisons have to be performed against the look-ahead buffer for every string in the look-back buffer.

### Superstor/DS

Superstor/DS is a real-time compression scheme integrated into PC DOS 6.x. It uses several variations of the LZ compression algorithm to compress and decompress data. Compression and decompression are done in the background during disk read and write, completely transparent to the user and to application programs.

Superstor/DS's compression ratio is nearly 2:1, essentially doubling the disk space. The actual disk space savings depends on the content of the file being compressed.

### DBLSPACE.BIN Driver

Superstor/DS compression and decompression is performed by a special driver, DBLSPACE.BIN, which is loaded during PC DOS system configuration. DBLSPACE.BIN is not an ordinary driver in that it cannot be loaded using the DEVICE= statement in the CONFIG.SYS file. Instead, DBLSPACE.BIN is loaded before the CONFIG.SYS file is processed; it becomes a system file like IBMBIO.COM and IBMDOS.COM.

In this scheme, CONFIG.SYS, all the device drivers, and AUTOEXEC.BAT can be kept on the compressed drive. Also, this scheme lets you use your CONFIG.SYS and AUTOEXEC.BAT files without any modifications.

Why are these things beneficial? Older compression techniques required the

| Look-Back Buffer                       | Look-Ahead Buffer | Token |
|--|-------------------|-------|
| The farther backward you can look, the | he farther        | 35,11 |
| er backward you can look, the farther  | ward you          | 34,9  |

Figure 2. Lempel-Ziv (LZ) Encoding

The farther backward you can look, the farther forward you are likely to see

The farther backward you can look, <35,11>for<34,9> are likely to see

Figure 3. The LZ-Encoded Compressed Text

compression driver to be loaded in the CONFIG.SYS file. The CONFIG.SYS and the AUTOEXEC.BAT could not exist on the compressed drive. The user had two sets of configuration files to maintain—one set on the host drive and the other set on the compressed drive. This implementation did not make compression seamless and caused problems for installation programs that needed to modify the configuration files.

During a disk write, PC DOS 6.x calls DBLSPACE.BIN, which compresses each data cluster before writing to the disk. Compression is done at the cluster level. After that, clusters are mapped into associated sectors, and the final compressed data is stored by sector in a location called the sector heap. (Clusters and sectors are discussed in more detail later.) During a disk read, DBLSPACE.BIN reads the sectors associated with each cluster and decompresses the data before passing it to PC DOS.

#### DBLSPACE.SYS Driver

Loading the compression driver DBLSPACE.BIN before processing the CONFIG.SYS file unfortunately means not being able to load DBLSPACE.BIN into upper memory, thus risking losing 18 KB

of precious conventional memory. To alleviate this problem, a special driver called DBLSPACE.SYS moves the compression driver DBLSPACE.BIN from conventional memory to upper memory.

The user has to add DBLSPACE.SYS to the CONFIG.SYS file, as follows:

```
DEVICEHIGH=[drive:][path]
\DBLSPACE.SYS
```

When CONFIG.SYS is processed, DBLSPACE.SYS is installed in the DOS directory with the rest of the DOS utilities and compression utilities.

Using DBLSPACE.SYS also affects the drive-letter assignments, as shown in Figure 4. Note that the drive letter assignments change when DBLSPACE.SYS is loaded.

DBLSPACE.SYS controls drive-letter assignments by setting the final-location bit. This bit signals IBMBIO.COM to hand out the drive letters in DOS 5.0 fashion—every time a block device (such as RAMDRIVE.SYS) is loaded by the CONFIG.SYS file, IBMBIO.COM assigns a drive letter to that block device. When compression enters the picture,

IBMBIO.COM needs to keep calling the compression driver, DBLSPACE.BIN, because DBLSPACE.BIN needs to track the total number of block devices loaded in order to assign the compression drive letters correctly.

For example, suppose that IBMBIO.COM loaded two RAM drives (block devices): block device 1, assigned to drive E:, and block device 2, assigned to drive F:. The DBLSPACE.BIN compression driver keeps track of these two drive assignments, so it knows that the first available drive letter for compression is G:. Then, if DBLSPACE.BIN needs three compression drives, they are assigned to G:, H:, and I:.

After DBLSPACE.SYS is loaded, IBMBIO.COM no longer needs to call the compression driver DBLSPACE.BIN, because DBLSPACE.SYS locks the final location of all the compression drives, and all drives that are required for the compression are assigned. If any more block devices are encountered after DBLSPACE.SYS is loaded, they are assigned a drive letter greater than the last drive letter used for compression.

#### Compressed Volume File

When DBLSPACE.BIN writes compressed

### CONFIG.SYS without DBLSPACE.SYS

DEVICE=RAMDRIVE.SYS

C: Compressed drive

D: RAM drive

E: Host drive

### CONFIG.SYS with DBLSPACE.SYS

DEVICE=DBLSPACE.SYS

DEVICE=RAMDRIVE.SYS

C: Compressed drive

D: Host drive

E: RAM drive

Figure 4. Drive-Letter Assignments With and Without DBLSPACE.SYS

data to a disk, it is stored in a compressed disk that resembles a normal disk partition but has no physical storage medium of its own. Instead, the compressed disk is a special file, called a compressed volume file (CVF), that is kept on a fixed or removable decompressed drive called a host drive (see Figure 5).

The CVF is created and maintained using a special utility program, SSTOR.EXE, with the option PREPARE.

The DBLSPACE.BIN driver mounts the CVF to the root directory of the host drive and assigns a drive letter to the CVF file. For example, after the boot drive is swapped out to what is now called the host drive, a drive letter also must be created for what is now the host drive. This assignment can be viewed in the DBLSPACE.INI file that exists on the host drive. A sample line in DBLSPACE.INI might be ACTIVATEDRIVE=D,CO. In this example, the host drive becomes drive letter D:

As shown in Figure 5, the CVF consists of the following fields:

#### File Allocation Table

The file allocation table (FAT) contains information necessary to locate the file blocks stored in the CVF. Normally a file consists of one or more blocks scattered throughout the CVF's sector heap. Each block is a collection of contiguous clusters, *cluster* being the unit of disk space that PC DOS allocates to a file. Each cluster identifies a set of contiguous sectors in the sector heap.

Unlike a standard decompressed FAT cluster containing 4 sectors per cluster, a CVF cluster contains a variable number of

sectors up to 16. Also, unlike the decompressed FAT cluster, which occupies the entire sector range regardless of the amount of data contained in the cluster, the number of sectors allocated for a cluster in the CVF sector heap varies from 1 to 16, depending on the extent to which the cluster data is compressed.

For instance, a decompressed cluster of 16 sectors compresses to 8 sectors when the compression ratio is 2:1. The range of sectors associated with each cluster, whether it is compressed or decompressed, is recorded in the Microsoft Dblspace File Allocation Table (MDFAT).

#### Sector Heap

All of a file's compressed and decompressed data that is written to the compressed drive is stored in the sector heap. Unlike data in a normal FAT partition, file data in a sector heap is stored in units of sectors rather than in clusters.

During a file write, when a cluster of data is stored, DBLSPACE.BIN searches for free sectors using the free-sector bits in the BitFAT area. A group of sectors is then allocated for the cluster, and the range of allocated sectors is then recorded in the MDFAT.

During a file read, the number of sectors associated with each cluster is obtained from the cluster entry in the MDFAT. The DBLSPACE.BIN driver reads data from the sectors and decompresses each sector before passing it on to PC DOS.

#### MDFAT

The MDFAT area contains a 32-bit entry for each FAT cluster on the compressed drive. Each entry describes the range of sectors (in the sector heap) associated

with a specific cluster and its status. These entries are used for mapping a cluster to sectors during a file read. The bit definition of each entry is given in Figure 6.

#### BitFAT

For every sector in the sector heap, the BitFAT area indicates which contiguous sectors in the sector heap are free and which ones are in use. The BitFAT area is used during sector allocation to store the compressed and decompressed data on the sector heap.

#### Other Fields in the CVF

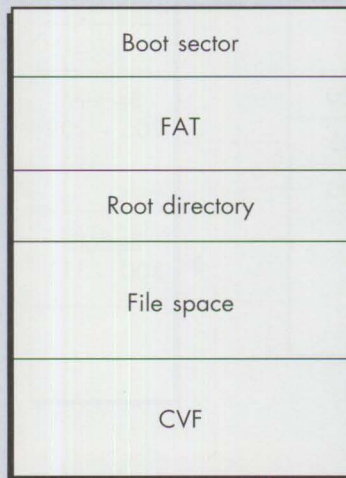
The remaining fields in the CVF are:

- MDBPB, which contains compressed drive-specific information and a standard BIOS parameter block (BPB).
- Boot sector, which contains the standard PC DOS boot sector. It is present only for compatibility; it is not used for booting the computer.
- Root directory, which contains standard PC DOS 32-bit directory entries for all files and subdirectories in the compressed drive's root directory.
- Second stamp, which contains the second Superstor/DS signature. When sector zero is read from a CVF drive, the second-stamp sector is passed to the read. This stamp contains BPB information that is consistent with, and maintains compatibility with, what is expected from reading sector zero on a physical device.

#### Using the CVF

With DBLSPACE.BIN installed, PC DOS 6.x accesses files through DBLSPACE.BIN. During a file write, DBLSPACE.BIN compresses each cluster's data and maps each cluster into a set of contiguous sectors. The cluster's data is then stored in the

### Decompressed Host Drive



### Compressed Volume File

| Entry Name     | Size       |
|----------------|------------|
| MDBPB          | 1 sector   |
| BitFAT         | varies     |
| Reserved       | varies     |
| MDFAT          | varies     |
| Reserved       | varies     |
| Boot sector    | 1 sector   |
| Reserved       | varies     |
| FAT            | varies     |
| Root directory | 32 sectors |
| Reserved       | varies     |
| Sector heap    | varies     |
| 2nd stamp      | varies     |

Figure 5. Structure of Decompressed Host Drive and Compressed Volume File

associated sectors, and the sector bits in the BitFAT are marked as in use. The range of sectors associated with a specific cluster is recorded in the MDFAT cluster entry.

During a file read, PC DOS searches the FAT for the clusters associated with the file data. Using the cluster information in the MDFAT, DBLSPACE.BIN maps each cluster into the associated sectors. The compression driver then reads the associated sectors and decompresses the data before passing it to PC DOS.

#### Example of Cluster-to-Sector Mapping

Cluster-to-sector mapping is illustrated in Figure 7. In Figure 7, the file called FILE.TXT requires three clusters. PC DOS allocates three clusters (75, 76, and 77) in the FAT and records the starting cluster number in the file entry. During file write, PC DOS passes these clusters to DBLSPACE.BIN. It compresses cluster 75 to a compression ratio of 16:9 that requires 9 sectors. It allocates free sectors 120 through 128 using the free-sector information from the BitFAT, and it stores the compressed cluster 75 in sectors 120 through 128 of the sector heap. The range of sectors allocated to cluster 75 is then recorded in the secStart and csecCoded entries in the MDFAT; these sectors are also marked as unavailable in

| Bit(s)  | Handle    | Function   |
|---------|-----------|--|
| 0 - 20  | secStart  | Specifies the first sector number of the sequence of contiguous sectors representing the cluster.  |
| 21      | reserved  | Reserved for future use.   |
| 22 - 25 | csecCoded | A zero-based count that specifies the number of sectors used to store the information for this cluster. The number of sectors depends on the extent to which the cluster is compressed. For instance, a count of 15 means 1:1 (no compression) and a count of 7 means 16:8 (50 percent compression). |
| 26 - 29 | csecPlain | Count specifying the number of sectors required to store the data for this cluster in decompressed format.   |
| 30      | fUncoded  | This bit is set to 0 if the data in this cluster is stored as compressed, and 1 if the data in this cluster is stored as decompressed.   |
| 31      | fUsed     | This bit is set to 1 if the MDFAT entry is in use, or 0 if the MDFAT entry is not in use.  |

Figure 6. Bits in an MDFAT Entry

the BitFAT. This process is repeated for clusters 76 and 77 by allocating sectors 205 through 209 for cluster 76 and sectors 300 through 311 for cluster 77.

During the file read, PC DOS gets the file's starting cluster 75 from the file entry. Using the starting cluster, it obtains the remaining clusters from the FAT. The clusters are then passed to the DBLSPACE.BIN driver. In turn, DBLSPACE.BIN maps each

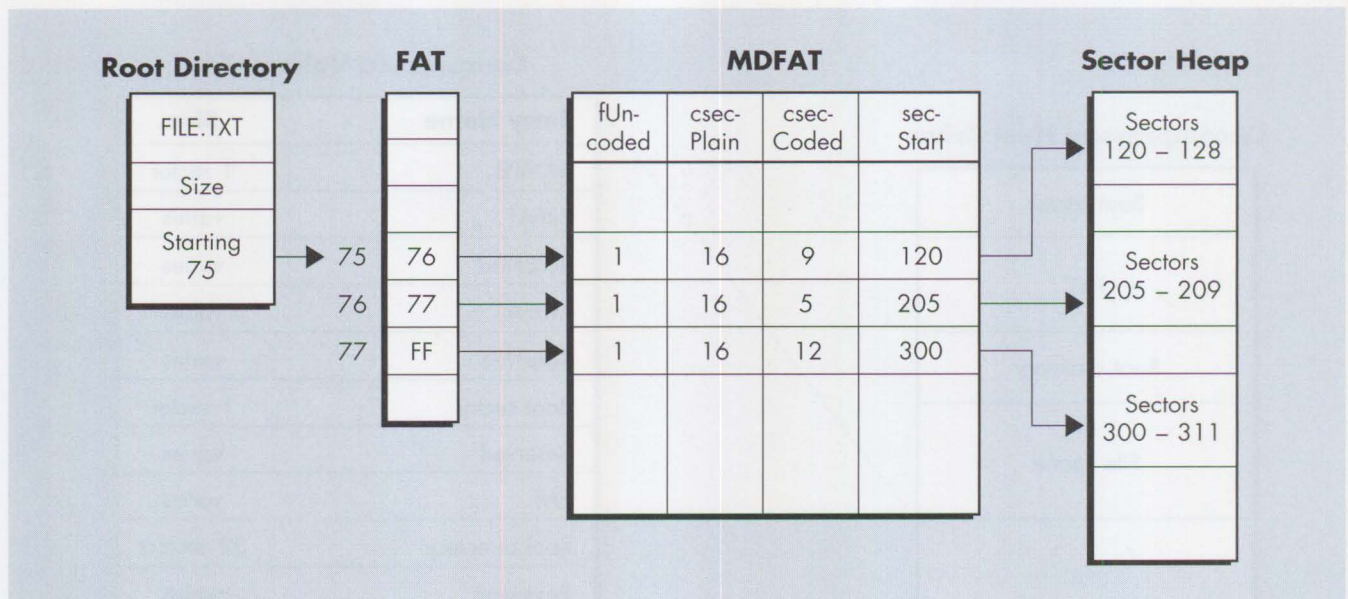


Figure 7. Mapping a FAT Cluster to Sectors in the Sector Heap

cluster to a set of sectors using the secStart and csecCoded entries in the MDFAT. For instance, cluster 75 is mapped to sectors 120 through 128. The sectors are then read and decompressed to 16-sector data before being sent to PC DOS.

### The DBLSPACE API

DBLSPACE.BIN provides a set of functions called the DBLSPACE application program interface (API). This API was defined by Microsoft for their DoubleSpace compression, and it is emulated in the PC DOS implementation of SuperStor/DS. Generally used only by PC DOS disk utilities such as CHKDSK, FDISK, and SYS, the DBLSPACE API functions are used for configuring and manipulating a SuperStor/DS disk drive.

### DBLSPACE Tools

PC DOS 6.x comes with several compression tools for creating a CVF as well as for making fixes in the event a problem occurs in the CVF. These tools are:

- SSTOR, which creates and updates the CVF
- SSUTIL, a full-screen program for repairing and defragmenting a CVF
- RTOOL, which inspects and repairs problems in the same manner as CHKDSK
- SSUNCOMP, which deinstalls the compression feature by restoring the system to the status it had before the compression software was installed

### DBLSPACE Compatibility

PC DOS 6.x compression is 100-percent compatible with MS-DOS 6.0 compression. Users can safely migrate data compressed by MS-DOS 6.0 systems to PC DOS 6.x. Conversely, computers running MS-DOS 6.0 can read and write data compressed by PC DOS 6.x.

In addition, you can transport compressed diskettes to computers that do not have the compression driver DBLSPACE.BIN loaded. To support such an environment, PC DOS 6.x has a Universal Data Exchange (UDE) function. When you create a compressed diskette using the SSTOR utility, you are prompted to include the files UDEON.COM and UDEOFF.COM on the diskette. UDEON.COM is a terminate and stay resident (TSR) program that loads the CVF read/write capability into the computer that is running DOS without compression. Conversely, after you finish writing or reading the compressed diskette, UDEOFF.COM removes the UDEON TSR from memory.

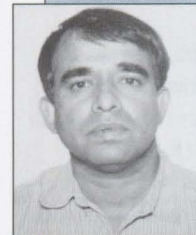
### References

- DoubleSpace Compressed Volume Overview*, Version 1.00.01, Microsoft Corp., 1992-93.
- DoubleSpace System API Specification*, Version 1.00.01, Microsoft Corp., 1993.

Mark Nelson, *The Data Compression Book*, M&T Books, 1992.

*PC DOS 6.1 Data Compression Guide*, IBM Corp., 1993.

Jeff Proisie, "Understanding Data Compression," *PC Magazine*, May 25, 1993



**Pylee Lennil** is an advisory programmer in the IBM Personal Software Products Division in Boca Raton, Florida. He is presently on the Workplace OS development team. Previously, he spent several years in PC DOS, OS/2, and AIX development. Pylee joined IBM in 1983. He earned a BS in physics from Kerala University in India and an MS in computer engineering from the University of Massachusetts at Lowell.



**Mark Nosewicz** is a senior associate programmer in the IBM Personal Software Products Division in Boca Raton, Florida. Mark joined IBM in 1982. He led the development of compression for IBM PC DOS 6.1, and he is currently working on a development project for the Entry Operating Systems and Extensions area. Mark graduated from Palm Beach Community College.



# LAN NetView Object Registration Services

*The IBM LAN NetView family of products provides enterprise LAN customers with systems management solutions. It uses Open Systems Interconnection (OSI) Managing and the Managed System Model, which uses object orientation to represent resources on the managed workstations. A process called object registration allows the managing system to know which objects are installed on the managed workstations. The Object Registration Services (ORS) component of LAN NetView provides the support for any object's definition. In this article, we take a closer look at the ORS database and the registration process.*

The Object Registration Services (ORS) database stores all the object definitions known by a particular system. The ORS database resides on both the managing and the managed workstations, as illustrated in Figure 1. Objects are registered in the ORS database when the installation process first installs the LAN NetView code on a workstation.

To view the contents of the ORS database, issue the `ORSUTIL -d` command. To capture the output in a file, use the `ORSUTIL -d > {file_name}` command.

---

Craig Elliott and  
Alice Turlington  
IBM Corporation  
Roanoke, Texas

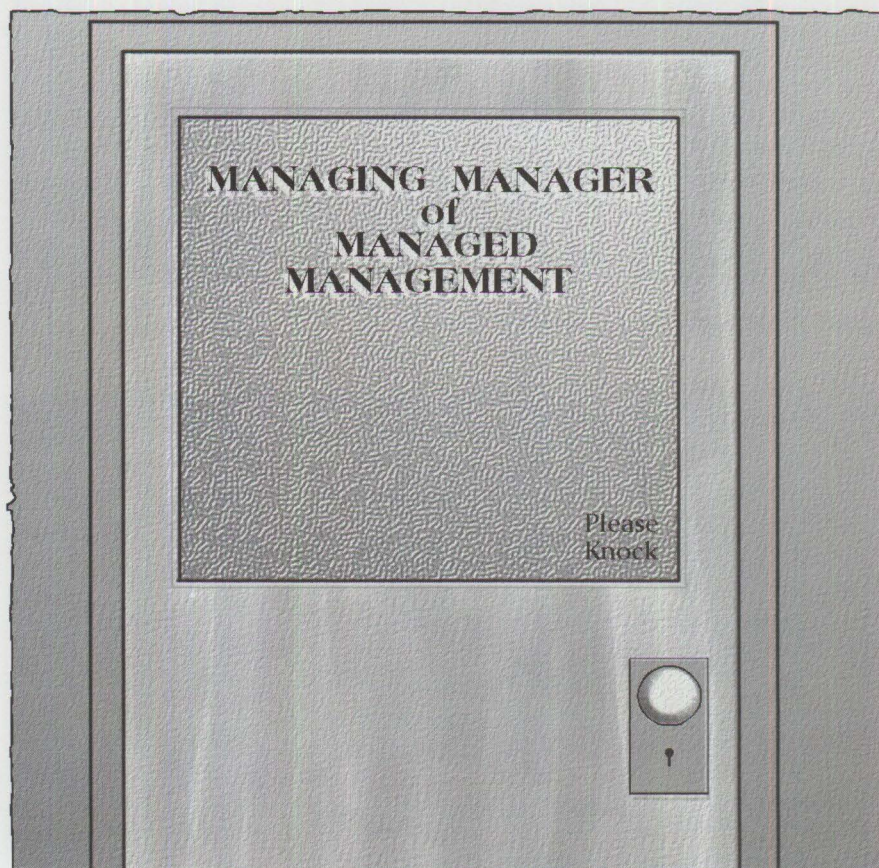
---

A partial example of the ORS file's contents is shown in Figure 2: the entries are listed sequentially by their entry number, and the number of entries depends on the agents installed on the workstation.

## Fields in the ORS Database

Figure 2's ORS database contains 25 entries. Each entry in the database begins with an entry number and the time the entry was created.

The next field in each entry is the object class field. In entry 25, the object class, displayed in dot notation as a string, is 1.3.18.0.0.3315.1.3.26. The long number in this string is called an *object identifier*. Every object identifier is defined by the Open Systems Interconnection (OSI) standards and represents a predefined object class. The translation between the object identifier and the object class it represents is found in the Managed Object Catalog (MOCat) for each of the installed agents. In Figure 2, the object manager name for this entry is `os2agent`, so its dot notation and its represented object class are documented in the OS/2 Agent MOCat. Looking in this catalog, we see that the object identifier number



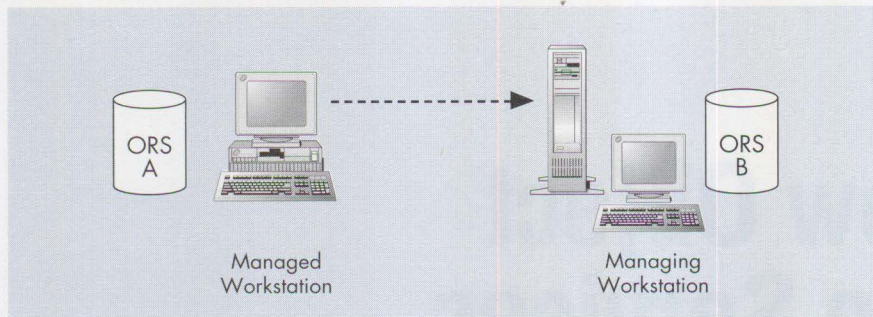


Figure 1. Where the ORS Database Resides

```

ORSUTIL: The database contains the following entries:
Entry # 1:
Time added:      Wed Feb 23, 1994, 14:32:15
Object class:    1.3.6.1.4.1.11.2.2.2.2
Object instance: 0.0.13.3100.0.7.30=pscc/2.9.3.2.7.4=alice/*
Operation:      YYYYYYYNNY
Protocol stack:  OVs_ACMOL
Quality of service: OVs_CMOL_LLC
Object manager name: esm
Password:
Target host:    pscc.alice
Host name:     pscc.alice
.
.
.
Entry # 25:
Time added:      Wed Feb 23, 1994, 14:32:36
Object class:    1.3.18.0.0.3315.1.3.26
Object instance: 0.0.13.3100.0.7.30=pscc/2.9.3.2.7.4=alice/*
Operation:      YYNYNNYNNY
Protocol stack:  OVs_ACMOL
Quality of service: OVs_CMOL_LLC
Object manager name: os2agent
Password:
Target host:    pscc.alice
Host name:     pscc.alice

Total number of
entries found:   25

```

Figure 2. Partial Contents of a Sample ORS Database

| Position | Operation                  |
|----------|----------------------------|
| 1        | Get                        |
| 2        | Set (confirmed)            |
| 3        | Set (unconfirmed)          |
| 4        | Action (confirmed)         |
| 5        | Action (unconfirmed)       |
| 6        | Create                     |
| 7        | Delete                     |
| 8        | Get-next                   |
| 9        | Cancel-get                 |
| 10       | Event-report (confirmed)   |
| 11       | Event-report (unconfirmed) |

Figure 3. Eleven Operations in the Operation Field

translates into the object class O-Spoiled-Printer. Additional information about the object classes can be found in the various LAN NetView Managed Object Catalogs.

The object instance field defines the system name. In entry 25, 0.0.13.3100.0.7.30 is the network ID, and 2.9.3.2.7.4 is the system ID.

The operation field specifies the operations that the management program can perform on the managed object. There are 11 operations for this field, as listed in Figure 3. If the object supports a specific operation, the entry is Y (for Yes); otherwise, the entry is N (No).

The entries for protocol stack and quality of service point to the same protocol layer, which is either CMOL (common management information protocol [CMIP] over logical link control [LLC]) or CMOT (CMIP over transmission control protocol/internet protocol [TCP/IP]). Use Figure 4 to determine whether you are using CMOL or CMOT.

The object manager name specifies the component of LAN NetView to which this object belongs. In entry 25, the "O-Spoiled-Printer" object belongs to the OS/2 agent.

The password and target host fields are used in the simple network management protocol (SNMP) environment. LAN NetView applications are written to CMIP, not to SNMP; however, SNMP applications can be written to support SNMP-supported devices.

The last field in the Figure 2 entries specifies the host name of the system where the object is stored. For CMOL devices, the host name is the NETWORK\_ID.SYSTEM\_ID. For CMOT devices, the host name is the same as the TCP/IP host name.

For the managing system to manage other systems, it needs to know about the objects that are installed on each of the managed systems. This is accomplished by registering the managed system's object at the managing system (see Figure 5). The registration tells the managing program the location of the managed system's object, as well as what kinds of operations can be performed on that object.

During the registration process shown in Figure 5, the ORS database on the managed workstation is either copied to or reconstructed on the managing system. After the registration is done, the ORS database on the managing system contains a copy of the ORS database from the managed workstation. Also, the managing system can issue commands to perform actions on the managed system. For example, it can issue a directory command on the managed system by using the remote command line.

### Local Registration File

The Local Registration File (.LRF) is a specially formatted ASCII file that contains a component's information within LAN NetView and the managed objects about which the component knows. The .LRF resides on both the managing and managed workstations. By default, it is installed in the \LNV\ETC\LRF subdirectory on both workstations.

The registration process discussed earlier uses the .LRF as input to create the ORS database. There are two sections within the .LRF. The first section describes a component—its name, where its executable code is located, and how to start it. During registration, this section adds the startup information about this component to the LAN NetView startup file (SVSUF). The second section in the .LRF contains the object definitions.

The .LRF plays an important role in the registration of objects in the ORS database. Two of the three registration methods (discussed in the next section) directly use the Local Registration Files.

The LAN NetView agents contain many classes of manageable objects. These manageable objects are defined in the .LRF for the agent. Each entry in the .LRF represents a particular object class that can be managed. A sample .LRF is detailed in Figure 6. Not all of the entries are shown in this figure.

Figure 7 lists all of the Local Registration Files.

### Three Registration Methods

There are three ways to register managed objects with a managing system: SVADDOBJ command, ORSLOAD command, and establishing a master/slave

|                    | CMOL      | CMOT         |
|--------------------|-----------|--------------|
| Protocol Stack     | OVs_ACMOL | OVs_CMOL_LLC |
| Quality of Service | OVs_ACMOT | OVs_CMOT_TCP |

Figure 4. Determining Which Protocol Layer Is Used



Figure 5. Registering the Managed System

```
# IBM LAN NetView OS/2 Agent's Local Registration File
#
os2agent:\nos2agt.exe:ovs_cmol_llc:
#
ovs_yes_start:postmaster,systemagent::ovs_well_behaved::
# NOTE:
# If OVs_WELL_BEHAVED is changed to OVs_DAEMON, the OS/2 agent will
# be limited from performing CREATEs of OS2PROCESS instances, and
# from retrieving certain types of information, such as that from
# displays and serial ports.
#
1.3.18.0.0.3315.1.3.2:0.0.13.3100.0.7.30=pscc/2.9.3.2.7.4=alice/*
:OVs_GETR/OVs_SETC/OVs_ACTC/OVs_CREA/OVs_DELE/OVs_EVTU:OVs_ACMOT:
1.3.18.0.0.3315.1.3.3:0.0.13.3100.0.7.30=pscc/2.9.3.2.7.4=alice/*
:OVs_GETR/OVs_SETC/OVs_ACTC/OVs_CREA/OVs_DELE/OVs_EVTU:OVs_ACMOT:
.
.
.
1.3.18.0.0.3315.1.3.9:0.0.13.3100.0.7.30=pscc/2.9.3.2.7.4=alice/*
:OVs_GETR/OVs_SETC/OVs_ACTC/OVs_CREA/OVs_DELE/OVs_EVTU:OVs_ACMOT:
1.3.18.0.0.3315.1.3.10:0.0.13.3100.0.7.30=pscc/2.9.3.2.7.4=alice/*
:OVs_GETR/OVs_SETC/OVs_ACTC/OVs_CREA/OVs_DELE/OVs_EVTU:OVs_ACMOT:
.
.
.
1.3.18.0.0.3315.1.3.25:0.0.13.3100.0.7.30=pscc/2.9.3.2.7.4=alice/*
:OVs_GETR/OVs_SETC/OVs_ACTC/OVs_CREA/OVs_DELE/OVs_EVTU:OVs_ACMOT:
1.3.18.0.0.3315.1.3.26:0.0.13.3100.0.7.30=pscc/2.9.3.2.7.4=alice/*
:OVs_GETR/OVs_SETC/OVs_ACTC/OVs_CREA/OVs_DELE/OVs_EVTU:OVs_ACMOT:
.
.
.
```

Figure 6. Sample Local Registration File (.LRF)

| .LRF Name    | LAN NetView Component                               |
|--------------|---|
| LNCMOL.LRF   | CMOL discovery agent                                |
| LNCMAGT.LRF  | Communications Manager/2 agent                      |
| LNDBAGT.LRF  | Database Manager agents                             |
| LNOS2AGT.LRF | DOS agent, Windows 3.0 agent, and Windows 3.1 agent |
| LNELM.LRF    | Event log agent                                     |
| LNESM.LRF    | Event sieve agent                                   |
| LNLRAGT.LRF  | LAN Requester agent                                 |
| LNSAGT.LRF   | LAN Server agent                                    |
| LNMETDAT.LRF | Metadata agent                                      |
| LNNETW.LRF   | NetWare discovery                                   |
| LNORS.LRF    | Object registration service                         |
| LNOS2AGT.LRF | OS/2 agent  |
| LNPRFAGT.LRF | Performance agent                                   |
| LNPRFLOG.LRF | Performance agent                                   |
| LNPM.LRF     | Postmaster  |
| LNSYSAGT.LRF | System agent  |
| LNTCPIP.LRF  | TCP/IP discovery                                    |
| LNTOP.LRF    | Topology agent                                      |

Figure 7. List of All Local Registration Files

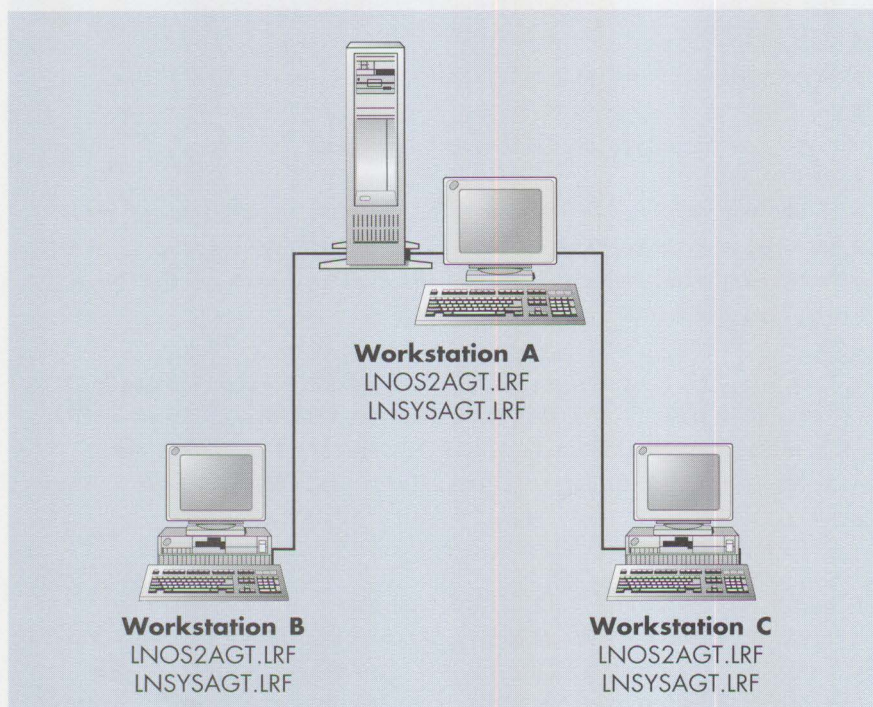


Figure 8. .LRF Locations and Names Prior to Copying

relationship. For each of the three registration methods, this section covers the configuration process followed by a discussion of how to automate the registration.

### SVADDOBJ

The first method of registering an agent with the ORS database uses the SVADDOBJ command. Issuing SVADDOBJ is a manual

process. The input for this command is the name of the .LRF to be registered.

### Configuring for SVADDOBJ

The SVADDOBJ command is used on the managing workstation to register the .LRFs in the managed workstations. Therefore, before issuing the command, all .LRFs must be copied to the managing workstation. But, because all the .LRFs

are installed with the same names for each of the managed workstations, the .LRFs must be renamed before copying to the managing workstation.

For example, in Figure 8, workstation A manages workstations B and C. On each workstation, both the OS/2 and system agents are installed with the workstation-specific information.

As part of the SVADDOBJ process, the .LRFs from workstations B and C are copied to workstation A. However, since workstation A already has files named LNOS2AGT.LRF and LNSYSAGT.LRF, the files being copied from workstations B and C must be renamed with unique file names.

Figure 9 shows the result of copying the .LRFs to the managing workstation, then renaming the copied .LRFs. In Figure 9, the nomenclature of the copied file names on workstation A is:

- LNOS2AGT.LRF and LNSYSAGT.LRF are the two .LRFs that are on workstation A; their names don't change.
- .LRFs from workstation B change from LNOS2AGT.LRF to WBOS2AGT.LRF and from LNSYSAGT.LRF to WBSYSAGT.LRF.
- .LRFs from workstation C change from LNOS2AGT.LRF to WCOS2AGT.LRF and from LNSYSAGT.LRF to WCSYSAGT.LRF.

An SVADDOBJ command is issued for each .LRF that needs to be registered. For example, on a managed workstation, there might be 10 Local Registration Files. In this case, SVADDOBJ will need to be issued 10 times—once for each .LRF. If there are 100 managed workstations each containing 10 .LRFs, then 1,000 SVADDOBJ commands must be issued.

### Automating SVADDOBJ

Using SVADDOBJ to manually register several workstations could take considerable time. An administrator must gather all the Local Registration Files, rename them to unique names, then run the SVADDOBJ command for each file.

There are many ways to automate this process:

- Instead of the administrator walking to each workstation, all of the .LRFs can be copied to a central location during

the configuration/installation/distribution (CID) automated installation process. For example, if LAN Requester is installed and started on the managed workstations, the .LRFs can be copied to the home directory.

- A program can be written to sequentially rename all the files and copy them to a single subdirectory.
- A program can be written to issue the SVADDOBJ command for each .LRF file. An example of this kind of program is shown in Figure 10.

## ORSLOAD

The second method of registering a management program with the ORS database is to use ORSLOAD. Like SVADDOBJ, ORSLOAD is a manual registration process. However, unlike SVADDOBJ, ORSLOAD does not require copying the .LRF file from the local node to the remote managing node. Instead, ORSLOAD uses an existing .LRF file on the managing node and modifies it to specify unique information about the object being registered or deleted. Then this newly created .LRF file is registered with the ORS database on the managing node. (The ORSLOAD method of object registration is documented in the *LAN NetView Manage Administration Guide* [S96F-8492], Chapter 21.)

An example of when ORSLOAD might be used plus steps for configuring ORSLOAD follow.

An OS/2 agent is installed on a managing node, which is also going to be used to manage OS/2 on a node located on another floor in the building. The SVADDOBJ manual method requires walking to the managed node, copying the .LRF file to diskette, walking back to the managing node, copying the .LRF file (using another name) to the managing node, and running SVADDOBJ to register the .LRF file.

However, using ORSLOAD requires only modifying the .LRF file for the OS/2 agent present on the managing node, then registering the new file, which now contains unique information for the managed node.

Run the ORSLOAD command to register objects for remote nodes on the managing node. The ORSLOAD command prompts

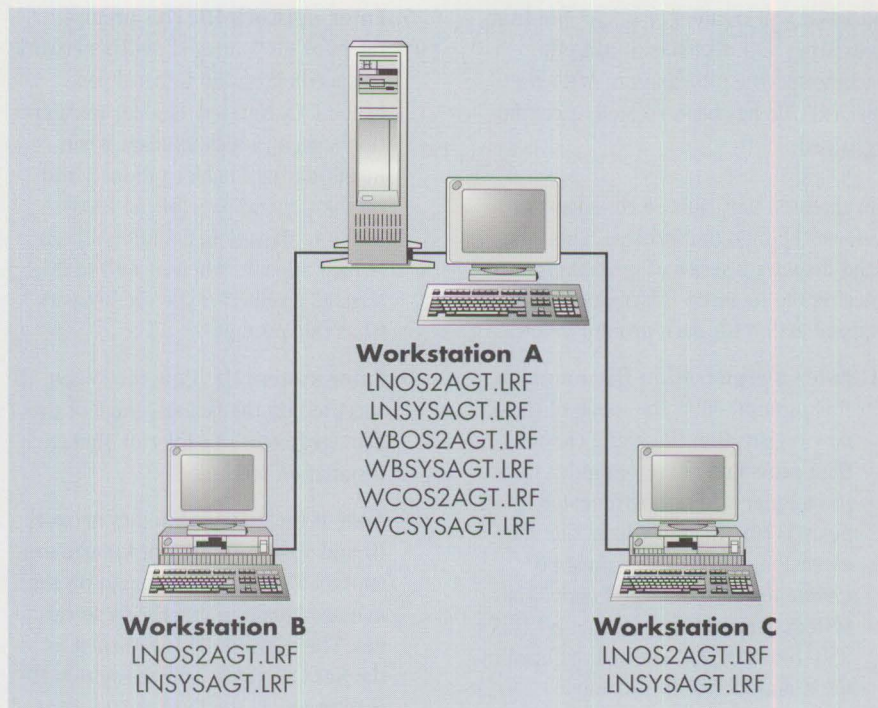


Figure 9. .LRF Names After Being Copied and Renamed

```

/* This is a REXX program to add to the ORS database by using      */
/* the SVADDOBJ command. It takes as input an ASCII file that     */
/* contains the names of all LRF files to be added to the         */
/* ORS database, and it then issues the SVADDOBJ command          */
/* for those LRF files.                                          */
                                                                    */
/* in_file takes the name of the text file containing the         */
/* list of LRF files.                                           */
                                                                    */
in_file = 'lrfall.txt'

/* This loop gets the LRF file name from the input text file    */
/* and then performs the SVADDOBJ command for that LRF file.    */
                                                                    */
do while lines(in_file) > 0
  LRF=linein(in_file)
  SAY "Adding LRF file name = "LRF
  /* Displays the current file being added                       */
                                                                    */
  string="svaddobj "LRF
                                                                    */
  /* Issues the SVADDOBJ command                                */
                                                                    */
  say string
end
exit

```

Figure 10. Program to Automatically Issue the SVADDOBJ Command

you while you create a new .LRF file from an existing .LRF file, substituting the machine-specific information. After the new .LRF file has been created, it can be registered.

Run the ORSLOAD /BUILD command to perform the ORSLOAD process. This command displays a series of prompts for entering the required information. The required action for each prompt follows.

**1. Enter a registration file name.** At this prompt, enter the name of the new registration file being created, then press Enter (the reminder to press Enter will not be repeated below). This file must have the .LRF extension, and its name must be unique for each managed node. The system name is a good name for this .LRF file. WRKSTN.LRF will be used in the remainder of this example.

**2. Enter action ("ADD," "DELETE") - default is "ADD."** If this .LRF file is being created to register management programs, type ADD. LAN NetView defines a *management program* as a collection of software functions used to perform specific types of systems management work on a computer. One or more management programs can assume either a manager or an agent role. If the .LRF file is being created to de-register management programs, type DELETE.

**3. Enter .LRF file name.** This prompt is asking for the name of the .LRF file on the managing node that is being used as a model. Enter LNOS2AGT.LRF for the OS/2 agent. A list of the available management programs and their associated .LRF files is documented in Figure 7 and can also be found in the *LAN NetView Manage Administration Guide* (S96F-8492), pages 31-1 and 31-2.

**4. Enter protocol ("CMOL," "CMOT," "UDP," "SNMP," "").** This prompt is asking you to specify the management protocol. If you are using CMIP over LLC, type CMOL. If you are using CMIP over TCP/IP, type CMOT. If a user-datagram protocol is being used, type UDP. If SNMP over TCP/IP is being used, type SNMP. If the managing node's protocol specified in the .LRF file is the same as the managed node's protocol, then you can accept that value simply by pressing Enter.

**5. Enter network ID.** This prompt refers to the managed node's network ID specified on the General page of the LAN NetView Enabler installation screen (which appears when installing the Enabler product) and on the General page of the LAN NetView Manage installation screen (which appears when installing the Manage product). Type the network ID at this prompt.

**6. Enter system ID.** Type the system ID specified on the General page of the managed node's Enabler or Manage installation screens.

*Note:* If the managed node's network ID and system ID are not known, execute the OVSGETDN command on the managed node to display these values. The network ID is returned as the NetID= value, and the system ID is returned as the SystemID= value.

**7. Enter hostname.** The hostname of the managed node depends on the management protocol being used by the managed node. For CMOL nodes, the hostname is typically a concatenation of the network ID and the system ID: NETWORK.SYSTEM. Enter the command HLMCFGCL GET AGENTINFO at the managed node to display the managed node's name. For a CMOT node, the hostname is the TCP/IP fully qualified hostname in the dotted notation. Enter the TCP/IP command HOSTNAME at the managed node to display the managed node's name.

**8. Enter automatic start behavior ("YES," "NO," "").** After entering the hostname in the new .LRF file, define the operational characteristics of the management program. If the management program is to be started automatically by the SVSTART command, enter YES. If the management program should be started manually, enter NO. If the default specified in the .LRF is the desired option, just press the Enter key. Typically, when registering management programs on a remote managed node, NO should be entered as the automatic start behavior.

**9. Enter start form ("DAEMON," "WINDOW," "").** If the management program is to be started in the background, type DAEMON. If the management program is to be started

in a window, type WINDOW. Some management programs require specific start forms and will not function if another is specified; therefore, you should select the default specified in the .LRF file by just pressing the Enter key.

**10. Is this correct (YES or NO)?** If the displayed information is correct, type YES; if incorrect, type NO. A "no" response will restart the process of specifying the unique workstation information. A "yes" response will complete the current entry.

**11. Continue to construct registration file entries (YES or NO)?** This prompt permits the registration of more than one management program in a single .LRF file. For this example, if you want to register the system agent with the OS/2 agent, enter YES to repeat the process, this time specifying the name of the .LRF file for the system agent (LNSYSAGT.LRT). However, if the OS/2 agent is the only management program to be registered, type NO.

**12. Save the constructed registration file as WRKSTN.LRF (YES or NO)?** To save this new .LRF file, type YES. To exit ORSLOAD without saving the .LRF file, type NO.

**13. Load the constructed registration file WRKSTN.LRF (YES or NO)?** Typing YES will register the management programs specified in the .LRF file with the managing node. Typing NO will exit the ORSLOAD process without registering the management programs. If the new .LRF file is not registered now, it can be registered later by entering the command ORSLOAD filename.LRF. For this example, filename.LRF is WRKSTN.LRF.

### Creating .LRF Files Automatically

Using the ORSLOAD method to register objects for several workstations, each with several management programs, could take a lot of time. However, automating this process can reduce the required time.

The ORSLOAD /BUILD command creates an .LRF file similar to the following:  
add:lnos2agt.LRF::networkid:  
systemid:networkid.systemid:::  
where networkid is the name of the management network, systemid is

the name of the managed system, and `networkid.systemid` is the hostname of the managed workstation. This file can be renamed and edited to change the network ID, system ID, and hostname to reflect another managed node.

This new .LRF file can also be programmatically edited to register the management programs for many managed nodes within a single .LRF file.

The sample program in Figure 11 uses a user-created file called `SYSTEMS.TXT`, which contains the network ID, system ID, and the hostname of managed nodes as its input. The program then creates a sample .LRF file called `SYSTEMS.LRF`, which contains entries that register the OS/2 agent for two managing nodes.

The `SYSTEMS.TXT` file in Figure 11 can be edited to contain all managed systems and register more than one managing program for each of the systems. This entails creating another entry similar to the `os2_string` entry, which is for the OS/2 agent.

The sample `SYSTEMS.TXT` and `SYSTEMS.LRF` files are shown in Figures 12 and 13 respectively.

### Master/Slave

The third, and the only automatic, process of registering management programs residing on a managed node with a managing node is called *master/slave*. When management programs are installed on any system, the objects associated with them are registered in the local ORS database. In the master/slave mode of ORS operation, when a new management program is installed on a slave node, the object registration information is automatically registered in the ORS database on the master node in the master/slave group. This automatic registration enables the managing node to manage the management program on the managed node.

### Configuration Process

When configuring the master/slave mode of ORS operation, one node is designated as the master and the remaining nodes are designated as slaves. In a large environment, more than one node can be designated as a master. However, a node can be a slave of only one master node.

```

/* Program to create an LRF file to register multiple management*/
/* programs from multiple workstations in the ORS database.      */
/*                                                                */
in_file = 'systems.txt'           /* Managed Systems */
out_file = 'systems.LRF'         /* New LRF file    */
/*                                                                */
/* Position pointer at beginning of files                        */
/*                                                                */
temp=linein(in_file,1,0)
temp=lineout(out_file,,1)
/*                                                                */
/* Process each of the systems                                  */
/*                                                                */
do while lines(in_file) > 0
  comment=linein(in_file)
  network_id=linein(in_file)
  system_id=linein(in_file)
  host_name=linein(in_file)
  os2_string="add:lnos2agt.LRF::"network_id":"system_id":
             "host_name":::"
  temp=lineout(out_file_os2_string)
end
/*                                                                */
/* Close the files when finished                                */
temp=lineout(in_file)
temp=lineout(out_file)

```

Figure 11. Sample REXX Program to Create .LRF File

```

***** Bill's system *****
pscc
bill
pscc.bill
***** John's system *****
pscc
john
pscc.john

```

Figure 12. Sample SYSTEMS.TXT File

```

add:lnos2agt.LRF::pscc:bill:pscc.bill:
::
add:lnos2agt.LRF::pscc:john:pscc.john:
::

```

Figure 13. Sample SYSTEMS.LRF File

By default, each node has its own master copy of the ORS database containing the objects associated with the locally installed management programs. To establish the master/slave relationship, a node designates another node as the master, thus making itself a slave. When the

master node is designated, the contents of the local ORS database on the slave workstation are copied to the master ORS database (see Figure 5).

The master doesn't have to be a managing node; it can be a managed node. If the

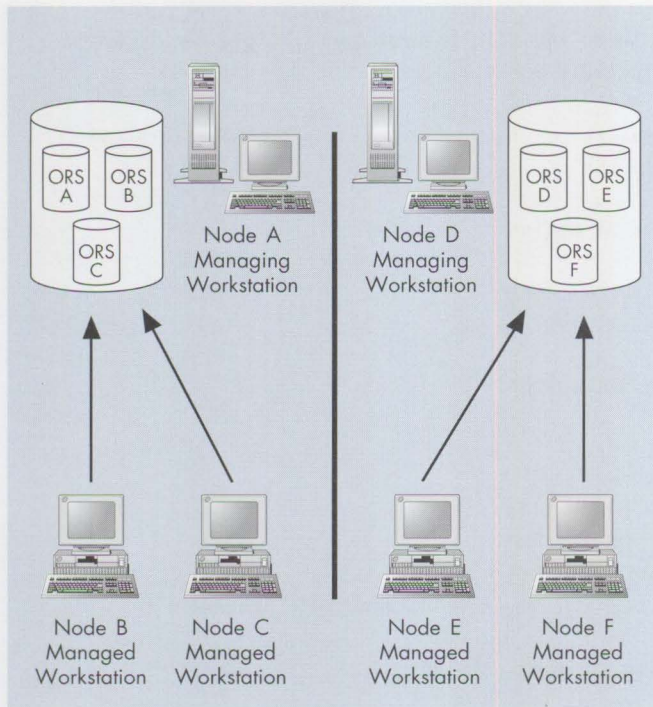


Figure 14. Two Independent Managing Nodes and Their Managed Nodes

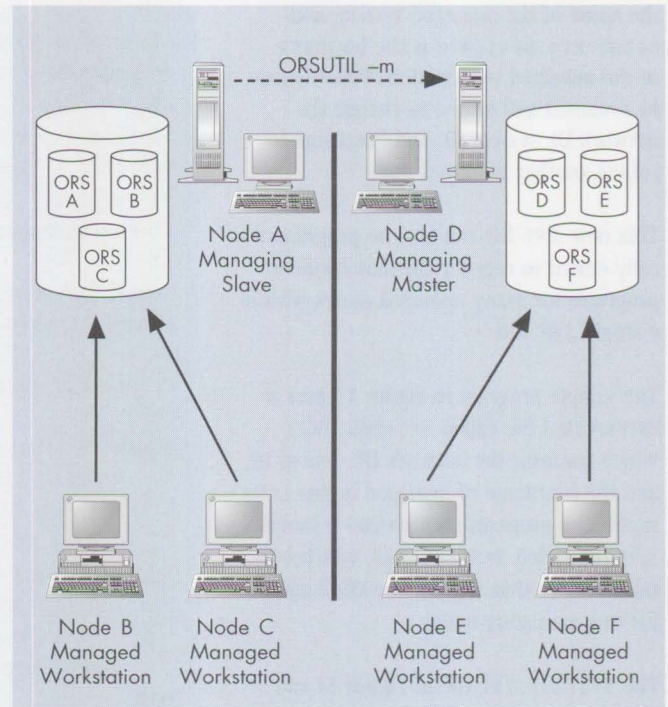


Figure 15. Node A Issues the ORSUTIL Command to Node D

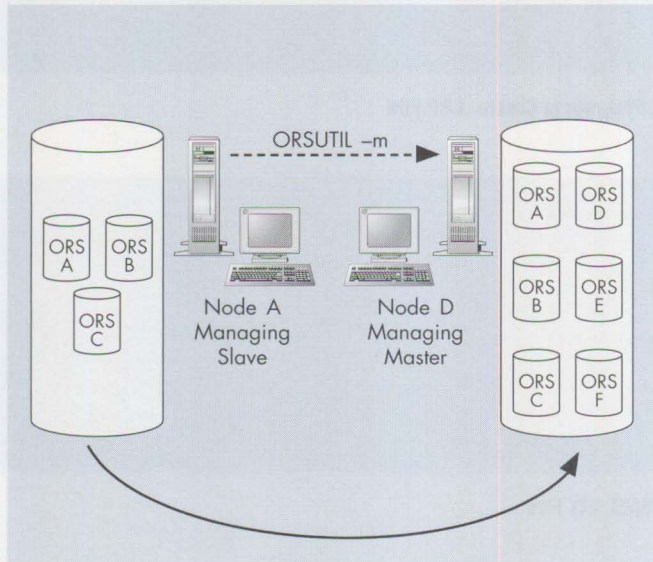


Figure 16. Node A Sends Its ORS Database to Node D

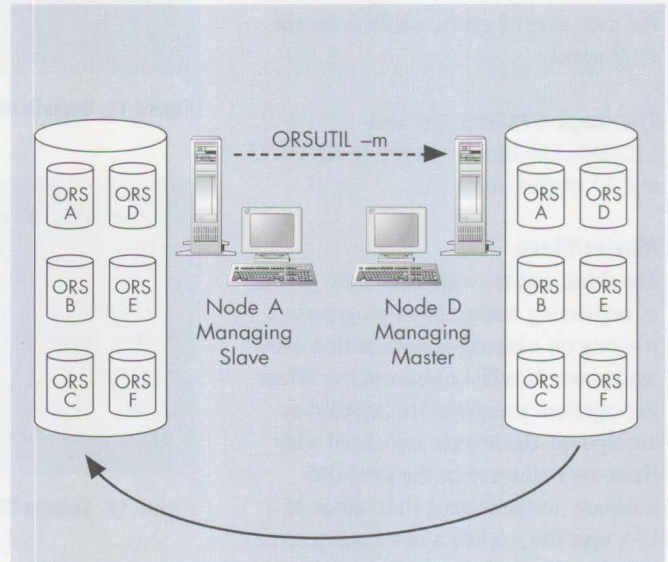


Figure 17. Node D Sends Its ORS Database to Node A

managing node is a slave, once the local ORS database is copied to the master, then a copy of the ORS database on the managed master is copied back to the managing slave. This copying identifies the management programs installed on all slave nodes to the managing node, enabling it to manage these programs.

Typically, a managing system is designated as a slave only when there is more than one managing system. For each

managing system to be able to manage all of the managed systems, the objects from each of the managed systems must be registered in each of the managing systems. Therefore, you can designate one managing system to have the master copy of the ORS database and the other systems to have slave copies of the ORS database.

Because of the way the ORS process works, each of the managing systems will have a duplicate copy of the ORS

database, even though one managing system is the master and the others are slaves. Also, because of the design of the ORS database, a managed system can contain the master copy of the ORS database. Although this is possible, it is not recommended.

To designate a master node, run the `ORSUTIL -m hostname` command on the slave node, where `hostname` is the host name of the master node.



Once the ORSUTIL command is issued, a message displays stating that a connection with the master is being established, the registration process is being performed, and the upload of the ORS database is complete.

The slave node can designate a different master by issuing the ORSUTIL command again, specifying the new master as the hostname. The slave can also become its own master by issuing the ORSUTIL command and entering its own name as the hostname. If a managing master node becomes the slave of another node, all of the managing node's slaves become slaves of the new master, since each node can have only one master and each node can only be either a master or a slave, but not both.

All of these relationships are depicted graphically in Figures 14 through 21, which represent a series of sequential actions. The first set, Figures 14 through 17, represents the master/slave relationship on two managing systems. In Figure 14, Node A and Node D are independent managing workstations. Each managing workstation has its own ORS database containing information about its managed nodes.

In Figure 15, Node A issues the ORSUTIL command to make Node D the master. Therefore, Node A becomes a slave. And, because both Node A and Node D still manage other workstations, Node A is now a managing slave, while Node D is a managing master.

Next, in Figure 16, as part of the ORSUTIL -m process, Node A sends a copy of its ORS database to Node D. Notice that Node D now contains both its own ORS database and the ORS database from Node A.

Finally, in Figure 17, since Node A is also a managing node, Node D sends a copy of its current ORS database to Node A. Notice that Node A now contains the same ORS database as Node D.

The second set of graphs in Figures 18 through 21 represents the process for a managing master node to request another managing node to be the master node. In Figure 18, Node A is a managing master with slave nodes B, C, and D.

In Figure 19, Node A issues the ORSUTIL -m command to make Node E the master node. Therefore, Node A becomes a slave and, because it still manages Nodes B, C, and D, Node A is a managing slave. As part of the ORSUTIL -m process, Node A sends a copy of its ORS database to the new master node, E. Node E now contains both its own ORS database and the ORS database from Node A.

In Figure 20, because Node A, even though a slave, is still a managing node, the ORS database at the managing master Node E is copied to Node A.

Finally, Figure 21 shows that Nodes B, C, and D, which had been slaves to Node A, now become slaves to Node E.

In Figure 21, since Node A now has the same ORS database as Node E, why are Nodes B, C, and D reassigned as slaves of Node E? It is because there can be only one master copy of the ORS database. It is the master's responsibility to keep up with (i.e., manage) the object registration information from all of the slave nodes.

---

*... each node can be either  
a master or a slave,  
but not both.*

---

If that master becomes the slave of another master, all of the slaves of the first master must become slaves of the second master.

### **Restoring the Initial Configuration**

Now that we have gone through several steps to build a series of new relationships among our nodes, how would we undo those relationships and restore the nodes to their original status (as shown in Figure 14)?

We have already noted that if a slave node wants to become its own master, it simply issues the ORSUTIL -m command and specifies its own name. The same is true for either a managing or a managed node. Therefore, to reset everything as it appeared in Figure 14, Managing Slave A would issue the ORSUTIL -m command and make itself its own master. Then each of the

other slaves would issue the ORSUTIL -m command and specify Managing Master A as their master.

Note that doing these things does not remove any of the slave nodes' ORS information from the ORS database on Managing Master E. It simply specifies a new master for this slave node.

The ORSUTIL command is documented in the *LAN NetView Manage Administration Guide* (S96F-8492), pages 33-20 through 33-22. Additional information about the master/slave mode of ORS operation can also be found in the *LAN NetView Manage Administration Guide*.

### **Establishing the Master/Slave Mode of ORS Operation Automatically**

As explained above, manual intervention is required on the slave node to establish the master/slave relationship. This requires the user to have some knowledge of LAN NetView and the master/slave relationship to establish this operating mode. Optionally, the system administrator can visit each slave node and designate it a master. Both options are impractical.

A solution to this inconvenience is to use the remote command-line function of LAN NetView on the managing node to remotely execute the ORSUTIL command on managed nodes. However, before the remote command line is functional, the OS/2 agent and system agent on the managed nodes must be registered with the managing node. To accomplish this, the ORSLOAD process described earlier can be used to register these two agents on the managing node. Then, using remote command-line, the ORSUTIL command can be issued on the managed nodes to establish the master/slave relationship.

Once the master/slave relationship is established, information regarding any management program that is added, updated, or deleted on a slave node is automatically sent to the master node. Additionally, if a managing node is also a slave, these changes will automatically be sent to this managing slave.

### **Deciding Among SVADDOBJ, ORSLOAD, and Master/Slave**

Three different methods of object registration—SVADDOBJ, ORSLOAD, and

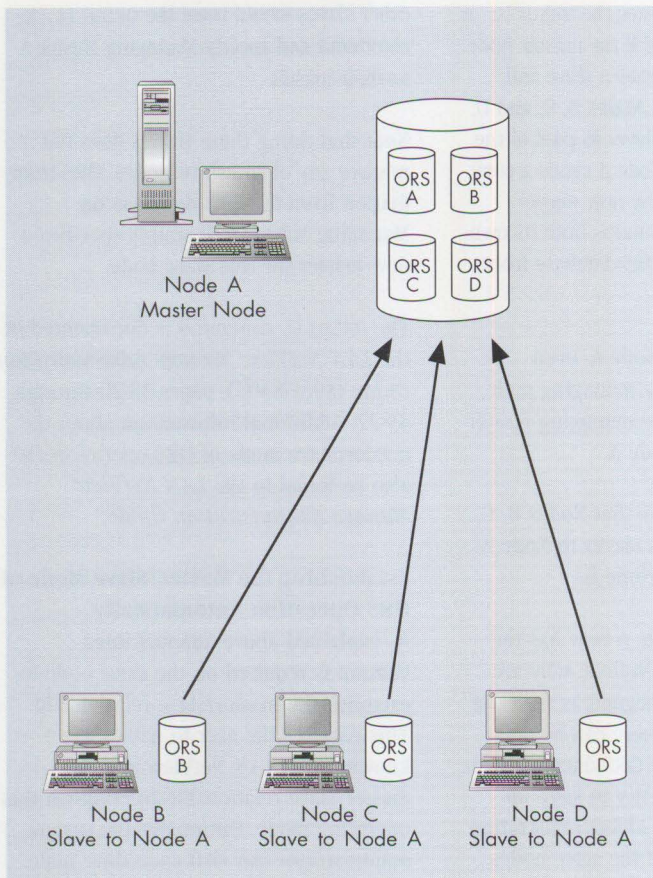


Figure 18. One Managing Node and Its Managed Nodes

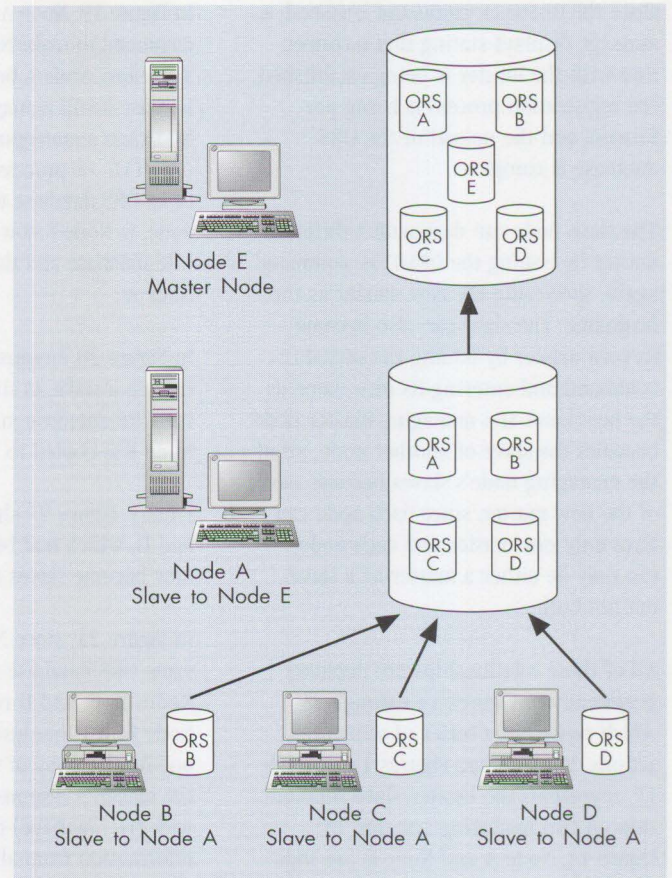


Figure 19. Issuing ORSUTIL and Copying ORS Database to New Master

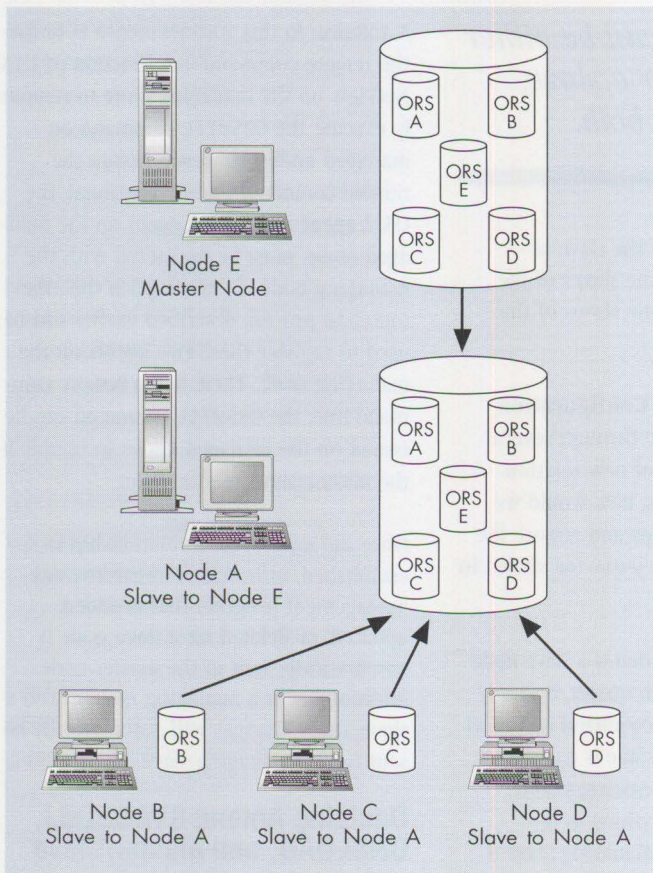


Figure 20. Node E Sends Its ORS Database to Node A

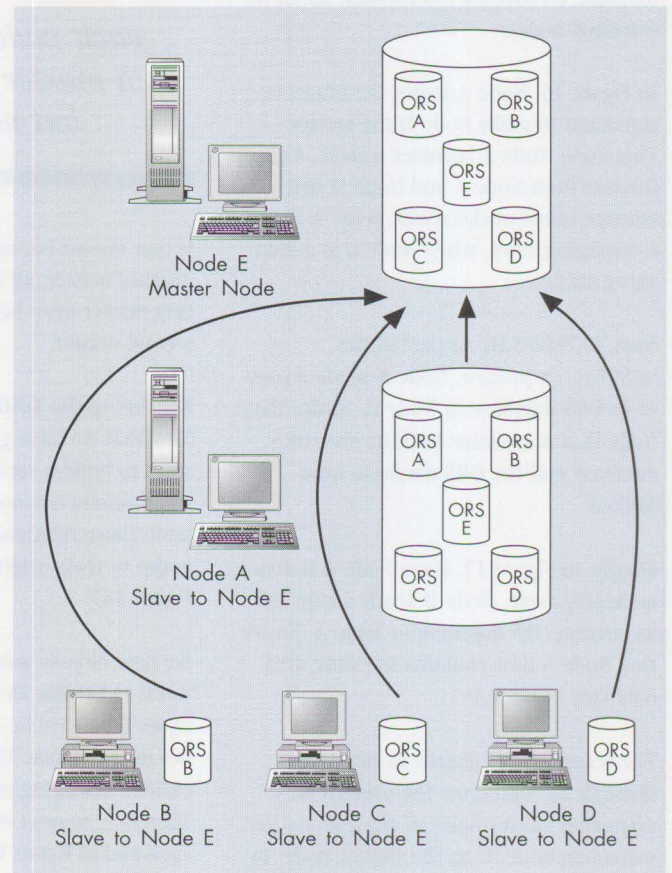


Figure 21. Slaves to Node A Become Slaves to Node E

master/slave—have been discussed. It is up to the individual system administrator to decide which method is appropriate for a specific environment. Each process has some unique characteristics that assist in this decision.

### SVADDOBJ

SVADDOBJ is perhaps the simplest method of object registration. However, it does require copying .LRF files from the managed nodes to the managing node. If there are only a few managed nodes located close to each other, then the inconvenience of having to copy the files may be minimized by the ease of use. Copying files can also be minimized if the workstations are all connected to a LAN server that allows .LRF files to be copied to a central place (perhaps user directories) and registered from there. Another benefit of SVADDOBJ is that only the management programs that are going to be managed are copied to the managing node, reducing the disk requirements of the managing node.

### ORSLOAD

Using ORSLOAD gives the system administrator the benefit of being able to register the management programs without having to visit each managed node to copy the .LRF files. This makes registering

management programs convenient when the workstations are dispersed. With ORSLOAD (as with SVADDOBJ), only the desired management programs are registered on the managing workstation, therefore reducing disk requirements. However, ORSLOAD is still a manual process, requiring the administrator to physically register each management program for each managed system.

### Master/Slave

Master/slave is the only automatic method of registering management programs with a managing node. This is very useful when management programs are being loaded (or updated) periodically, or if the network is very dispersed. Master/slave allows these new programs to be registered automatically. Both SVADDOBJ and ORSLOAD require manual registration each time a new management program is added.

Master/slave is required when managing the Communications Manager/2 agent, since this agent dynamically creates objects as it is executed. Since master/slave registers all management programs on the managed nodes with the managing node, it may require more disk resources than either the SVADDOBJ or ORSLOAD object registration methods.



**Alice Turlington** is an Advisory Marketing Support Representative in OS/2 Systems Support and Services in the IBM Personal Systems Competency Center in Roanoke, Texas.

She leads the OS/2 Systems Management team, which supports OS/2 Systems Management applications. Alice joined IBM in 1987 after earning BS and MS degrees in computer science from the University of Texas at Dallas. She can be reached at IBMMAIL userid USIB523J and Internet userid aturlington@vnet.ibm.com.



**Craig Elliott** is an Associate Marketing Support Representative in OS/2 Systems Support and Services in the IBM Personal Systems Competency Center in Roanoke, Texas.

His mission is to support OS/2 Systems Management applications. Craig joined IBM in 1991. He has a BS and an MBA degree in computer information systems from Tarleton State University. Craig can be reached at USIB5ST3 at IBMMAIL.

## 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.)

# OS/2 Tips and Techniques

## Hardware and Device Driver Conflicts

Hardware conflicts may arise when one or more peripheral cards (such as communications, scanners, sound) share any one of the following parameters: interrupt request (IRQ), port address, direct memory access (DMA) channel, or base address. Any one or all of the parameters may be used and some sharing between two devices is possible. However, failure to have unique parameters between devices may result in conflicts such as the following (to name a few):

- Error messages such as "device not found"
- Ring 0 system lockup
- Abnormal behavior by devices such as sound getting in a loop or hanging on a note

Conflicts not resolved at the hardware level may be due to device drivers. The three types of device drivers in OS/2 are:

- OS/2 drivers in the CONFIG.SYS file
- DOS drivers (such as virtual and physical device drivers) in the CONFIG.SYS and/or AUTOEXEC.BAT files
- Windows drivers in the AUTOEXEC.BAT file (usually as terminate-and-stay-resident [TSR] files with .EXE extensions) and in the SYSTEM.INI file in the OS2/MDOS/WINOS2 directory (with .SYS extensions)

Other drivers, which typically have .VXD extensions, aren't supported.

Solve hardware conflicts by maintaining a record that lists the four parameters discussed above and logs the device driver names and their paths to ensure there are no duplicate entries. If you suspect a device driver conflict, take these two preliminary steps:

1. Determine if the device works by itself. Remark out the suspect drivers (type REM at the beginning of the device lines) in the CONFIG.SYS and AUTOEXEC.BAT files. Place a semicolon (;) at the beginning of the SYSTEM.INI device lines. If the drivers function by themselves, the problem is elsewhere.
2. Identify the conflicting drivers by turning them on one at a time until the malfunction is repeated.

Once you find the conflicting drivers, you must separate them. If DOS drivers are moved from the CONFIG.SYS file, place them in the DOS\_DEVICE line of the application's Settings notebook. Drivers in the AUTOEXEC.BAT file can be separated by creating a separate batch file for the session housing the problem application.

If the conflict is caused by Windows drivers, do the following:

1. Create a clean SYSTEM.INI file and call it SYSTEM0.INI.
2. Create a separate SYSTEM.INI file for each problem Windows application and name it accordingly (SYSTEM1.INI, SYSTEM2.INI, etc.).
3. Create a separate batch file for each problem Windows application:

```
COPY SYSTEMX.INI SYSTEM.INI
WIN / applications_name
COPY SYSTEM0.INI SYSTEM.INI
```

This will launch the Windows application with its correct SYSTEM.INI file.

4. Create a separate program icon pointing to the appropriate batch file to launch the problem Windows application in a separate session.

Drivers may conflict with themselves. A driver that locks up part or all of a resource may have been loaded only once. If the driver is loaded

in the SYSTEM.INI file, it will only allow for one WIN-OS/2 session at a time unless the conflict resolution procedures above have been completed.

## Improving Performance in Windows Applications

Some windows applications may encounter problems when doing common functions such as saving a file, printing, or importing data from another application. Out-of-memory errors may occur or applications may appear to start but return to the icon.

To resolve these problems, change some of the settings for your Windows applications.

Open the Settings notebook for the Windows application. Click on the Session tab, then the WIN-OS/2 settings button. Change the attributes for the following settings:

```
DOS_HIGH = 0n
DOS_UMB = 0n
EMS_MEMORY_LIMIT = 0 or 2048
DPMI_DOS_API = Enabled
DPMI_MEMORY_LIMIT = 64
XMA_MEMORY_LIMIT = 4-10 MB
VIDEO_FASTPASTE = 0n
VIDEO_WINDOW_REFRESH = 5
HW_ROM_TO_RAM = 0n
HW_TIMER = 0n
IDLE_SECONDS = Max (60)
IDLE_SENSITIVITY = Max (100)
DOS_FILES = 50 (or more)
```

Once you change these settings, click on Save and exit from the program's Settings notebook.

## Changing Icon Font Sizes in WIN-OS/2

OS/2 2.1 does not normally provide for changing icon font size and type in WIN-OS/2 Program Manager. A workaround is available but remains limited in what it can do.

The workaround consists of adding two statements to the Desktop portion of the WIN.INI file:

1. From a command prompt in the OS/2 window, change to the WINOS2 directory by typing `CD OS2\MDOS\WINOS2`, then pressing Enter.
2. Type `E WIN.INI`, then press Enter.
3. Cursor down to [Desktop] and press Enter at the end of the last line to add a line.
4. Type:
 

```
IconTitleSize=n (where n is a number for the font size such as 8, 10, etc.)<enter>
IconFaceName=xxxxx (where xxxxx is a font name such as Helvetica, Roman, etc.)<enter>
```
5. Click on File, then Save.
6. If you get a prompt that there is no file type associated, click on Type, highlight Plain Text, then click on Set.
7. Exit the Editor and restart WIN-OS/2

*Note:* This is an unsupported workaround and not all font types or sizes will work.

### Lotus 1-2-3 4.01 General Protection Fault

When running Lotus 1-2-3 4.01 for Windows, you may get a general protection (GP) fault in the MAIN.EXE file. To eliminate this error, change two settings in the Lotus 1-2-3 Settings notebook:

1. Open the Lotus 1-2-3 Settings notebook.
2. Click on the Session tab, then WIN-OS/2 settings.
3. Scroll down to the following settings and highlight to change the values:

```
EMS_MEMORY_LIMIT 0
XMS_MEMORY_LIMIT 4096
```

### OS/2 for Windows General Protection Fault

If you have exactly 40 groups in the WIN-OS/2 full screen Program Manager using native DOS and Windows when you install OS/2 for Windows, you will get a GP fault in the PROGMAN.EXE file every time Program Manager runs, whether it's booted in OS/2 or DOS. This

is because OS/2 for Windows will create group number 41 for ATM Groups.

To fix this, you will need to edit the PROGMAN.INI file and remove references to Group 41 ATM.GRP:

1. Exit Windows Program Manager
2. Open the OS/2 Window or Full Screen session and type `CD WINDOWS` at the command prompt.
3. Type `E PROGMAN.INI` to edit the file.
4. Remove references to Group 41 ATM.GRP. Click on File, then Save.
5. If you get a prompt that there is no file type associated, click on Type, highlight Plain Text, then click on Set.
6. Exit the Editor and restart WIN-OS/2.
7. Move all icons from one group into another group and delete the unnecessary group, which is now only a folder.
8. Add ATM Group and a new icon for ATM Font Manager.
9. Exit WIN-OS/2.

### Printing Landscape on Legal-Sized Paper From Windows Applications

When you try to print in landscape format on legal-sized paper from a Windows application to a Hewlett-Packard LaserJet printer using the manual feed on the top paper tray, the document is read as being 11 inches in length. This occurs when there is no legal-sized paper tray.

The Paper Source in the WIN-OS/2 full screen Program Manager must be set to manual feed:

1. Double-click on the WIN-OS/2 full screen icon to open it.
2. Click on the Control Panel icon in the WIN-OS/2 Main folder.
3. Select Printers, then Setup.
4. Set Paper Source to Manual Feed.
5. Click on OK, then Close.
6. Exit and resend the print job.

### Disable Print Manager

If you want to keep the Print Manager window from opening every time a WIN-OS/2 full screen session or Program Manager session is started, edit the SYSTEM.INI file:

1. From a command prompt in the OS/2 window, change to the WINOS2 directory by typing `CD OS2\MDOS\WINOS2`, then Enter.
2. Type `E SYSTEM.INI`, then press Enter.
3. Cursor down to [BOOT] and press Enter at the end of the last line to add a line.
4. Type `MAVDMAPPS=`
5. Click on File, then Save.
6. If you get a prompt that there is no file type associated, click on Type, highlight Plain Text, then click on Set.
7. Exit the Editor and restart WIN-OS/2.

### OS/2 for Windows File Size Requirements

OS/2 for Windows requires that the files have the exact sizes as listed below. If any of the files differ, the install process won't generate an error and the Program Manager won't load.

The following files should be dated 3-10-92:

|             |              |
|-------------|--------------|
| KRNL386.EXE | 75490 bytes  |
| GDI.EXE     | 220800 bytes |
| USER.EXE    | 264016 bytes |
| WINFILE.EXE | 146864 bytes |
| CONTROL.EXE | 15872 bytes  |
| COMM.DRV    | 9280 bytes   |
| MOUSE.DRV   | 10672 bytes  |
| TIMER.DRV   | 4192 bytes   |
| VGA.DRV     | 73200 bytes  |
| 8514.DRV    | 92032 bytes  |

The following files should be dated 10-20-93:

|            |             |
|------------|-------------|
| WIN.COM    | 25640 bytes |
| WINOS2.COM | 25640 bytes |

The WINDOS.COM (44170 bytes) file will become WIN.COM when booted to DOS. The .COM files in the list are necessary to start Windows or launch the WIN-OS/2 full screen.

The Windows version can usually be determined by typing `WINVER` at an OS/2 or DOS command prompt. In the case of Windows 3.11, the version can be verified by going into Program Manager and selecting Help, then About. The dates on certain Windows files may also be 12-31-93.

# Corrective Service Information

Figure 1 shows maintenance release levels for the listed products. This information is effective as of May 30, 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<br>Service Requester<br>Fault Tolerance<br>User Profile Management (UPM) | 2.0     | IP06030   | IP06030    | 4-25-93     |   |
|   | 3.0     | IP07045   | IP07045    | 4-28-94     |   |
| LAN Server/DOS LAN Requester SelectPak  | 3.0     | IP07003   | IP07003    | 7-28-93     | Diskettes not available. Download from one of the BBSs.   |
| LAN Server HPFS   | 3.0     | IP07005   | IP07005    | 11-2-93     | IP07005 requires IP07001 be applied to system. IP07005 is not for LAN NetView users. Download from one of the BBSs.   |
| LAN NetView Prerequisite  | 1.0     | IP07006   | IP07006    | 11-8-93     | IP07006 is a prerequisite before applying LAN NetView. It contains IP07005 plus fixes for OS/2 2.x and DB2/2. Requires WR07010 applied with DB2/2 and XR06100 applied with OS/2 2.0. Download from BBS. |
| OS/2 Network Transport Services/2 SelectPak   | 1.0     | WR07020   | WR07020    | 10-11-93    | Diskettes not available. Download from one of the BBSs.   |
| OS/2 LAN Adapter and Protocol Support SelectPak   | 2.0     | WR07020   | WR07020    | 10-11-93    | Diskettes not available. Download from one of the BBSs.   |
| OS/2 Extended Services Database Manager ServicePak  | 1.0     | WR06035   | WR06035    | 11-18-93    | Supersedes WR06001, WR06002, WR06003, WR06004, WR06014, and WR06015.  |

Figure 1. Maintenance Release Levels

| Product/Component   | Release   | CSD Level | PTF Number | Change Date | Comments   |
|---|-----------|-----------|------------|-------------|--|
| 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   | WR07022    | 5-6-94      |  |
| 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   | WR07025    | 5-6-94      |  |
| Extended Services Comm Mgr ServicePak<br>3270, 5250 Emulation<br>CM SNA | 1.0       | WR06025   | WR06025    | 11-29-93    |  |
| System Performance Monitor (SPM/2)<br>ServicePak                        | 2.0       | WR06075   | WR06075    | 12/10/93    |  |
| OS/2 Network Transport Services/2<br>SelectPak                          | 2.00      | WR07045   | WR07045    | 4-27-94     |  |
| OS/2 LAN Adapter and<br>Protocol Support SelectPak                      | 2.20.2    | WR07045   | WR07045    | 4-27-94     |  |
| Communications Manager/2<br>Version 1.01 ServicePak                     | 1.00      | WR06050   | WR06050    | 6-11-93     | Available only on diskette.                                    |
| 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 Kit  | 2.0       | UN50382   | UN50382    | 12-31-93    |  |
| 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       | UN52906   | UN52906    | 12-31-93    |  |
| TCP/IP for OS/2 Programmer's Toolkit                                    | 2.0       | UN54155   | UN54155    | 12-31-93    |  |
| 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       | UN52386   | UN52386    | 12-31-93    |  |
| 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. The following items are trademarks or registered trademarks of their respective companies or organizations.

Advanced Peer-to-Peer Networking, AIX, AIX/6000, APPN, AS/400, CICS, Common User Access, Communications Manager, CUA, DATABASE 2, Database Manager, DB2/2, DCE/6000, DisplayWrite, Extended Services, IBM, IBMLink, Micro Channel, NetView, OS/2, Personal Computer AT, PowerPC, Presentation Manager, PS/1, PS/2, RISC System/6000, ServicePak, System/360, System/390, ThinkPad, ValuePoint, WIN-OS/2, Workplace OS, and

Workplace Shell; all of International Business Machines Corporation

ANSI; American National Standards Institute  
Apple and NuBus; Apple Computer, Inc.  
Borland C++; Borland International, Inc.  
CompuServe; CompuServe, Inc.  
Hewlett-Packard, HP, and LaserJet; all of Hewlett-Packard Co.  
Indianapolis 500 and Indianapolis Motor Speedway; Indianapolis Motor Speedway Corporation  
Intel, Pentium, 80386, 80486, 80386SX, 80386DX, 80386SL, 80486SL, and 82360SL; all of Intel Corporation  
Internet; Internet, Inc.  
Kerberos; Massachusetts Institute of Technology  
LapManager and USAC; United States Auto Club  
Lotus, Lotus 1-2-3, and Notes; all of Lotus Development Corp.

Network File System, Solaris, and NFS; all of Sun Microsystems, Inc.  
PCMCIA; Personal Computer Memory Card International Association  
Paradox and Quattro Pro; Borland International, Inc.  
ProComm; Datastorm Technologies, Inc.  
SMART, SourceLink, and Source Migration Analysis Reporting Toolset; all of One Up Corporation  
Stacker; Stac Electronics  
Quicken; Intuit Company  
UNIX and UNIXware; Novell, Inc.  
VESA; Video Electronics Standards Association  
Windows, Word for Windows, Windows NT, MS-DOS, and Microsoft; all of Microsoft Corporation  
WordPerfect; WordPerfect Corporation

# STOP!

**Have you used the reader service card to request fast, free information about the products and services advertised in *Personal Systems*?**

**NO?** **Caution.** Back up a few pages. 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.

- 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.** Turn the page and speed up on the job with the up-to-date information you'll receive from vendors!

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.



## 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.

### 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 Ultimea 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 DatagLANce 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 MPM/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  
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

## Issue 4, 1991

Power Factor: Non-Linear Loads and the Power Distribution  
System  
Database Manager: Highlights and Direction  
OS/2 Communications Manager  
Improving OS/2 Application Performance  
Creating PM Windows with Dialog Templates  
REXX Program for OS/2 LAN Server  
Micro Focus COBOL/2 and the DOS Database Requester  
IBM DOS 5.0 Facts and Features  
IBM DOS 5.0 Upgrade  
DOS 5.0 Performance Improvements  
DOS Memory Management Facilities  
Disk Caching Under DOS  
NetWare Client-Server Interaction  
LANACS Protocols

## B A C K I S S U E O R D E R F O R M

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

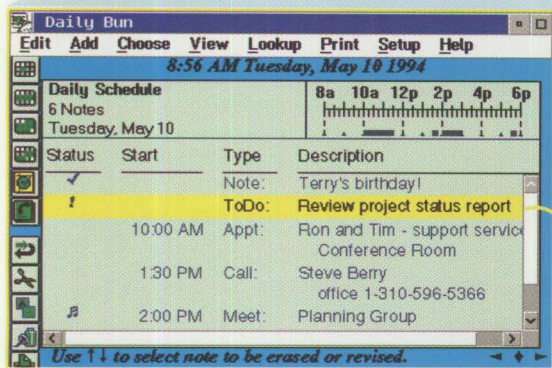


Relish Buns



## 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.



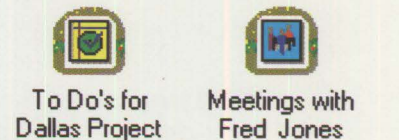
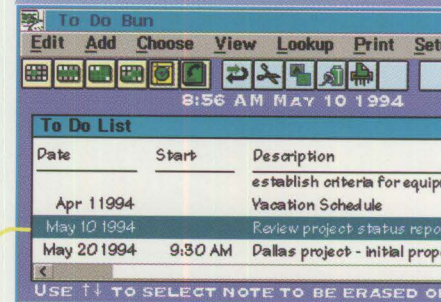
Here's how they work...  
 Buns give you views of your  
 information when and where you  
 need it. Use them just as you would  
 Relish - or customize them to match your needs.

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.



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

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!

**Call 310•596•5121**

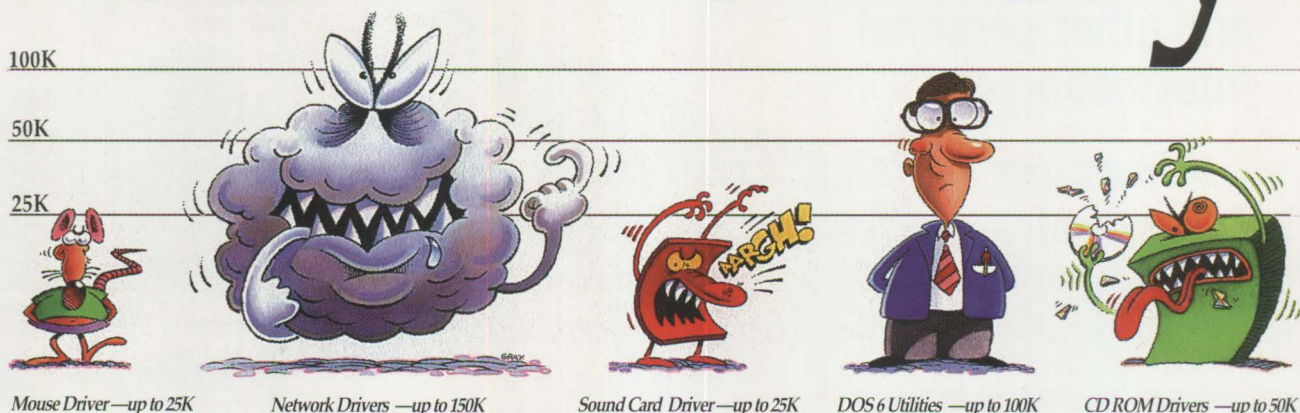
**Sundial Systems  
 Corporation**

909 Electric Ave, Suite 204,  
 Seal Beach, CA 90740 USA

**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.

# What's Eating Your Memory?



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.