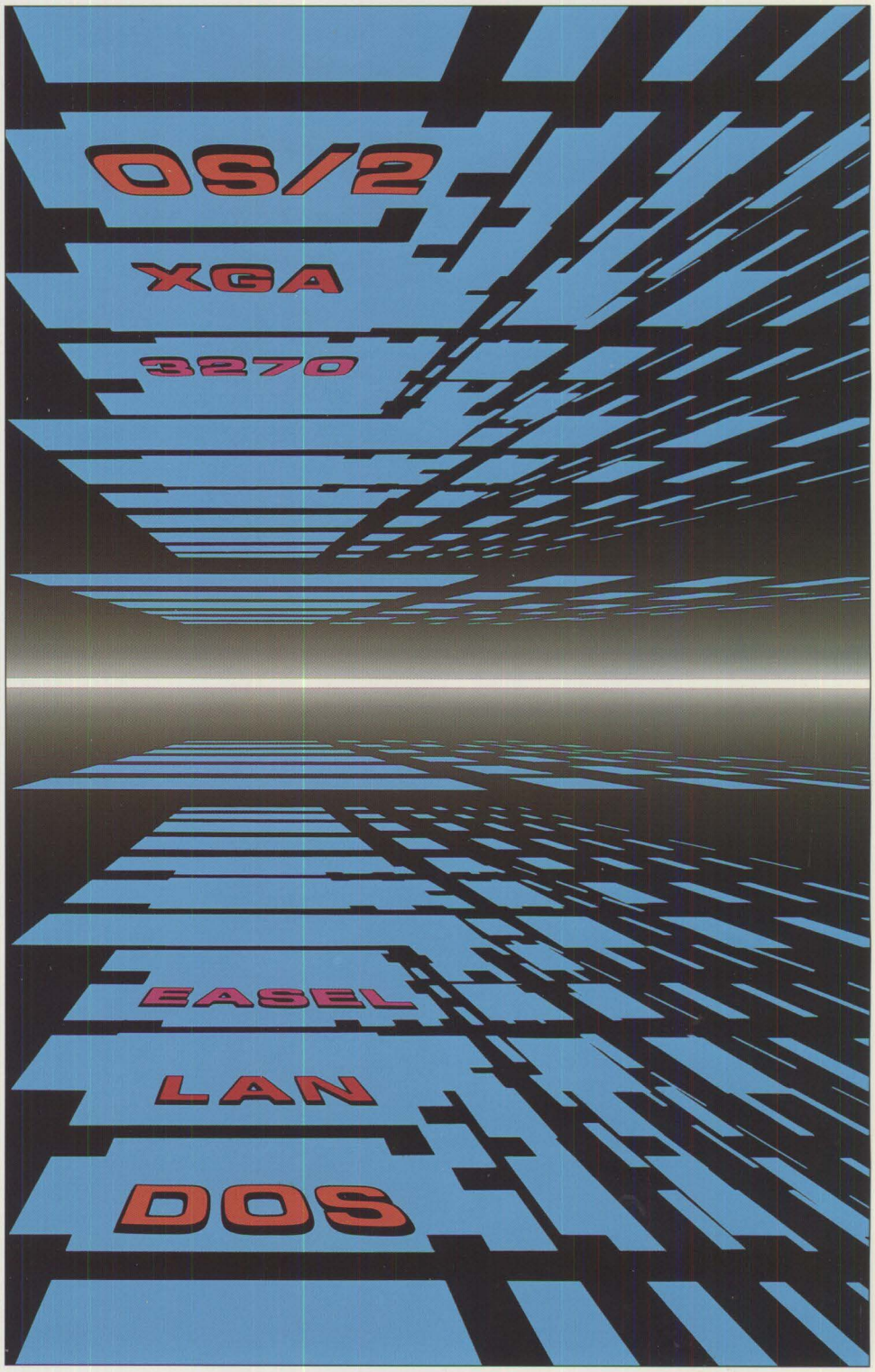
Issue 1, 1991

Personal systems



IBM Personal Systems Technical Solutions

IBM®

IBM Personal Systems Technical Solutions is published quarterly by the U.S. Marketing and Services Group, International Business Machines Corporation, Roanoke, Texas, U.S.A.

Editor	Ed Bamberger
Consulting Editor	Libby Boyd
Automation Consultant	Andy Frankford
Technical Consultant	Tom Sutherland
Little Solutions	Larry Pollis
Communications	Elisa Davis
Design	Corporate Graphics
Illustrator	Bill Carr
Manager	Bill Hawkins

To correspond with *IBM Personal Systems Technical Solutions*, please write the editor at:

IBM Corporation
Internal Zip 40-A2-04
One East Kirkwood Blvd.
Roanoke, TX 76299-0015

To subscribe to this publication, use an IBM System Library Subscription Service (SLSS) form, available at IBM branches, and specify form number GBOF-1229.

Contents

Trademarks

IBM, AIX, DisplayWrite, Micro Channel, NetView, Operating System/2, OS/2, Personal Computer AT, AT, Personal System/2, PS/2, PROFS, Quickwriter, Quietwriter, RT, Wheelwriter, and Writing to Read are registered trademarks of International Business Machines Corp.

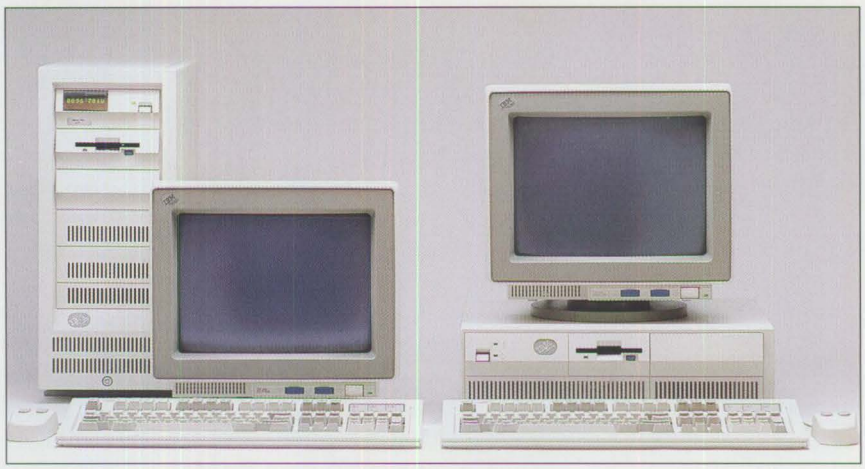
AD/Cycle, AIXwindows, Application System/400, AS/400, Assistant Series, Audio Visual Connection, AVC, BookManager, C/2, COBOL/2, Communications Manager, Database Manager, DisplayWrite 5, FaxConcentrator, FORTRAN/2, GDDM, Independence Series, KnowledgeTool, LinkWay, MVS/ESA, MVS/SP, MVS/XA, NetView/PC, OfficeVision, OS/400, Pascal/2, Personal Computer XT, PC XT, Presentation Manager, PS/1, Storyboard, Systems Application Architecture, SAA, Systems Network Architecture, SNA, System/370, VM/XA, VTAM, Virtual Telecommunications Access Method, Writing to Write, and 3090 are trademarks of International Business Machines Corp.

AT&T is a registered trademark, and Systimax is a trademark of AT&T.
AutoCAD is a registered trademark of Autodesk Inc.
Banyan Vines is a trademark of Banyan Systems Inc.
CHIPCOM is a registered trademark of CHIPCOM Corp.
DEC is a registered trademark of Digital Equipment Corp.
EASEL is a registered trademark of Interactive Images Inc.
Ethernet is a registered trademark of Xerox Inc.
Intel is a registered trademark, and i860, 386, i486, 486, 80286, 80386, 80386SX, and 80486 are trademarks of Intel Corp.
LAN TIMES is a trademark of McGraw-Hill Inc.
Lotus and 1-2-3 are registered trademarks and 1-2-3/M is a trademark of Lotus Development Corp.
Microsoft, MS, and Works are registered trademarks, and Microsoft C, Microsoft SQL Server, and Windows are trademarks of Microsoft Corp.
MYLAR is a registered trademark of E.I. du Pont de Nemours and Co.
Northern Telecom is a registered trademark of Northern Telecom Ltd.
Novell and NetWare are registered trademarks of Novell Inc.
Oracle Server for OS/2 is a registered trademark of Oracle Corp.
PC World is a trademark of PCW Communications Inc.
PostScript is a trademark of Adobe Systems Inc.
ProNET 80 is a registered trademark of Proteon Inc.
SQLBase Server is a registered trademark of Gupta Technologies Inc.
Software Digest, BUYERS ALERT, and RATINGS REPORT are registered trademarks of National Software Testing Laboratories Inc.
SynOptics is a registered trademark of SynOptics Communications.
Triumph and Triumph/DES are trademarks of Micronyx Inc.
WordPerfect is a trademark of WordPerfect Corp.

XGA – Raising Video Expectations

Jim Paolantonio
IBM Corporation
Boca Raton, Florida

This article reviews the Extended Graphics Array (XGA) video subsystem, along with a short introduction to the new PS/2® models 90 XP486 and 95 XP486, in which the XGA resides.



PS/2 Models 95 XP486 (left) and 90 XP486 (right)

The two new "high-end" PS/2s are the desktop Model 90 XP486 and the floor-standing Model 95 XP486. Both models share some common features. One of the unique new features is an "expandable processor complex," based on the Intel® 32-bit 80486 processor, which is available with speeds of 25 or 33 MHz, and cache memory. The processor complex consists of a card that plugs into a special connector on the system board. In all previous PS/2s, the processor was integrated onto the system board. The processor complex card allows system flexibility for upgrading to new processors at a later date.

The Small Computer System Interface (SCSI) (an industry-standard interface for attaching hard files, CD-ROM drives, diskette drives, and tape backup) adapter, comes standard in both systems, along with a selection of 80, 160, or 320 MB SCSI hard files.

A new level of video that "raises the bar" has been introduced on these two systems. In the Model 90 XP486, the XGA video subsystem is integrated directly on the system board. The Model 95 XP486 has the XGA Display Adapter/A resident in a new Micro Channel® slot.

XGA Video – "Raising the Bar"

In 1987, the Video Graphics Array (VGA) was introduced as the base video for all PS/2 systems; subsequently, it became pervasive throughout the personal computing industry. VGA combined all the functions of its predecessors – MDA, CGA, and EGA – along with new graphics resolutions ranging up to 640 x 480 pixels with 16 colors. At the same time, the 8514 Display Adapter/A was introduced to meet customer requirements for high-resolution 1024 x 768 (with 256 colors) graphics.

Now, you might think that the next logical step for video on PS/2 systems would be to combine the capabilities of both VGA and high-resolution graphics. You're absolutely right!

In response to market requirements, the Extended Graphics Array (XGA) video subsystem has been introduced on the new PS/2 models 90 XP486 and 95 XP486. XGA combines the best of both worlds, offering a higher speed VGA and Extended Function 1024 x 768 graphics resolution, as illustrated in Figure 1.

In the desktop PS/2 Model 90 XP486, the XGA video subsystem is integrated directly on the system board. Now VGA and high-resolution 1024 x 768 graphics are available together in the base PS/2 system unit without requiring an additional Micro Channel display adapter.

This sets a new level of expectation for video on PS/2 system units.

The XGA Display Adapter/A has also been developed. This Micro Channel adapter extends XGA's advantages to the new Model 95 XP486 (which has no video on the system board) and to current PS/2 systems that have 80386™, 80386SX™, or 80486™ processors. This includes systems such as the PS/2 models 55, 65, 70, and 80.

Considerable attention was given to providing XGA application program and operating system support.

XGA software device drivers were developed to support the existing 8514/A DOS application program base and the Microsoft® Windows™ and OS/2 Presentation Manager™ operating environments.

Faster Than a Speeding VGA...

XGA has two mutually exclusive operating environments – VGA compatibility mode and Extended Graphics function.

The XGA supports all existing VGA video modes. In addition, current VGA applications will run faster on the XGA.

What accounts for this improved performance? XGA has employed video random access memory (VRAM) technology.

Unlike Dynamic RAMs (currently used in most VGA subsystems), VRAMs are dual-ported, which simply means they can "juggle two things at once."

The XGA display controller can update the video data in the VRAMs simultaneously while the VRAMs are busily refreshing the display. The net effect: noticeably increased performance.

XGA – High-Resolution Support

A large number of popular applications, such as AutoCAD® and WordPerfect® have been developed to support the 8514/A high-resolution, 1024 x 768 graphics mode. These applications are written to the 8514/A Adapter interface, a software interface between the application and the 8514/A hardware.

The XGA's Extended Graphics Function carries this momentum forward, by maintaining 8514/A DOS application program compatibility at the adapter interface level. This allows portability of 8514/A applications to the new XGA hardware.

A new DOS Adapter Interface Version 3.0 for the XGA offers a superset of the original 8514/A DOS Adapter Interface. Most applications currently written to the 8514/A DOS Adapter Interface will run on the new XGA DOS Adapter Interface. Over time, these applications may be expanded to exploit some of the new functions offered by the XGA DOS Adapter Interface, such as a Sprite (hardware cursor).

So you ask... "What's new about the XGA Extended Graphics Function mode?" XGA offers some impressive enhancements, which include the Direct Color and Hardware Drawing Assist.

XGA offers resolutions up to 1024 x 768 pixels with 256 colors (depending on the amount of video RAM installed, as you will see later). XGA introduces a new 16-bit-per-pixel (BPP) "Direct Color" function that delivers an impressive 64 K (65,536) range of colors at a resolution of 640 x 480 pixels. This allows more realistic images to be portrayed on the display.

Another XGA function is Hardware Drawing Assist. The XGA Display Controller coprocessor hardware can directly change ("draw") the

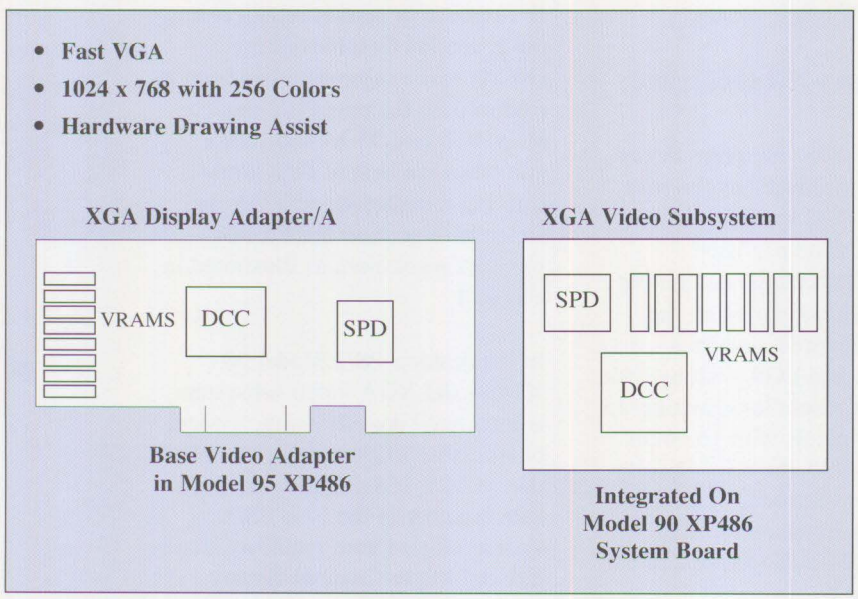


Figure 1. XGA Video – Combining the Best

data stored in the video display buffer or system memory, under control of simple commands from the system processor. These commands allow lines to be drawn, blocks of data to be moved, and areas of color to be filled on the display. These Hardware Drawing Assist functions are available with a variety of line styles and colors, along with a set of arithmetic/logical/pattern mixes that can used to combine new data with the displayed data.

Two new XGA Hardware Drawing Assist functions, notably the bus master function and the Sprite (hardware cursor) have been added to accelerate performance.

Acting as a 32-bit bus master, XGA can access both system memory and the video display buffer memory.

How is this bus master function beneficial? Application programs can construct images (bitmaps) in the system memory for later display on the screen. The XGA, acting as a bus master, then transfers the images from system memory to the video buffer for immediate display. This dramatically improves video performance.

The second enhancement offered by XGA is the Sprite. Typically, cursors displayed on the screen are software cursors. As the cursor (or icon) maneuvers along the screen, the application software must constantly save and restore the background video data that the cursor overwrites, which can result in reduced performance and a flickering effect.

The XGA has a distinct Sprite buffer in addition to the video display buffer. The Sprite overlays the background video display buffer at its current position without changing the background data. The application is relieved of the burden of saving or restoring the background video data, thereby increasing performance and enhancing "front-of-screen" quality.

The XGA Sprite has a resolution of 64 x 64 pixels with two colors available, which is useful for icon display.

Acting as a 32-bit bus master, XGA can access both system memory and the video display buffer memory.

The Dynamic Duo – XGA Plus 8515

Early in 1990, IBM introduced the 8515 Color Display. The 8515 has a market-driven design that provides the same resolution, 1024 x 768, as the 8514 display, but has three distinct advantages: more compact design, improved ergonomics, and lower price.

The 8515's smaller, 14-inch screen and "front-of-screen" controls made it more suitable for placement on top of a PS/2 system unit.

But the most persuasive advantage was the price. The 8515's competitive price placed it within the reach of customers who normally might have opted for only a VGA display. Coupled with XGA, standard on PS/2 models 90 XP486 and 95 XP486, users now have a cost-effective, high-resolution graphics solution in the base system unit.

As to be expected, the XGA also supports the full range of PS/2 displays, starting with the monochrome 8503 (640 x 480 with 64 gray shades) through the 8514 (1024 x 768 with 256 colors).

Video Memory Expansion – More Colors

The base XGA video subsystem on the Model 90 XP486 and the XGA Display Adapter/A both come with 512 KB of video memory installed. This allows resolutions of 640 x 480 with 256 colors, or 1024 x 768 with 16 colors.

The PS/2 Video Memory Expansion option allows the system to be upgraded to 1 MB maximum. This extends resolutions to 1024 x 768 with 256 colors, or 640 x 480 with 64 K colors (direct color).

BVEC – Bridging the Connection

Up to now, all PS/2 system units have had VGA video integrated on the system board. The VGA subsystem (in addition to supporting a VGA display directly off the system board) drives video data out to the Micro Channel Auxiliary Video Extension Connector (AVEC) slot. This allows display adapters plugged into the AVEC slot (like the PS/2 Image Adapter/A that does not have VGA capability onboard) to display this VGA data on the monitor connected to the adapter.

On the PS/2 Model 90 XP486, the XGA video subsystem is resident on the system board. The XGA (while in VGA mode) maintains the same AVEC support as VGA did in the past.

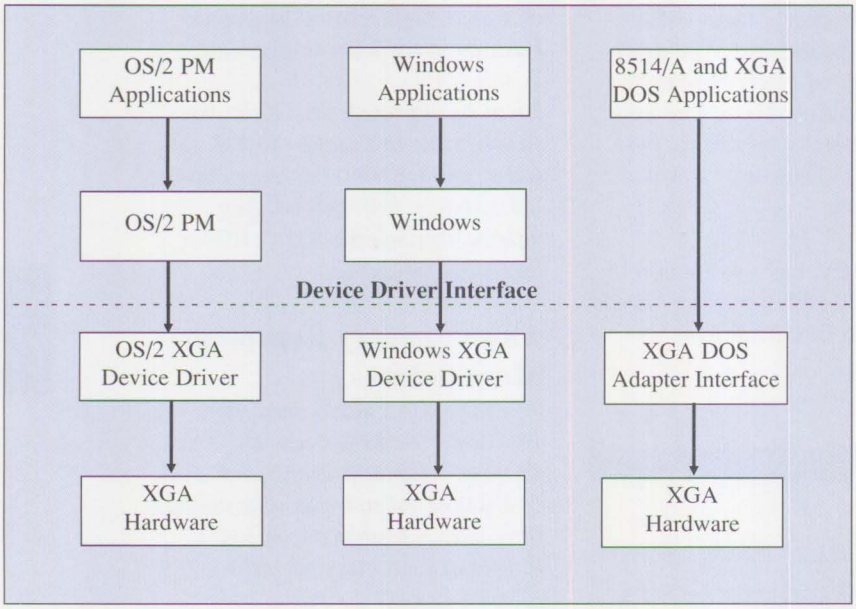


Figure 2. XGA Device Driver Interfaces

On the PS/2 Model 95 XP486, base video is generated by the XGA Display Adapter/A resident in a newly designed Base Video Extension Connector (BVEC) Micro Channel slot. This BVEC slot is identical in function to the previous AVEC slot, although its video data pins are physically offset.

The XGA display adapter drives VGA data from the new BVEC slot out to a separate AVEC slot. Other display adapters (such as the Image Adapter/A) plugged into the AVEC slot, receive the VGA data as before.

Multiple XGA display adapters may coincide in a system unit, under the control of the Adapter Interface or application program.

Device Drivers – Strategic Support

It is projected that over the next few years, most mainline applications will be written to the various windowing managers' Graphical User Interfaces (GUIs). Window managers typically require more pixels on the screen to support multiple nonoverlapped windows. The XGA hardware has been optimized for high resolution and performance in this environment.

With this in mind, several high performance XGA device drivers have been developed. These drivers bridge the interface between the XGA hardware and the application or operating environment, as shown in Figure 2.

Three XGA device-driver diskettes are shipped with the models 90 XP486 and 95 XP486 systems and with the XGA Display Adapter/A. These support the following strategic applications and operating environments:

- DOS Adapter Interface Version 3.0 (8514/A and XGA application support)
- Microsoft Windows 286/V2.1
- Microsoft Windows 3.0
- AutoCAD Release 10
- OS/2 Presentation Manager 1.2

ABOUT THE AUTHOR

Jim Paolantonio is an advisory engineer at the IBM's Entry Systems Division Laboratory in Boca Raton, Florida. Presently, he works in the system planar development department where he is responsible for the XGA video subsystem. Since joining IBM in 1982, he has been involved with the development of the PC XT 286, PC AT, and the PS/2 Model 80-311. Jim received his B.S. and M.S. degrees in electrical engineering from Purdue University.

Choosing between Shielded and Unshielded Wiring for Data Transmission

Ned Sigmon
AMP Incorporated
Winston-Salem, North Carolina

This article will help users decide which cabling is a better choice for their LAN environments: shielded or unshielded twisted pair.

Last year, a new 10BASE-T supplement to the Institute of Electrical and Electronic Engineers (IEEE) 802.3 networking standard was voted out of committee. This addition will govern the use of unshielded twisted pair (UTP) wiring for 10 Mbps Ethernet® applications.

For some time, many end users and systems integrators have expressed a high level of interest about the issue of adopting 10BASE-T, and a flood of compatible products has emerged. Many leading industry experts have predicted that users will select UTP instead of coax as the Ethernet media of choice. The coax-based share of new Ethernet shipments is expected to decline from 90 percent in 1988 to about 20 percent by 1993.

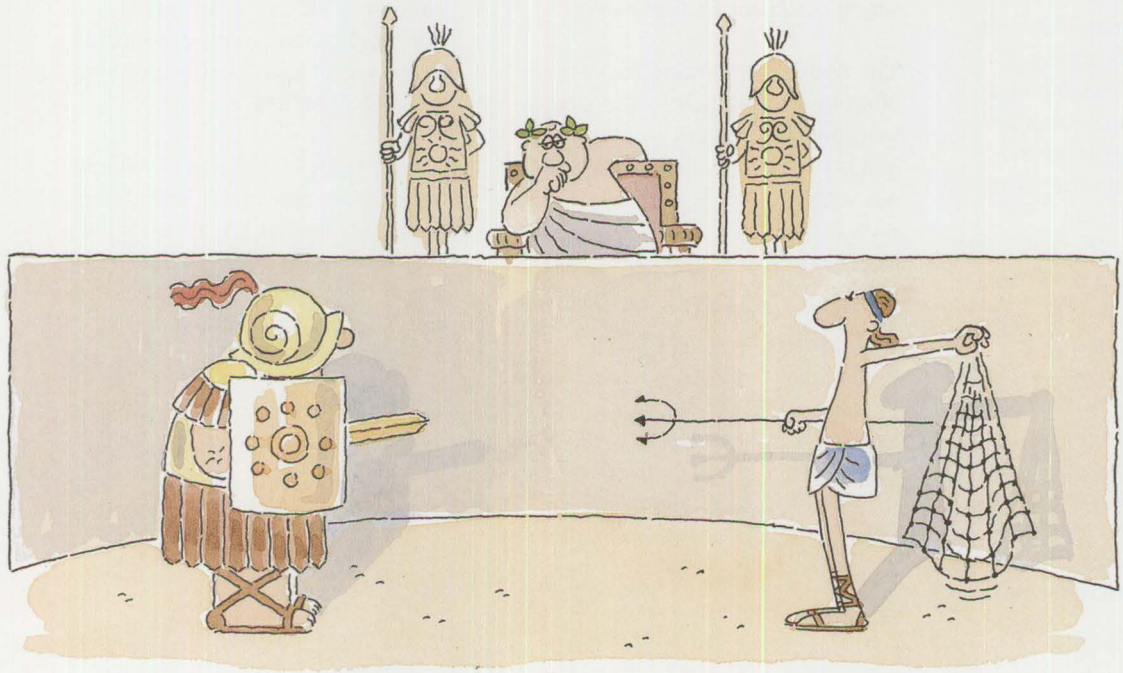
Presently, there are many new products available that claim to facilitate 16 Mbps token ring data communication over UTP. A new IEEE study group is investigating the feasibility of a standard to cover this application.

Some have interpreted these developments to mean that 150-ohm shielded twisted pair (STP) may also be replaced by UTP as a universal media for all current network applications. But we shouldn't be too quick to write the obituary on shielded twisted pair wiring. Many technical issues surrounding the use of UTP for 16 Mbps token rings have not been resolved, and the jury is still out on whether all of the 16 Mbps UTP products recently introduced will work in the majority of actual user environments.

High-Frequency Transmission Considerations

Most of the UTP wiring in use or being installed today was originally designed for analog voice transmissions and has not been optimized for digital transmissions. The most common type is the telephone-grade distribution inside wiring (DIW), which uses polyvinyl chloride (PVC) insulations.

DIW UTP has a higher attenuation than STP. It is more susceptible to impedance variations, crosstalk, transient damage to the data link interfaces, and outside electrical interferences. It emits more electromagnetic radiation at higher frequencies. Attenuation, crosstalk, noise, and the reflections introduced by impedance variations are all factors that degrade signals and increase timing jitter, making the pulses



more difficult to distinguish at the receiver. The result can be increased transmission errors and network time lost in regenerating the signal. In the worst-case scenario, regeneration may not be enough, and the network communication may be disrupted altogether.

Because attenuation and jitter increase with distance, the use of UTP with either a 4 or 16 Mbps token ring will limit the total length of the ring and the number of stations that can be attached. The IEEE study group hopes to accommodate 72 stations with UTP, and in that event, only if all stations are served from the same wiring closet. The limit could prove to be much lower, probably around 40 workstations with DIW-grade wiring. While that may accommodate departmentally sized rings, it will not keep pace with the trend toward company-wide rings with 100 or more stations. In contrast, the 150-ohm STP accommodates up to 260 stations on a 16 Mbps token ring without repeaters or converters.

Although UTP can have advantages over STP – smaller size, lighter weight, and greater flexibility – the decision to use UTP is almost always an economic decision based on installation costs alone, not a technical decision to provide the most viable information transport system.

In new installations, the economic advantages of UTP over STP may not prove to be as significant as might first appear. Industry studies have shown that cable material costs generally amount to less than 10 percent of the total cost of an installed network. Installation labor is the overriding factor, running four to five times the cost of the cable. Overall, the actual difference between the installation costs of UTP and STP usually amounts to less than one-tenth of one percent of the cost of building and outfitting a new office. The on-going maintenance cost and cost of additional electronics far exceed any savings gained by installing UTP over STP.

The Use of Existing UTP Cable Plants

One of the perceived economic advantages of UTP is that it is already installed in existing telephone cable plants. Such installations can have a number of deficiencies, such as cable not made to today's standards, multipaired cables with shared services, crossed-pair polarities, or hidden bridged taps and poor terminations, either of which can create signal reflections. There have been reports of 10BASE-T installations that do not work because of deficiencies in existing wiring. At higher data rates these deficiencies become even more critical. Simple continuity tests are not sufficient for identifying potential problems. To use existing wiring, elaborate tests may have to be conducted, and in some cases the installer will have to find a combination that works.

The costs of qualifying the cable plant to the LAN application, or reworking the problems found, could quickly offset the cost of pulling new cable.

Waiting for Standards That Ensure Consistency and Reliability

Realistically, a standard governing the use of UTP for 16 Mbps token rings is at least two years away. The IEEE study group is still struggling to define technical and economic feasibility, which is only the preliminary step in developing detailed specifications.

Until these specifications are finalized, and all vendors are building to uniform specifications, the responsibility of determining whether the UTP LAN products will work in a particular environment falls upon the end user. Those choosing a particular system may be locked into a single source. This is because all current systems are proprietary, and there is no guarantee that a station lobe or a ring will operate with multiple products from different manufacturers. There is also no guarantee that any current system will be compatible with the standard when and if it is issued.

The Differences Outweigh the Similarities

The fact that the 10BASE-T standard is now virtually complete may suggest that the groundwork is complete, too, and a 16 Mbps standard will be a simple extension. However, operating a 16 Mbps token ring over UTP is much more formidable. In a 10BASE-T system, both the terminal and the concentrator in the wiring closet transmit data frames. The maximum distance the signal has to travel before regeneration is 100 meters.

In token rings that operate without repeaters or converters, only the terminals retransmit the token or data. With 100-meter lobe lengths, the signal must survive a total transmission distance of 200 meters (or more, if the terminals are separated by more than one wiring closet). This makes token ring transmissions more vulnerable to attenuation or degradation from external interference.

In a UTP installation, crosstalk can occur between active pairs of a cable (even if both are being used for token ring) or between cables closely confined in a conduit or raceway. In contrast, the pairs of a Type 1 STP cable are individually shielded, and the crosstalk of this configuration was considered to be so low in the formulation of the original 802.5 standard that crosstalk specifications were not even mentioned.

Ethernet also has the advantage that with a 10 MHz fundamental frequency, the frequency band containing the majority of signal energy is still below the radiation frequency range regulated by Federal Communications Commission (FCC) Part 15, Subpart J (which is the frequency range controlled by FCC regulations on electromagnetic radiation). The bandwidth of 16 Mbps token ring signals extends well within the range of FCC control, and filtering or other nonstandard transmission techniques are required to suppress the high frequency energy in the absence of shielding. The additional cost for electronic devices to enable the use of UTP can easily exceed the cost difference between UTP and STP.

The Telephony Influence on UTP

Because DIW-grade premise UTP was originally designed for analog voice applications, high-frequency transmission characteristics are not tightly controlled. At this point it is still unclear whether a token ring standard can guarantee acceptable transmission distances and ring sizes over the full range of UTP variability.

Prior to the deregulation of premise wiring, UTP wiring within a building was controlled solely by the telephone company. Telephone and data systems were kept rigidly separated, and data systems manufacturers specified their own forms of wiring.

Since deregulation, the UTP specifications have remained relatively unchanged and continue to reflect a telephone background, which is a design intended for analog voice applications. Few systems manufacturers issued minimum requirements for acceptable transmission properties when UTP is used with their equipment (for example, IBM's Type 3 Specifications). However, in recent years, through the efforts of the EIA/TIA TR 41.8.1 committee that is developing a standard for commercial building wiring, there has been an attempt to set industry-wide specifications. This standard is still being finalized, and should soon be published under the EIA/TIA 568 designation.

The IEEE study group intends to use the EIA/TIA standard as a baseline. The EIA/TIA specifications written for the better-quality DIW-grade cables allow for more crosstalk than a 16 Mbps token ring operating over 100-meter station lobes could tolerate. Current UTP token ring products take advantage of EIA/TIA specifications that represent the extremes of manufacturing tolerances, and the majority of DIW-grade cable pairs have properties exceeding minimum EIA/TIA requirements. Nevertheless, the EIA/TIA limits are real and can be expected for about five percent of the pairs.

Most of the attention on UTP has been focused on the wiring between the telecommunications closet and the work station outlet. But in a token ring environment, the most critical length of cable is the line cord from the terminal to the outlet. This is the length closest to the transmitter, where both the signal strength and the chance of introducing crosstalk are the greatest. Unfortunately, it is also the length that traditionally has used the poorest-quality cable. The parallel conductor line cords commonly used for telephone connections are not acceptable for 16 Mbps token rings.

Temperature Variations

The properties of PVC insulation used on DIW-grade cables are highly temperature-dependent. As a result, temperature variations in the installed environment can be a major factor in the performance variability of UTP wiring. Between room temperature and 104 degrees Fahrenheit, the attenuation of a 100-meter length of PVC-insulated wiring can increase by more than 20 percent. The jitter measured at 16 MHz can increase by more than 40 percent between the same two temperatures. It would not be unusual for cables routed through the ceiling or near heating ducts to undergo these kinds of temperatures.

The 150-ohm STP cables use foamed polypropylene or fluoropolymers (FEP) insulations, both of which are more stable than PVC. The attenuation of 16 MHz jitter of FEP-insulated wiring will increase less than two percent between 77 and 104 degrees Fahrenheit.

Connector Considerations

Much of the connecting hardware used with UTP wiring is a holdover from the telephone era, and in extreme cases may introduce as much high-frequency crosstalk as the cable. Modular connectors and telephone cross-connect blocks have not been designed or qualified for

high-frequency transmissions. On the other hand, the four-position data connector used with STP was designed from the start to satisfy digital transmission requirements.

Installation practices and termination techniques can have a major influence on the UTP transmission performance. A common practice in telephone wiring is to strip back several feet of cable jacket to break out the pairs for termination. This allows the conductors of the pairs to separate from each other and to intermix with the conductors of other pairs. The separation changes the impedance of the pair, and the intermixing increases the chance for crosstalk coupling. Even seemingly harmless practices, such as closely bundling pairs or cables to improve wire dress, can noticeably increase the crosstalk coupling at 16 Mbps.

The unpredictability of pair separation and intermixing can also be found in preassembled modular outlets and patch panels. The type of construction that is particularly vulnerable is where the modular jack contacts are individually wired with loose, discrete conductors. Patch panels using printed circuit board are more predictable, but their performance may still vary from pair to pair unless the manufacturer has designed the board to match impedance and minimize crosstalk. Studies have shown that patch panels with properly designed boards can have around 13 dB lower crosstalk at 16 MHz than flying-lead designs.

The Trend to Higher Performance Cables

System vendors and cable manufacturers both agree that 16 Mbps is pushing the practical limits of DIW-grade cable. Some cable manufacturers are now introducing UTP that is physically interchangeable with DIW, but offers improved specifications for attenuation, crosstalk, and electromagnetic interference (EMI) effects. Northern Telecom® markets a cable of this type called building data network (BDN), and AT&T® announced its Systimax™ Type 2061A. At least one of the pioneer 16 Mbps-over-UTP systems manufacturers now specifies the use of such cables, conceding the need for the better transmission characteristics.

These cables can support higher bit rates over greater distances, but the trade-off is a significantly higher cost than DIW. Because they can be used with the same connecting hardware as DIW, the higher-grade cables may have applications in retrofitting installations that are already heavily committed to modular, nonshielded wiring. In new installations, the higher cost erodes any economic justification for using UTP.

The Future of Fiber and STP

In some cases, the shift away from STP is based on the assumption that the introduction of Fiber-Distributed Data Interface (FDDI) and other fiber-based network standards lower the cost of fiber and make shielded copper wiring obsolete for higher data rates. Some users are installing both fiber and UTP to workstations with the intention of using UTP for current networking requirements, leaving the fiber unterminated until needed.

FDDI is several years away from being practical for anything but the most powerful workstations. In the meantime, there is a large installed base of STP that systems manufacturers cannot afford to ignore. There are products available that can handle video over STP. Proteon markets an 80 Mbps token ring system called ProNET 80® that operates over STP. SynOptics® Communications, CHIPCOM® Corporation, and DEC® have all announced plans to support FDDI to the workstation over STP wiring, projecting that the STP approach could lower FDDI connection costs by 60 percent or more. The future capabilities of STP have yet to be explored both in data and broadband applications.

Summary

Unshielded twisted pair wiring has its place in small networks and in installations that are difficult to recable. However, UTP is not a universal cable for all applications, because it does not offer the future flexibility that shielded twisted pair provides. In all cases, careful consideration should be given to the ramifications of choosing between shielded and unshielded twisted-pair media, both now and in the long term.

ABOUT THE AUTHOR

Ned Sigmon is a project development engineer with the Communication Products Division of AMP Incorporated, a leading producer of electrical and electronic connectors. Ned joined AMP Inc. in Winston-Salem, N.C., in 1979, and has held a number of positions in development engineering. Currently, he is responsible for the development of premise wiring systems for data communications. He has served on numerous committees on developing industry standards for premise wiring or networking.

Compatibility of LAN Servers and Requesters

Roy Feigel, Steven French, and Carolyn Easter
IBM Corporation
Austin, Texas

With the increasing number of IBM LAN server and requester products, one can easily become confused about their compatibility. The similarities and differences of the products are shown, which will help readers plan for LAN installation and administration, as well as migrate from one product to another. A technical overview of the requester/server relationship includes some tips and techniques for dealing with mixed LAN environments.

The requester/server relationship of primary interest is that of a particular level of requester to a specific level of domain or server. Within each requester/server relationship, there are functions that may or may not operate depending on the specific relationship.

The relationship between the requester and the server varies depending upon the version level. Certain functions may or may not operate between certain levels of servers and requesters. These functions are:

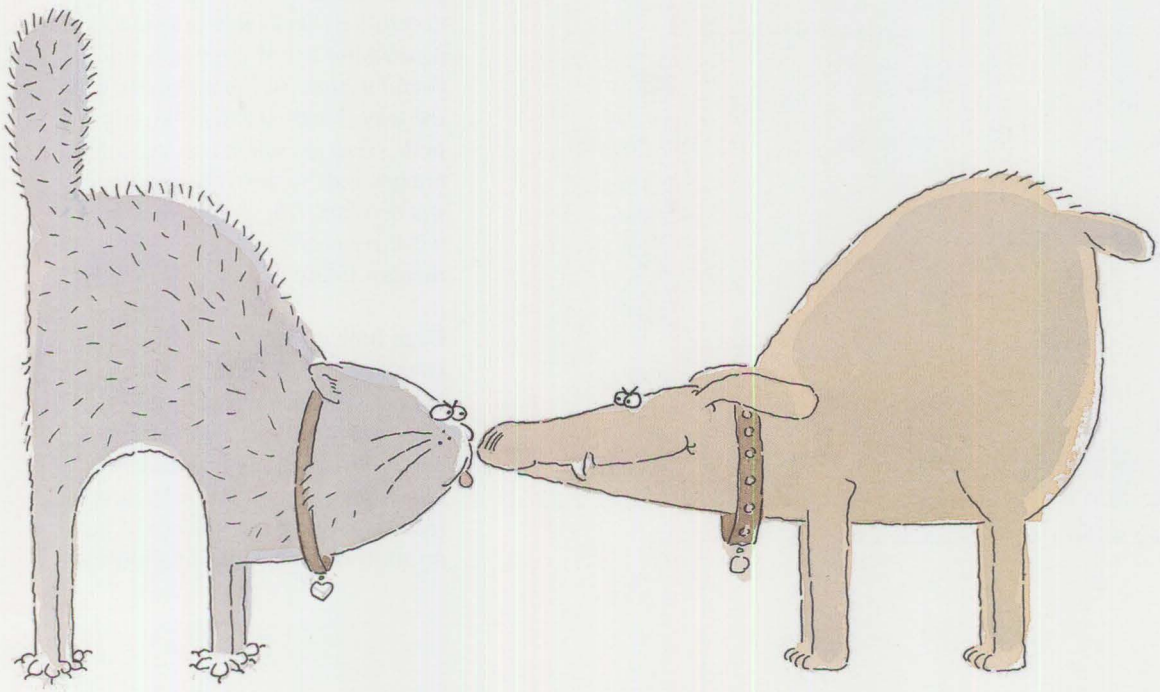
- Domain configuration
- Logon compatibility
- Administration
- Resource access

Readers should have a basic understanding of at least one of the following LAN products:

- PC LAN Program 1.3
- OS/2 LAN Server 1.0, 1.2, or 1.3
- OS/2 Extended Edition LAN Requester 1.1, 1.2, or 1.3
- DOS LAN Requester

Domain Configuration

A domain is a logical grouping of servers that act as a single system. There are many objects that make up or define a domain; for example, the servers themselves, user accounts, resources, and access control lists. These definitions, for the most part, are distributed throughout individual servers in the domain; but to end users and often the administrator, the domain appears as a single system. In fact, in a carefully administered domain, ordinary users would assume they are on a single system. This lack of perception, so to speak, is the fundamental basis of the domain concept.



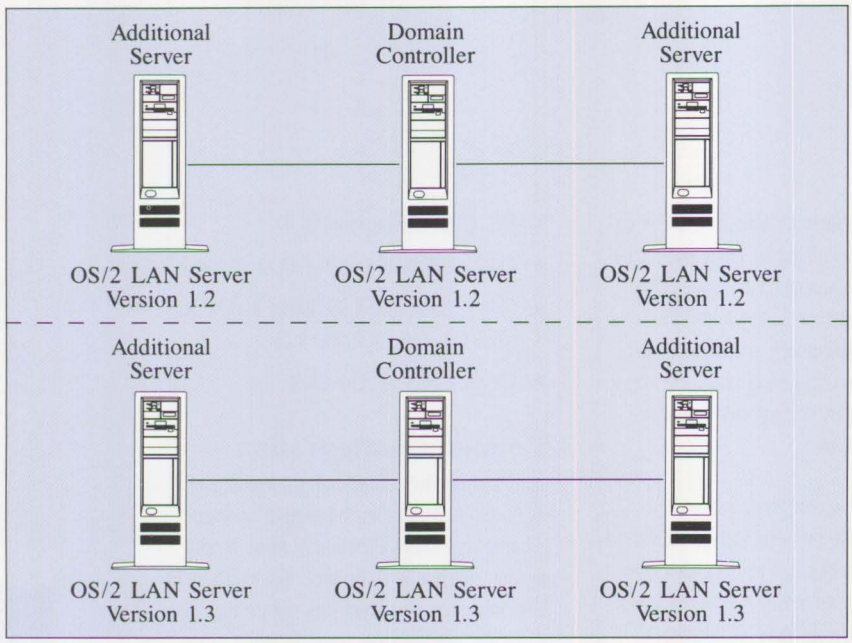


Figure 1. Compatible Server Configurations in a Domain

To configure a domain, a specific server-to-server relationship is required. Specifically, all servers in a given domain *must* be at the same release level. To ensure integrity of the domain across the servers, it is recommended that all servers in the domain be at the same corrective service level. Figures 1 and 2 are examples of compatible and incompatible server configurations within a domain.

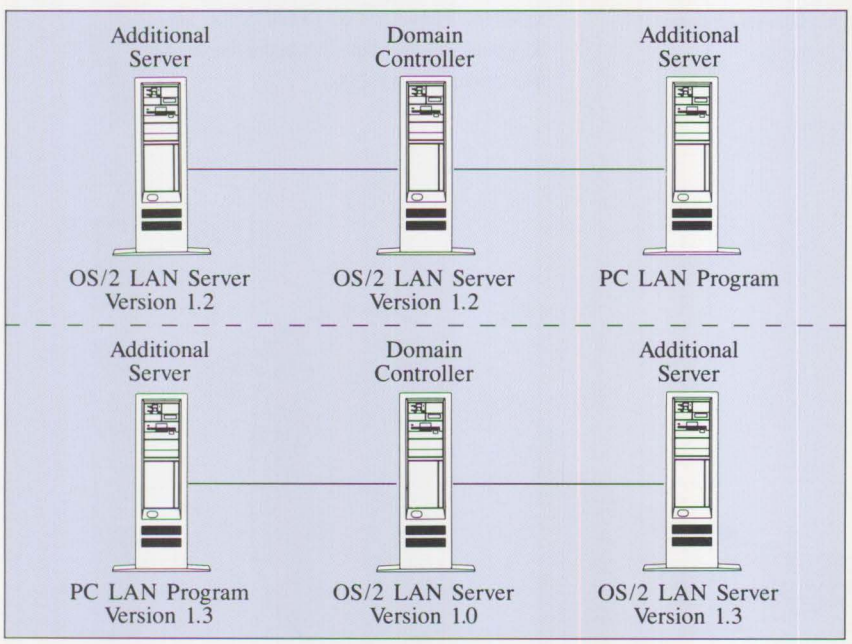


Figure 2. Incompatible Server Configurations in a Domain

More configurations are incompatible than compatible. At first glance, this may appear to be a severe limitation, especially when migrating existing servers to newer versions. But more important than the servers' relationship within the domain is the relationship of the requesters to the servers in the domain. This relationship is the focus of this article.

Logon Compatibility

Users of LAN requesters are required to log on to a given domain prior to accessing any network resources or performing any network management or administration tasks.

There are three basic internal processes associated with logon:

- User authentication
- Local user registration
- Connection to a set of predefined resources

Users are authenticated once the userid and password are entered and sent (encrypted) to the controller of the specified domain. The domain controller checks this information against the list of defined users. If found in the list, the requester stores the userid and password locally for authentication when making future connections to any other server in the network. The predefined network connections (logon assignments), if any, are then made.

Each time a requester attempts to connect to a server, either subsequent to or during logon, the requester sends the server the userid and password. Each target server must validate this information upon receipt, or the connection will be disallowed. This is true regardless

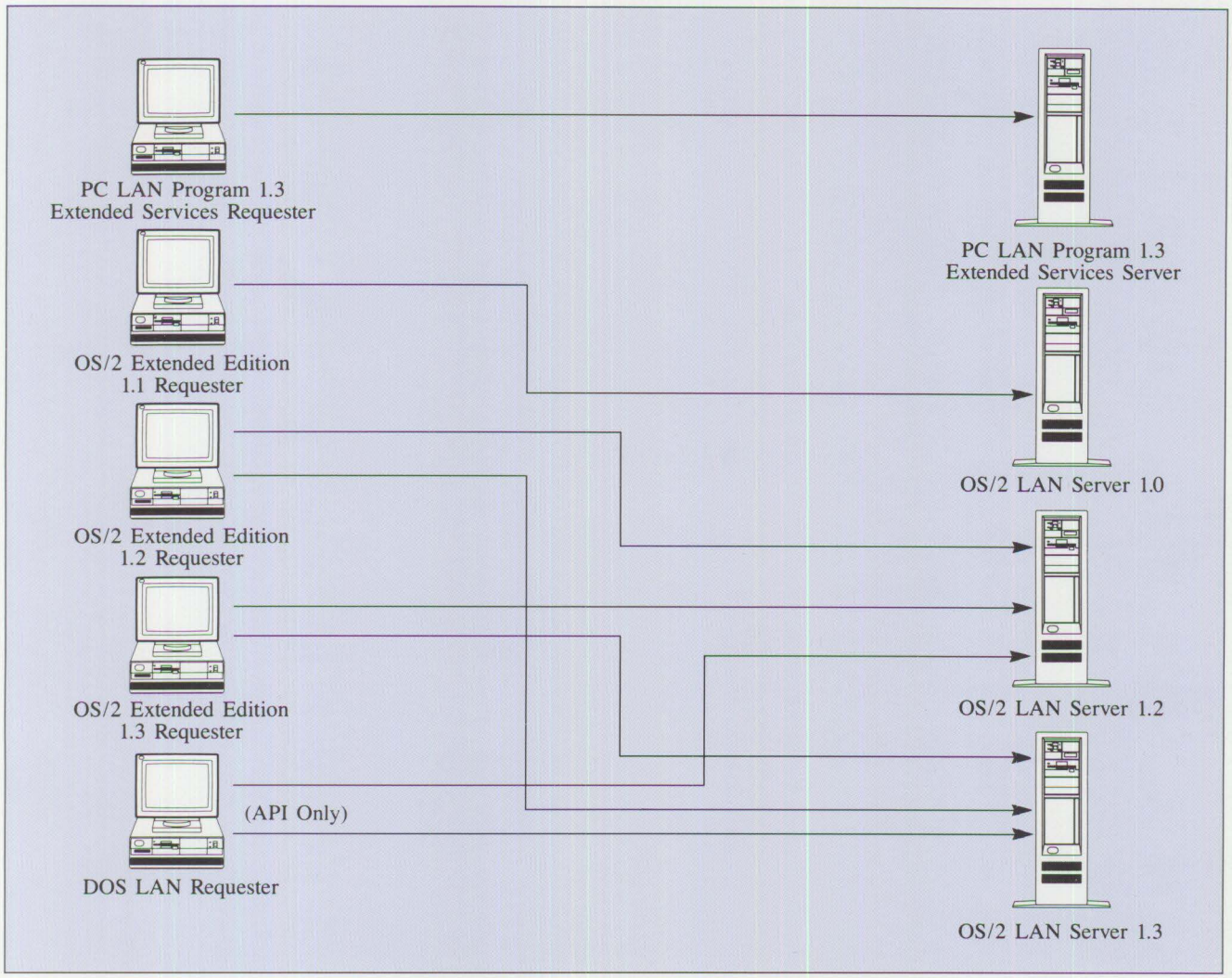


Figure 3. Valid Requester-to-Domain Logons

of whether the server is internal or external to the logon domain, but there is a subtle distinction.

This system guarantees that the same user account information is available to every server in the domain. At logon, the domain controller simply prevalidates that users are known to the domain. Therefore, if users successfully log on to a domain, all attempts to connect to available resources at any server in the domain will succeed. Likewise, all attempts to connect to servers outside the logon domain may fail if the userid and password are not valid at the external server or domain.

The version of the requester determines the domain(s) where users can logon. While there are some restrictions within the matrix, the inability to logon to a given domain should not be mistaken for inability to access resources within it. Once successfully logged onto a valid domain, access to any other server is possible. This also applies to administration of a server from a requester in many cases.

Figure 3 illustrates valid requester/domain logon relationships. There are two deficiencies to the matrix: neither PC LAN Program 1.3 nor OS/2 EE 1.1 requesters can log on to domains of OS/2 LAN Server 1.2 or greater. Keep this in mind when migrating to OS/2 LAN Server 1.2 or 1.3 domains.

A facility that automatically installs the DOS LAN Requester is incorporated into the server package as well as the DOS LAN Requester software. This facility is used with domains configured for PC LAN

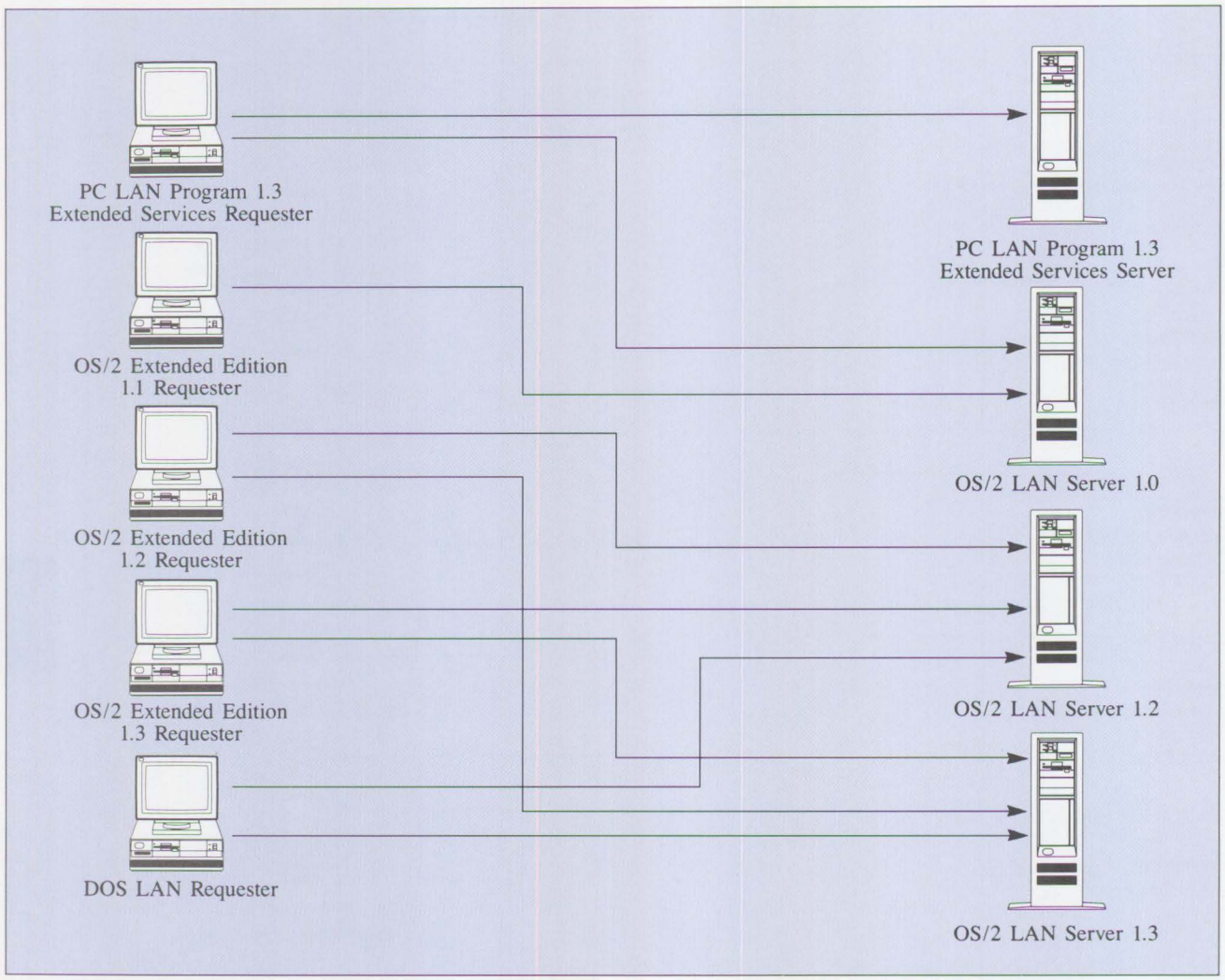


Figure 4. Valid Administration Paths

Program (PCLP) requesters migrating to OS/2 LAN Server 1.2 or 1.3. Once a domain has migrated to OS/2 LAN Server 1.2 or 1.3, users with PCLP simply log on as usual. Users are given the choice of automatically installing the DOS LAN Requester program from the server to the requester. The DOS LAN Requester is started, and users can now log on to the new domain. For additional details on this subject, refer to the article titled "A New LAN Requester for DOS Systems," which appears in Issue 3, 1990, of this publication.

Administration

The administration function allows authorized users to modify the objects and definitions that make up the domain. For example, an administrator may add and delete users from the domain, define resources of a specific server to be shared, and grant or restrict specific users' ability to access those resources. Administrators can perform these functions from a requester.

Figure 4 shows the administrative compatibility between requesters and servers. In general, DOS requesters do not have a user interface for performing administrative functions. In addition, OS/2 EE 1.1 requesters are allowed to administer the OS/2 LAN Server 1.0 only. There are APIs for developing programs that allow authorized users to perform administrative functions from DOS LAN Requester and OS/2 EE 1.1 requesters to OS/2 LAN Server 1.2 and 1.3.

Server and Requester Types

Figure 5 shows that all requesters can connect to resources on all servers. As shown in Figure 3, users

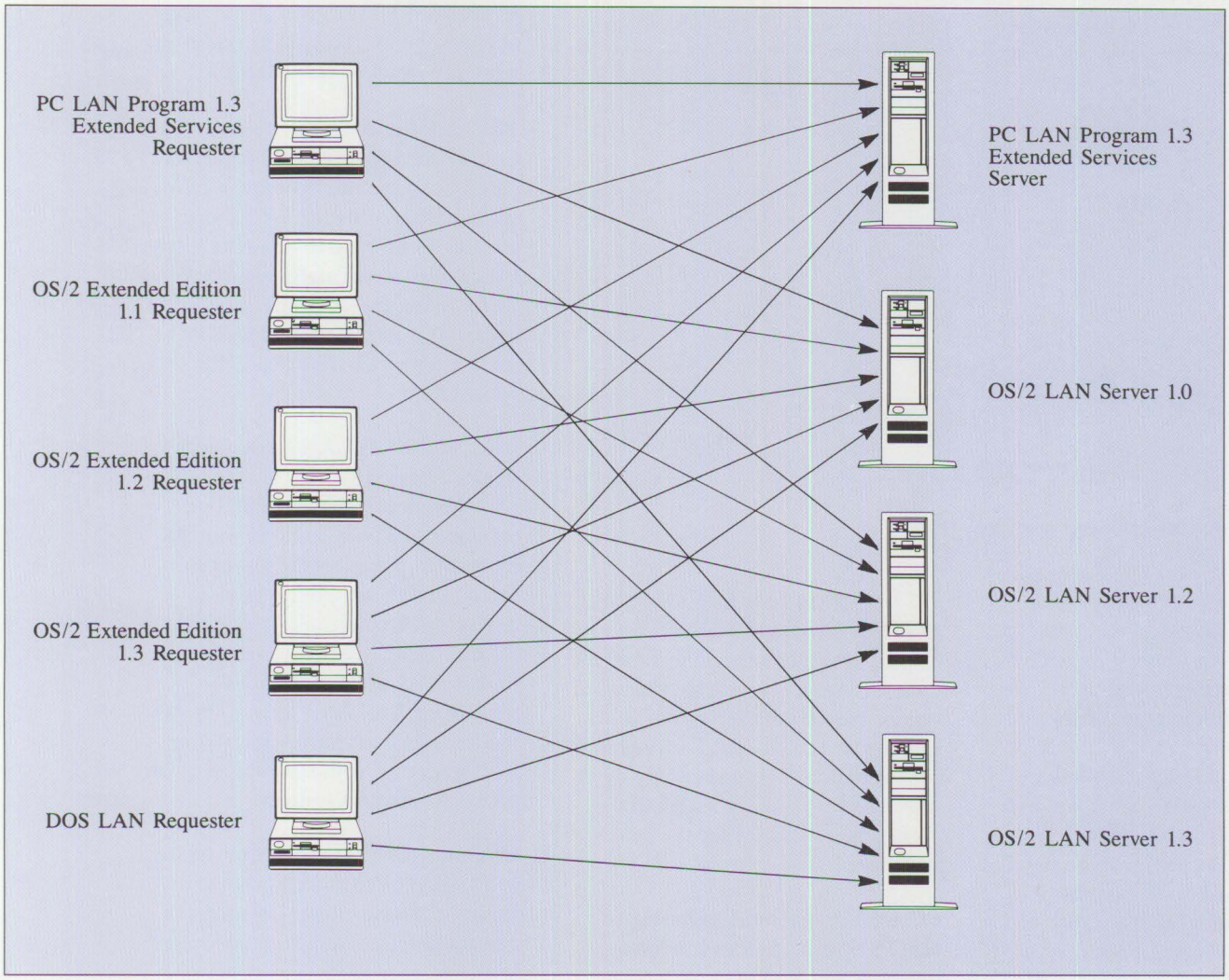


Figure 5. Valid Resource Connection Paths

must log on to a valid domain. Subsequently, access to any resource on any server on the network is possible. The key is making the connection properly.

The LAN products discussed have two types of server security – resource- and user-based security. All versions of the PCLP support resource-based security only. Although version 1.3 with Extended Services requires user logon, the basic system is resource-based. OS/2 LAN Server versions 1.0, 1.2, and 1.3 support user-based security only.

Resource-based security uses an optional password and access list for each shared resource, both of which are assigned by the administrator. The resource password must be specified to make a connection. Subsequently, the access list determines which functions are allowed (Read Only, Read/Write, and so forth). This design requires the administrator to make the password known to all users desiring connection. In addition, all users connecting to the same resource are granted the same access.

When user-based security is employed, connection is made once a valid userid and password are transmitted to the indicated server. Once connected, the actions allowed (read, write, and so forth) are defined by the resource access list. This list contains individual entries for selected users and groups. Connections to a resource are granted for a valid userid and password

Do you have a password at your logon domain?	Do you have a password at the other domain, or does the server require a password?	Do the two passwords match?	Then specify this:
Yes	Yes	Yes	password or blank
Yes	Yes	No	other domain password
Yes	No	N/A	null password ("")
No	Yes	N/A	other domain password
No	No	N/A	null password or blank

Figure 6. Table for Determining Proper Passwords

even if users are not in the access list. Once the connection is made, it is possible that no access is granted. This would be the case if users don't appear in the access list.

The two basic models of LAN requesters are machine name and user-based requesters. All versions of PCLP requesters are machine-based. In other words, no userid information is transmitted to the server as part of the resource connection process (NET USE). Userid requesters, which consist of all OS/2 requesters and the DOS LAN Requester, require users to logon prior to establishing connections. The userid and password are stored at the requester and transmitted to each server when attempting to establish an initial connection.

Resource Connection

Here are some specific types of requester-to-server connections. Figure 6 can serve as a reference for the following sections.

PC LAN Program Requester to PC LAN Program Server: In this example, a machine name requester connects to a resource-based server. Neither the server nor requester processes userid information for validation. The server only determines that users have supplied the correct password for the particular resource.

PC LAN Program Requester to OS/2 Servers: In this example, a machine name requester connects to a user-based server. There is an inconsistency between the information transmitted and the information required by the server. The user-based server allows a connection only if the userid and password are valid. The requester requires no user logon and cannot supply this information directly.

Here is how the inconsistency is resolved. When starting the PCLP, users must specify a machine name. Upon connection, this name is transmitted to the server. A password, if required, is specified on the NET USE command. The server uses the requester's machine name as the userid. Prior to allowing connection, the machine name and password are checked for validity.

The administrator must define userids and passwords to the server based upon the predetermined machine names of the PCLP requesters. One drawback to this operation is that users must always start the PCLP using the same machine name. This generally implies that users are restricted to a specific workstation.

OS/2 and DOS LAN Requesters to OS/2 Servers: This is an example where a user-based requester connects to a user-based server. As mentioned previously, users must log on before any connections to any server are allowed. This is necessary for the userid and password to be stored at the requester for transmission to other servers. In this model, users can move freely from workstation to workstation and gain access to server resources.

OS/2 and DOS LAN Requesters to PC LAN Program Servers: In this example, a user-based requester connects to a resource-based server. This also represents an inconsistency between the information transmitted by the requester and that expected by the server. Users must successfully log on prior to attempting the connection, but the server is interested in the resource password only.

This inconsistency is handled by protocol negotiation. This negotiation allows the requester to know if the server is resource- or user-based. When a user-based requester connects to a resource server, userid information is ignored. As a result, this case is identical to the PCLP Requester to the PCLP Server connection.

PC LAN Program Extended Services: Although PCLP Extended Services looks like a user-based system, it is a machine name requester and a resource server model identical to PCLP Base Services. Extended Services requires user logon and validation before resources are displayed on menus. But the basic security of the server is resource level. When users select a resource to connect to, the workstation receives the correct password for the resource from the server (assuming the user has permission). The password is randomly generated by the system when the administrator defines the resource. But, in theory, any workstation of any type can access the resource if the password is known.

In fact, a PCLP Extended Services server validates and responds to all requesters outside of the server's domain the same way as a Base Services server. Also, PCLP Extended Services requesters are handled as machine name requesters by all servers in the network. Thus, there is little distinction between Base Services and Extended Services.

Internal Connections: An internal connection is from a requester to a server that is part of the domain where the user is logged on. This concepts applies to user-based requesters only.

PCLP Base Services requesters cannot be considered part of any domain. These requesters are discussed in the following section, "External Connections."

Logon validation and duplication of userid and password information by each server in the domain guarantees that connections to any server in the domain will succeed. This is true for connections made when users select an internal alias (an alias that defines a resource within the domain) from a menu or command line. When an alias is selected from the menu, there is no opportunity, nor is it necessary, to specify a password. In fact, if a password is specified with a NET USE command, and is different from the logon password, the connection will fail.

Using external alias definitions is a common way to make external connections.

External Connections: Connections to servers outside of the logon domain are possible, but not always as straightforward as those inside the domain. The logon userid must be defined at the target server when connecting to any user-based server (all OS/2 servers). Because the server is outside the logon domain, there is no guarantee that the userid will be valid at the server. Administrators must be careful when managing user accounts of multiple domain networks. Resource-based servers (DOS servers) do not process the userid; therefore, the userid is never an issue when connecting to a resource-based server.

Assuming that the userid is valid at the target server, the proper password is also required to complete the connection. Because the target server is outside the logon domain, there is no guarantee that the password for the userid is the same one used to log on. The best way to ensure successful connections is for administrators and users to keep their passwords the same across the domains. If passwords are different across domains, connection is still possible if the correct password is explicitly specified when the connection is attempted.

The override password specified on the NET USE command will be transmitted to the target server regardless of the password used at logon. If the userid at the target server has no password, a pair of double quotes can be used to send a null password.

Using external alias definitions is a common way to make external connections. The external alias defines the path to a specific server and resource outside the logon domain. When the administrator defines the external alias, batch files (ALIAS.BAT for DOS, and ALIAS.CMD for OS/2) are created. The administrator then edits the files to contain the proper NET USE command for the desired connection. When the alias is selected, the batch file is executed and the connection is made.

The administrator can insert an override password in the batch file. Because the password will be the same for all users, this can be useful if the target server is a resource-level server.

Connecting to user-based servers with an external alias can present a problem if all users of that alias do not keep their logon password synchronized with the target server. This is especially true if the requester runs on OS/2. The problem is that the NET USE statement placed in the .CMD file for the alias can contain either a password for all users or no password at all. It is un-

likely that a particular password will work for all users.

Here are a couple of solutions to this problem.

OS/2 LAN Server Versions 1.2 and 1.3 have a function called guest account. Basically, a guest account is set up by the administrator of a particular server. Once established, the server will accept a connection from any requester on the network if the logon userid (or machine name, for PCLP requesters) is not known to the server's domain. This "generic" account does not compromise the security of resources at the server, because users are allowed to perform those operations (such as read, write, and delete) specifically allowed by the guest account. Thus, the guest account can be a graceful solution when defining an external alias that is to be made generally available. This could be for a resource, such as a set of common tools or a printer.

The guest account solves the problem equally as well for DOS LAN Requesters, but there is another approach worth exploring. Using the NET USE command for DOS requesters, an asterisk (*) may be placed instead of an override password. The asterisk (*) indicates that users will be prompted for the password. So the solution is quite simple: the administrator incorporates this syntax into the .BAT file created for the alias, and then users specify their own password. Users no longer need to keep their passwords synchronized across domains, and there is no need to use the guest account. Figure 6 shows the proper use of passwords for the various types of requesters based on the logon password and the server type.

Printer Management Considerations: With the release of OS/2 Version 1.3, many changes were made to improve print manager usability and spooler performance. As a result of these changes, minor modifications were made to LAN Server 1.0 and 1.2, which allows 1.3 requesters to manage 1.0 and 1.2 queues (and vice versa). Corrective service diskettes reflecting these changes are now available for both versions.

ABOUT THE AUTHORS

Roy Feigel is an advisory programmer in IBM's Entry Systems Division in Austin, Texas. He is the lead designer for the OS/2 LAN Server 1.2, Extended Edition 1.2 LAN Requester, and DOS LAN Requester. In 1982, Roy joined IBM in Austin as a programmer assigned to text applications for the 5520 Administrative System. In 1985, he became responsible for the server component of the PC LAN Program 1.1, and has since been involved in design and development of LAN products. Roy has been a presenter for IBM's Technical Coordinator Program television broadcasts to its customers and has been an advisor for customers' large network designs. He received a B.S. in computer science from the University of Southwestern Louisiana.

Steven French is a senior associate programmer at IBM's Entry Systems Division in Austin, Texas. He joined IBM in 1989 and presently is the technical interface to LAN development. Steve received a B.A. in computer science and an M.S. in electrical and computer engineering from Rice University.

Carolyn Easter is an advisory programmer in IBM's Entry Systems Division in Austin, Texas. She joined IBM in 1980 and was assigned to program the files and data base system on the 5520 Administrative System. She has a B.S. in mathematics from Southern Methodist University and a B.A. in computer science from the University of Texas at Austin.

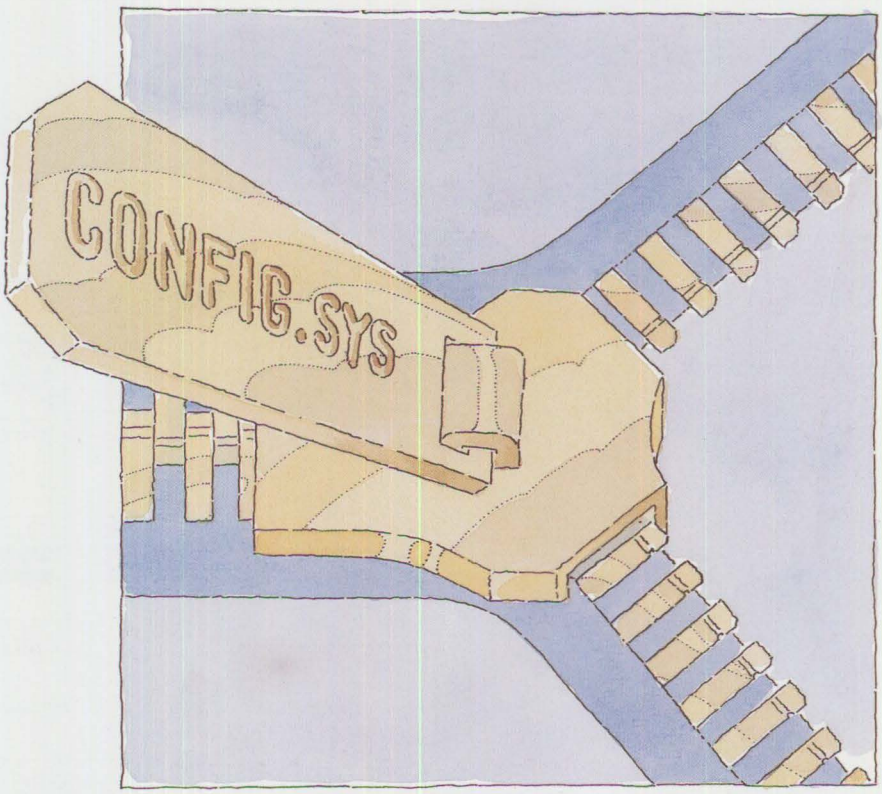
Running DOS LAN Requester and Novell NetWare Concurrently

Mike Granelli
IBM Corporation
Boca Raton, Florida

This article shows how a single DOS user can connect to both the OS/2 LAN Server 1.2 and Novell NetWare 286/386 at the same time. The key is the CONFIG.SYS file. The ability of requesters to utilize the strengths of both networks as well as IBM hosts is now easily demonstrated.

Editor's note: The information contained in this article represents the results of a technology demonstration. This implementation has not been formally tested by either IBM or Novell, and is offered as is. No additional installation or usage support is implied.

"How can I log on to IBM's LAN Server 1.2 and still preserve my access to resources on Novell servers?" This question is often asked by many LAN users. The answer to this question is a simple demonstration of how to modify the CONFIG.SYS file so a single DOS workstation can access both NetWare servers and LAN Server 1.2. No additional software is required. Access to IBM hosts can also be concurrent in this "interoperability" mode. Interoperability is the ability to operate concurrently with more than one platform.



Configuration

The configuration for this environment is all 386-based PS/2s, as shown in Figure 1. The IBM server runs LAN Server 1.2, while the Novell server runs NetWare 386. Printers are attached to both servers. The DOS Requester is running DOS 4.01, has 2 MB of memory with Expanded Memory Support (EMS). The network used is a 16/4 Mbps IBM Token Ring Adapter/A running at 16 Mbps.

The testing was performed at the following levels:

- OS/2 Extended Edition 1.2 – Corrective Service Diskette (CSD) level 4053
- OS/2 LAN Server 1.2 – CSD 4053
- DOS LAN Requester – CSD 4053
- Personal Communications/3270 – release level 1.01
- Novell NetWare 386 – versions 3.0 A and 3.1 A
- Novell Advanced NetWare 286 – versions 2.12 and 2.15

CONFIG.SYS is the Key

Figure 2 lists the CONFIG.SYS statements that allow concurrent connection to IBM and Novell servers. The changes in bold type have been made to the default CONFIG.SYS that is created when the DOS LAN Requester (DLR) is installed. The first change is the **O=Y** and **ES=1** parameters that were added to the DXMT0MOD.SYS device driver. This extra SAP or Service Access Point (ES=1) allows the NetWare protocol stack to coexist with NETBIOS within the DOS worksta-

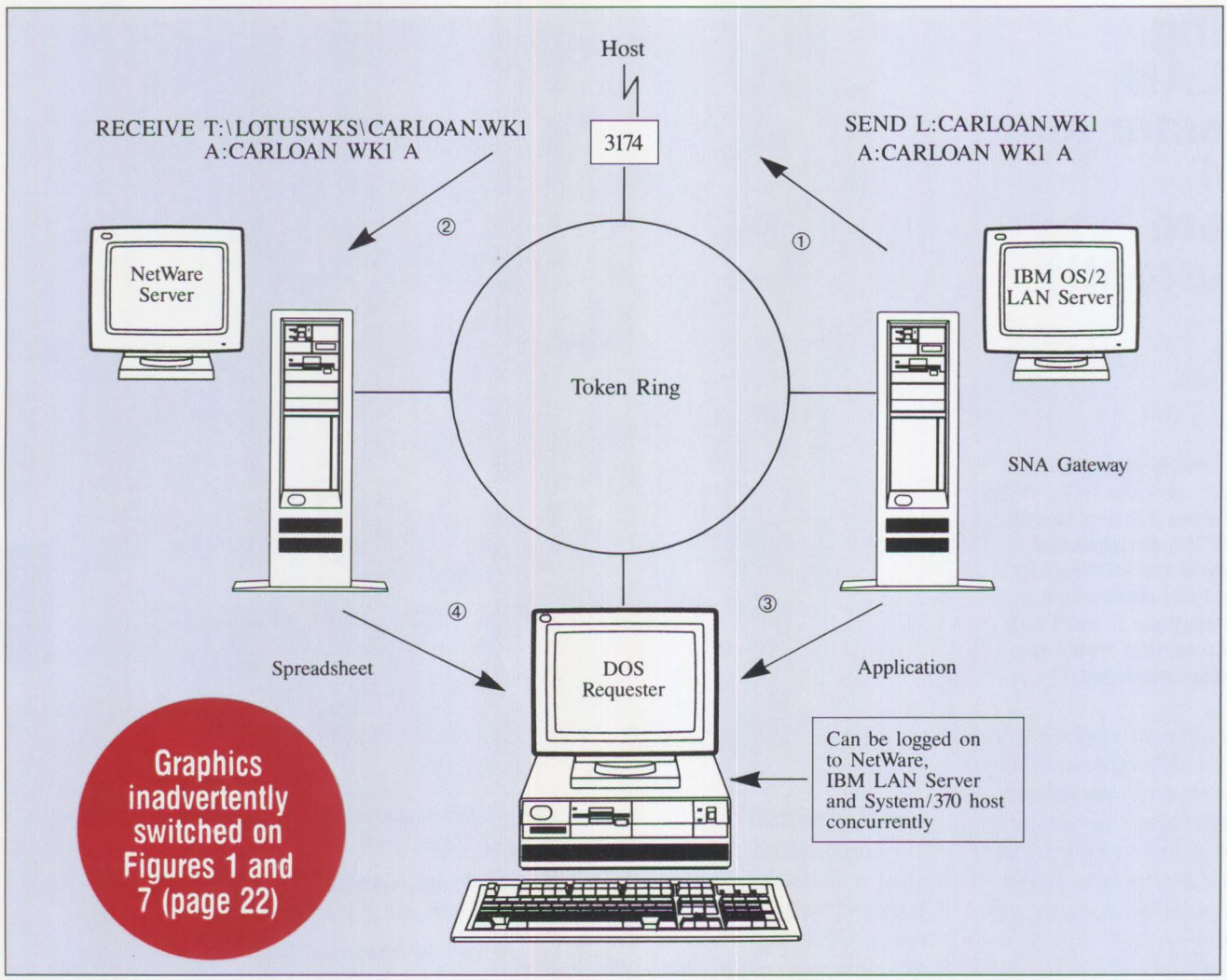


Figure 1. Interoperability LAN Configuration

tion. Open=Yes **(O=Y)** ensures that this extra SAP is available when the CONFIG.SYS is installed and opens the token ring adapter. The minimum configuration for connecting the IBM and Novell networks is:

```
O=Y   ES=1
```

One important point to remember: The concurrent connection to IBM and NetWare is handled through a single token ring adapter card. The extra SAP acts as the plug into the second software stack; that is, the IPX/NET4 shell. In this way, IBM and NetWare networks each have their own SAP that allows the two shells to communicate to their respective servers. The device drivers that talk to the token ring adapter are found in the LAN Support Program (LSP). The IBM Token-Ring Adapter Card, LSP, and the IEEE 802.2 make these connections work, and are required to make interoperability possible.

The relationship between hardware and software is shown in Figure 3. The extra SAP for NetWare IPX has been added. You might notice that the extra SAP has been added as a "non-NETBIOS" SAP. When extra SAPs are added, they are added as non-NETBIOS SAPs. NetWare and the IBM Personal Communications/3270 (PC/3270) program do not use NETBIOS. The device driver for the configuration shown has the following parameters:

```
O=Y   ES=2   EST=1
```

This is how the CONFIG.SYS may be modified in order to simultaneously connect among an IBM LAN server, Novell NetWare

server, and a System/370™ host using PC/3270. An additional SAP **(ES=2)** and link station **(EST=1)** must be added for the 3270 connection. This link station is used for connection-oriented communications with remote devices, which in this case is PC/3270. Link stations exist at the end of logical connections, and they send and receive data with other link stations. Now the NetWare shell and the 3270 emulator have their own SAP.

```
BREAK = ON
BUFFERS = 20
FILES = 50
LASTDRIVE = R
FCBS = 16,8
SHELL = C:\COMMAND.COM /E:2000 /P
DEVICE = \DXMA0MOD.SYS 001
DEVICE =\ DXMC0MOD.SYS 400000000012
DEVICE = \DXMT0MOD.SYS S=12 C=12 ST=12 O=Y ES=1
DEVICE = C:\DOSLAN\HIMEM.SYS
```

Figure 2. CONFIG.SYS for IBM/Novell Interoperability

The second change is an administrative one. By changing the LASTDRIVE= statement in the CONFIG.SYS, NetWare now has virtual drives available for its use. If the default CONFIG.SYS is taken (LASTDRIVE=Z), it does not allow for NetWare drive mappings. In Figure 2. LASTDRIVE was set to **R**, because this IBM LAN Server was sharing drives up to **R**. This drive assignment varies according to individual needs. Note that the IBM LAN Server looks at LASTDRIVE as the final virtual drive to assign while NetWare uses LASTDRIVE as a starting point to assign virtual drives. If these two LANs used the same drive assignment convention, this interoperability would not be possible.

Finally, memory utilization must be factored into this equation. The last line of the CONFIG.SYS shows the use of the HIMEM.SYS driver found in the DOSLAN subdirectory. This will move almost 40 KB of the DLR into extended memory, which allows more room for applications. Memory below 640 KB is a concern in this environment because two network stacks are being loaded. Further relief can be achieved by using the EMS drivers in place of HIMEM, especially

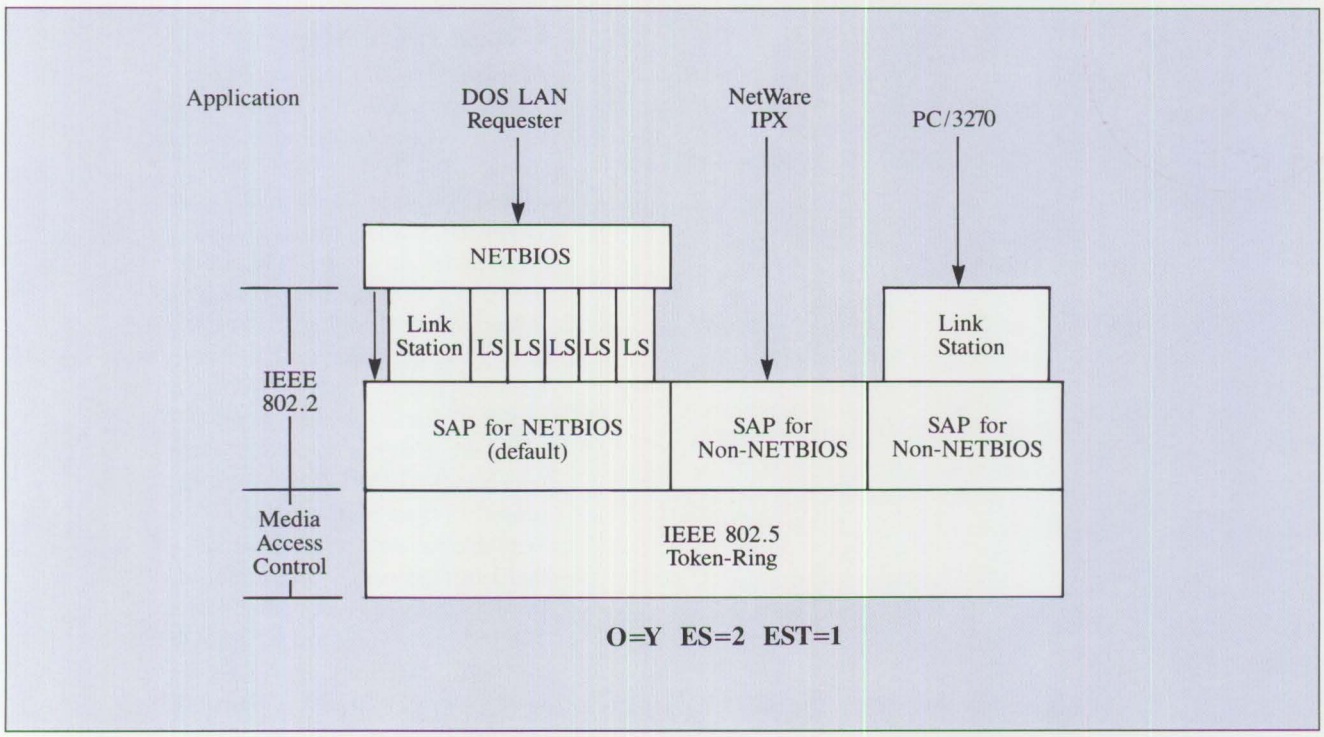


Figure 3. Applications, NETBIOS, IEEE802.2, and Hardware Relationship

```
@ECHO OFF
SET COMSPEC=C:\DOS\COMMAND.COM
PATH=C:DOSLAN;C:\DOS;C:\NETWARE
REM APPEND=C:\DOS
PROMPT $P$G
YNPROMPT Y N 30 START DOS LAN REQUESTER (Y/N)?
IF ERRORLEVEL 1 GOTO NODLR
NET START
IF ERRORLEVEL 1 GOTO NODLR
CALL INITFSI.BAT
NET
:NODLR
YNPROMPT Y N 30 START NETWARE REQUESTER (Y/N)?
IF ERRORLEVEL 1 GOTO NONW
IPX
NET4
PROMPT $P$G
S:
LOGIN WS1
PATH Z:.;Y:.;C:\DOS;C:DOSLAN;C:\
:NONW
```

Figure 4. AUTOEXEC.BAT Example

Status	Local device	Network name
OK	S:	\\NETWARE386\SYS
OK	T:	\\NETWARE386\SYS
OK	U:	\\NETWARE386\SYS
OK	V:	\\NETWARE386\SYS
OK	W:	\\NETWARE386\SYS
OK	Y:	\\NETWARE386\SYS
OK	Z:	\\NETWARE386\SYS
OK	LPT2:	\\NETWARE386\PQ1
OK	D:	\\OS2DC\CLOCK
OK	I:	\\OS2DC\DW5DOC
OK	J:	\\OS2DC\BRIDGE
OK	K:	\\OS2DC\DW5
OK	L:	\\OS2DC\DOSLOTUS
OK	M:	\\OS2DC\COURIER
OK	N:	\\OS2DC\LOTUS123
OK	O:	\\OS2DC\LOTSHARE
OK	P:	\\OS2DC\WDPERF
OK	Q:	\\OS2DC\USER1
OK	R:	\\OS2DC\IBMLAN$
OK	LPT1:	\\OS2DC\LPT1Q

Command completed successfully

Figure 5. NET USE Command Shows Mapping of all Drives

when the application recognizes EMS and takes advantage of it; for example, DisplayWrite 5.0™.

AUTOEXEC.BAT Starts the Requesters

It is important that the DOS LAN Requester is started first, followed by the NetWare stack (IPX/NET4). Figure 4 contains an example of an AUTOEXEC.BAT that accomplishes this goal. The DLR has been started using the default DOSLAN.INI file. The user logs on, then exits to DOS, and the NetWare IPX and NET4 shell are loaded; log-on takes place to complete the AUTOEXEC.BAT sequence. You are now logged on to both servers concurrently! At this point, you must *not* log off from the IBM server or you will lose all access to it and won't be able to log on again without rebooting. This is because paths to the IBM server have been lost.

Virtual Drive Maps

After successfully logging on to both servers, a map is needed to show all available shared resources. Because the IBM network was started first, it has kept track of all virtual drive assignments including NetWare's. As shown in Figure 5, the IBM command **NET USE** gives this complete mapping. Drives S through Z, including LPT2, have been mapped to the NETWARE386 server. Drives D through R plus LPT1 are assigned to the OS2DC, which is the IBM LAN Server. All resources from both servers are available to users transparently, either from the command prompt or from the Served Application screen. The IBM user shell can be accessed by simply executing the **NET** command. Novell's **MENU** command can also be used. Applications residing on the IBM LAN server and the

NetWare server can be run from either menu.

A benefit of this transparency is the ability to run applications from one server and load work files from the other. Print jobs can be spooled to either server. Files and applications can be copied from one server to the other.

Memory Availability

Memory is always an important consideration when running in a DOS environment. Figure 6 shows 447 KB available in the concurrent mode with both requesters installed. Memory availability for both the HIMEM and the EMS configurations is also shown. Many standard spreadsheet, word processing, and E-mail packages can run with this available memory. Also, there are memory manager packages available that make more space available within the 640 KB limitation of DOS. Workspace over 535 KB has been demonstrated using these methods.

Host Connectivity

An even larger opportunity for interoperability is the host environment. Earlier in this article, configurations were discussed that offered host connectivity concurrent with the DOS LAN Requesters. As shown in Figure 7, the PC/3270 emulation software can be loaded on the workstation; files can then be transferred between a server and the host (1); downloaded to the other server (2); the application is loaded from the IBM server (3); and the spreadsheet is retrieved from the NetWare server (4). All transfers are under the control of the single DOS requester.

Guidelines

When implementing in this environment, it is important to consider the following:

- Extra SAPs must be defined when opening the token ring adapter
- LASTDRIVE= must have a drive letter assignment less than Z
- The DOS LAN Requester must be loaded before NetWare's requester
- After logging off the IBM network, further logons to the IBM network are not possible without rebooting

Conclusion

This demonstration shows some very important concepts. First, IBM and Novell LANs can coexist on the same token ring. From a single DOS requester, a user can log on to both servers and access all resources available on both servers. This is particularly important to users migrating to OS/2 LAN Server that want to take advantage of its capabilities while preserving a connection to their company's installed base of Novell LAN resources. At the same time, this capability will allow Novell users additional functions available to IBM LAN Server DOS Requesters, such as IBM emulators to access IBM hosts. These users allow expansion of their existing capabilities into IBM's SAA platform while pro-

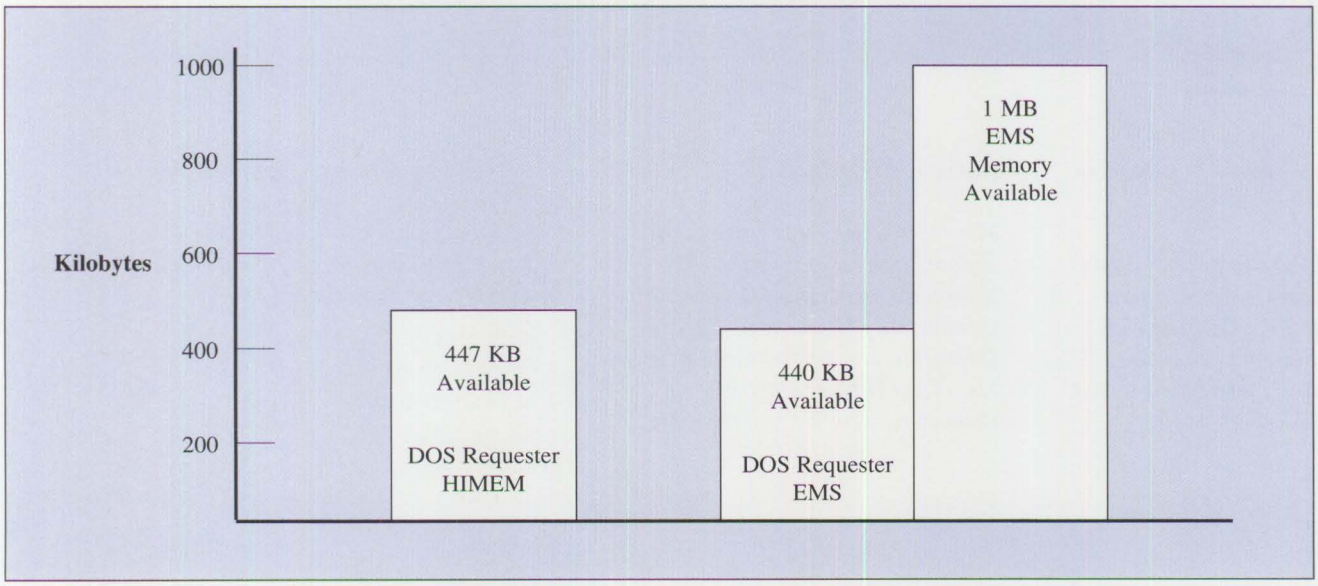


Figure 6. Memory Available for DOS Applications as Tested

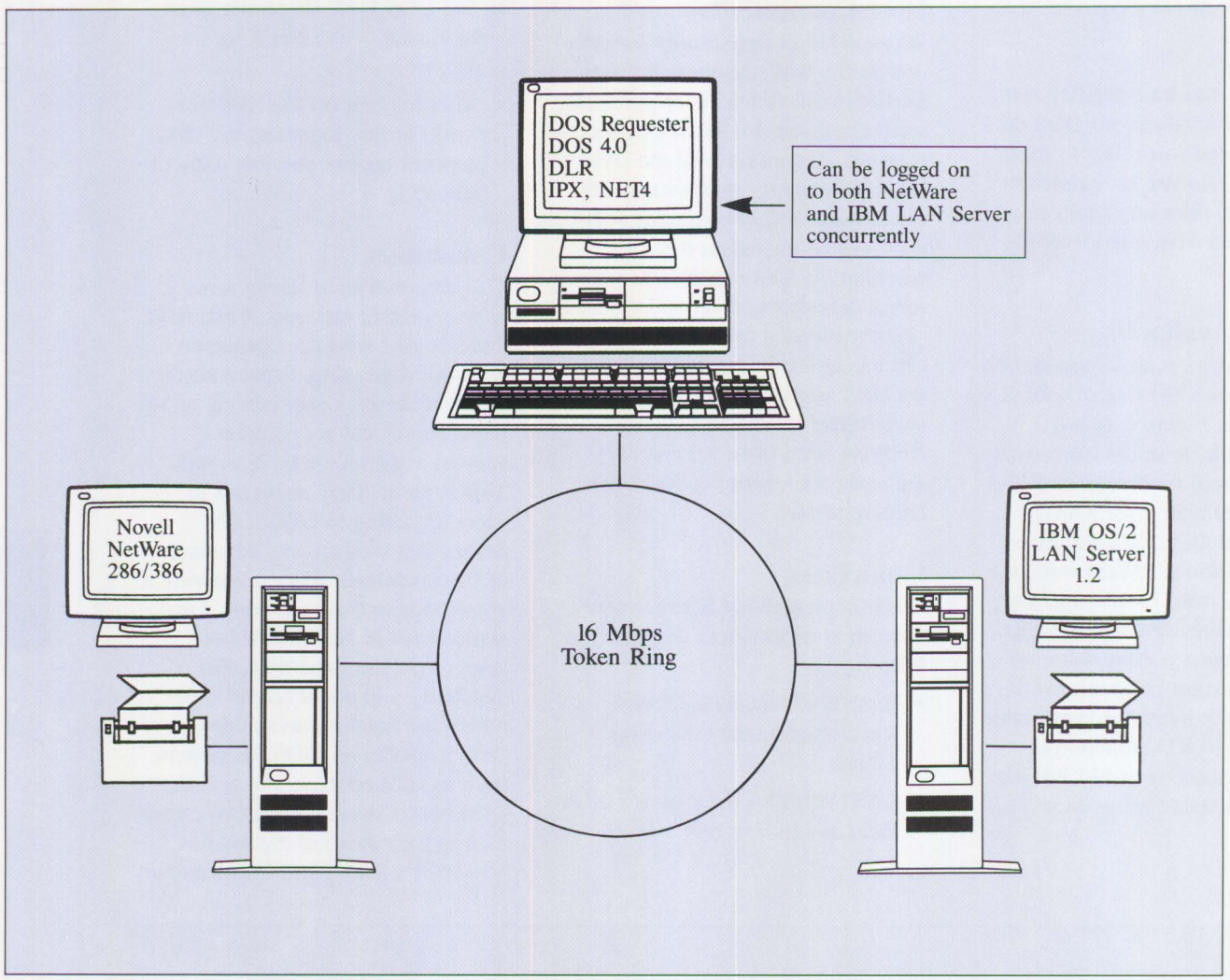


Figure 7. LAN/Host Scenario

tecting the investment in their current NetWare server.

For more information on the LAN Support Program, refer to *Guidelines for Setting Local Area Network (LAN) Support Program Parameters For Use With Selected IBM Products*, GG22-9430.

ABOUT THE AUTHOR

Mike Granelli is a senior market support representative for IBM's National Distribution Division PC Connectivity Marketing Support Department in Boca Raton, Florida. He joined IBM in 1981 in Princeton, New Jersey. His IBM assignments have included planning for biomedical systems products and lead planner for the PC/XT Model 089. Mike received a B.S. in physics from Drexel University.

Breaking the 640 KB DOS Memory Barrier

Pylee Lennil
IBM Corporation
Boca Raton, Florida

This is an in-depth discussion of how to break the 640 KB DOS memory limit.

The major problem with applications running under DOS is limited memory space. Even though 1 megabyte (MB) of memory can be accessed under real mode, only the first 640 kilobytes (KB) are available to DOS and DOS applications. This is because memory between 640 KB and 1 MB is occupied by video buffers, hardware adapters, and ROM BIOS. In fact, available memory for applications is even less than 640 KB, because DOS and installable device drivers occupy portions of the memory. This leaves the applications with much less than 640 KB, especially if large applications like network programs or emulation programs are already resident in the memory.

One way to solve this memory limitation is to load portions of the application initially, and load other portions on demand from disk. This overlay scheme has its own drawback because of the slow speed of the disk access, seriously affecting the performance.

Therefore, the following memory management schemes can be used for applications to break the 640 KB memory barrier:

- Expanded Memory Management
- Extended Memory Management
- DOS Extender

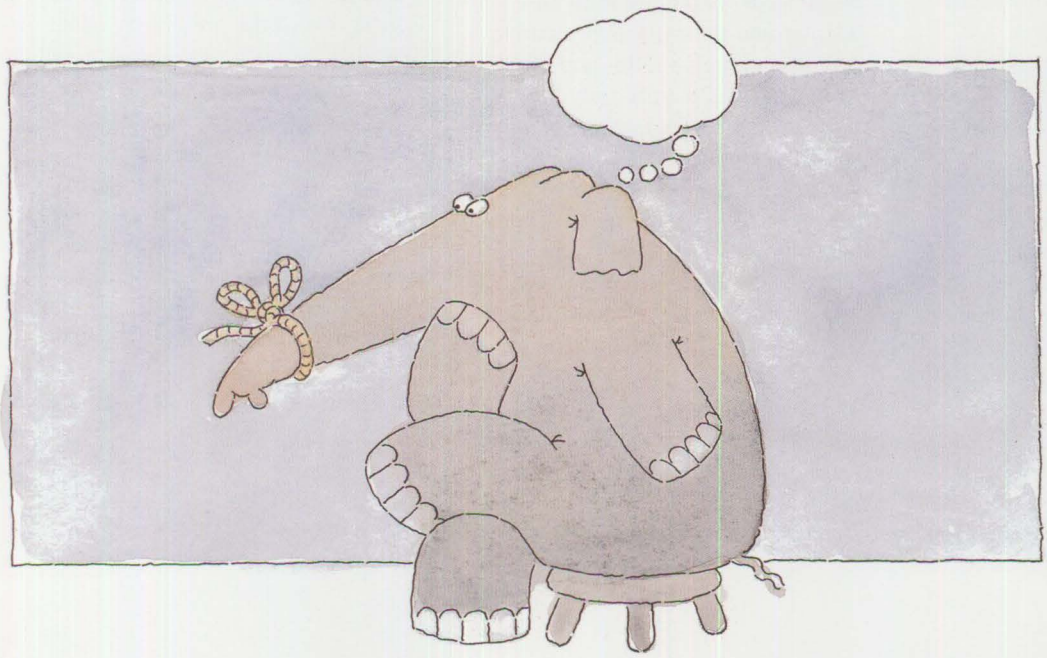
System physical memory can be divided into the following memory areas:

- Conventional Memory – occupies from 0 to 1 MB of the physical memory (Figure 1).
- Base Memory – occupies up to 640 KB of conventional memory. This memory can be addressed directly by DOS and DOS applications.
- Extended Memory – occupies from 1 MB and higher up to the available physical memory on the system. This area is not directly addressable by DOS or DOS applications, but can be accessed using BIOS function call or Extended Memory Specification (XMS).
- Expanded Memory – is a part of Extended Memory that can be accessed using the Expanded Memory Specification (EMS).
- High Memory Area – is a 64 KB block of memory available at the beginning of the second megabyte. Applications can access this area using the Extended Memory Specification (XMS).

Now that we have learned about the different types of memory, let's examine each memory management scheme in detail and see how DOS applications can use these schemes.

Expanded Memory Management

The Expanded Memory Specification (EMS) is a memory management scheme defined jointly by Lotus®, Intel, and Microsoft to break the DOS 640 KB memory restriction. This is done by allocating one or more 16 KB physical pages between 640 KB and 1 MB (Figure 2).



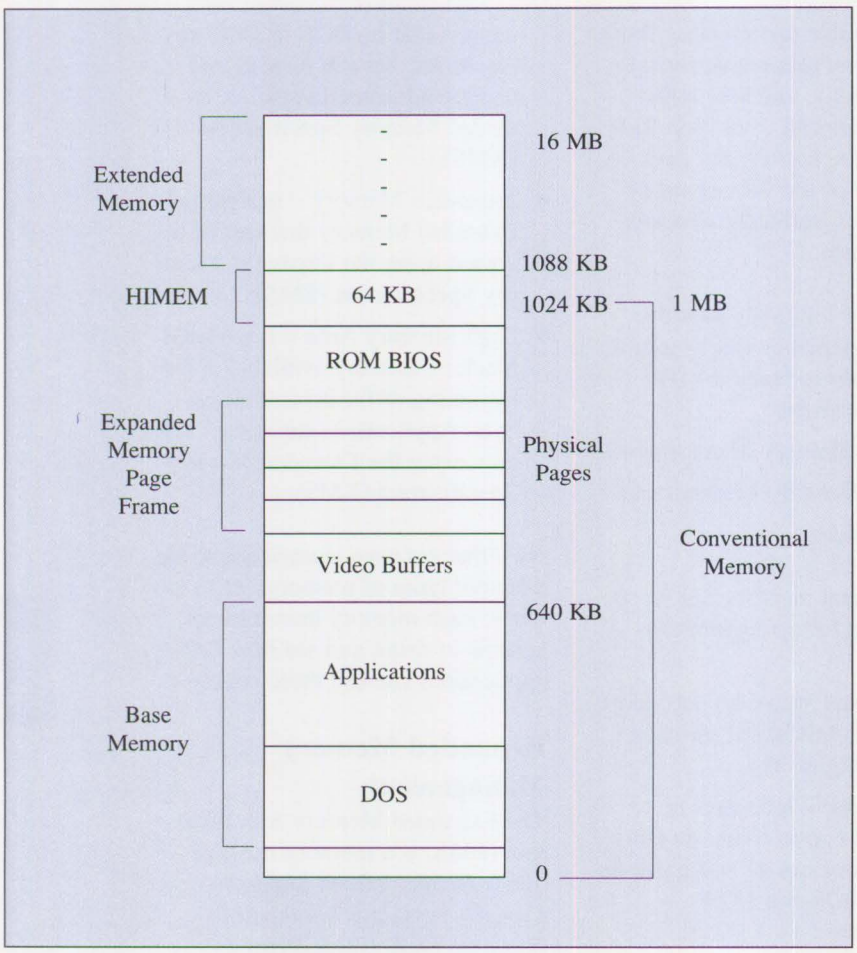


Figure 1. System Memory Areas

A page frame consists of four contiguous physical pages with pages numbered from 0 through 3. As shown in Figure 2, the expanded memory manager divides memory over 1 MB and memory below 640 KB into 16 KB logical pages. Also in Figure 2, the expanded memory driver divides the memory over 1 MB into 16 KB logical pages. Using physical pages as a window to the logical pages, the application can access one or more of the logical pages by mapping logical pages to physical pages (Figure 2).

The Expanded Memory Management (EMM) consists of a memory adapter and an expanded memory driver. The 8088- and 80286™-based systems require both an adapter and the expanded memory driver, although 80386- and 80486-based systems require only the driver. The memory adapter that can be plugged into the system contains memory chips and special I/O ports to access the memory. The expanded memory driver can be loaded using the DEVICE= command in the CONFIG.SYS file during system load and configuration.

The expanded memory driver for 80386 and 80486 systems is a software emulator capable of providing expanded memory support without any special memory adapter. The emulator uses the Virtual 86 mode. Under this environment, the emulator runs in the protected mode while DOS and the applications run in the Virtual 86 mode. Using the protected-mode page table mechanism, the driver emulates the 16 KB physical pages below 1 MB and creates 16 KB logical pages above 1 MB in the extended memory area.

Expanded Memory Management provides these classes of services:

- Get Expanded Memory Manager (EMM) version
- Get status of expanded memory subsystem
- Get physical page frame address
- Get number of expanded memory pages
- Allocate expanded memory page
- Map logical page into physical pages
- Deallocate expanded memory pages
- Save and restore physical page map

The application interfaces with the Expanded Memory Manager using the software interrupt (INT 67H) with a function code in AH to request one of the previously listed expanded memory services. A detailed description and the calling sequence of commonly used EMS functions are shown in Figure 3.

Next, we'll examine how the application program can access the expanded memory. To access the expanded memory, the application uses the following steps:

1. Check if the expanded memory manager is installed.

This can be done by using either the open file method or interrupt

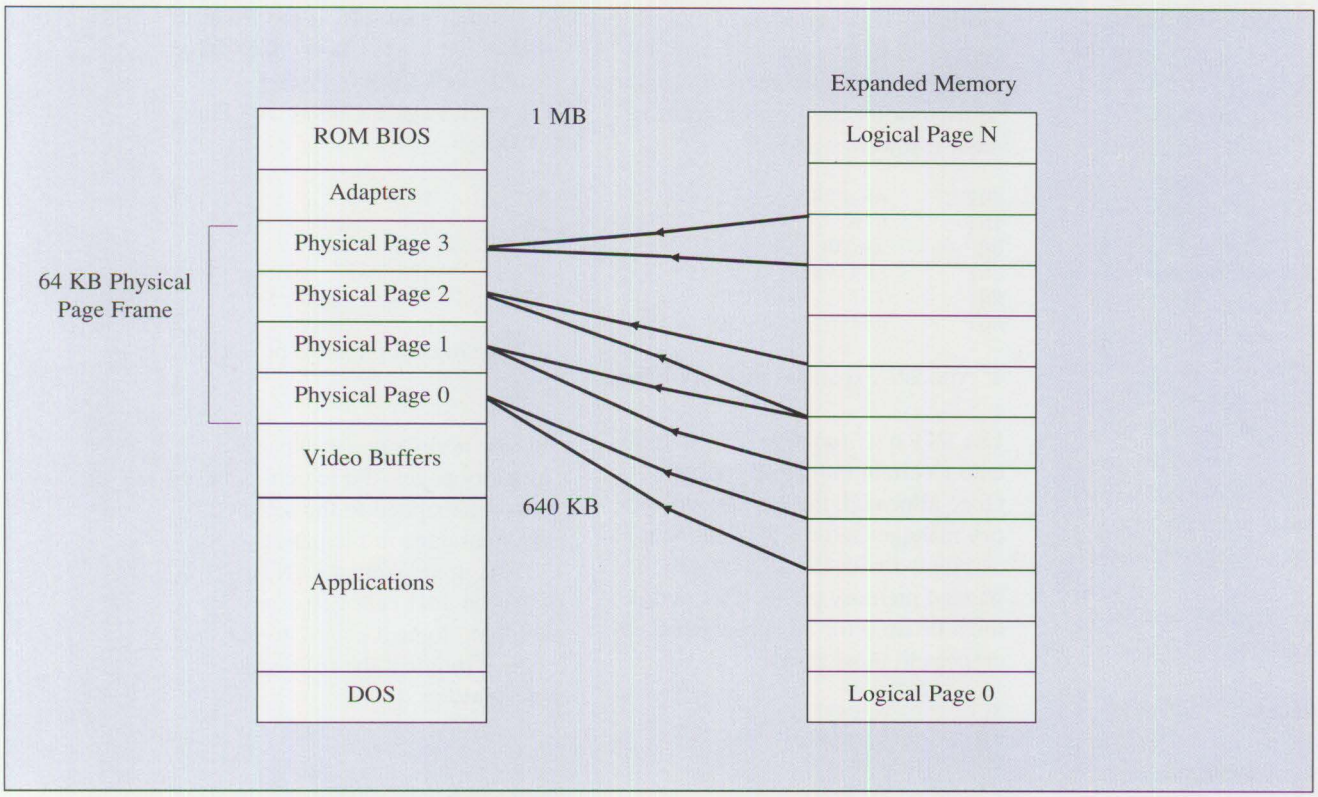


Figure 2. Expanded Memory Physical and Logical Pages

Function	Input	Output
Get EMM version	AH = 46H	AH = status AL = version
Get status	AH = 40H	AH = status
Get number of expanded memory pages	AH = 42H	AH = status BX = available pages DX = total pages
Get physical page frame address	AH = 41H	AH = status BX = frame segment
Allocate logical pages	AH = 43H BX = number of pages	AH = status DX = EMM handle
Map logical pages to physical pages	AH = 44H AL = physical pages BX = logical page DX = EMM handle	AH = status
Deallocate logical pages	AH = 45H DX = EMM handle	AH = status

Figure 3. EMS Functions

method. In the open file method, the application must try to open the expanded memory driver using its logical name using the DOS INT 21H function 3DH. If successful, make sure it is not a regular file by issuing an IOCTL (Function 44H) with subfunctions 0 and 7. The application must also close the handle using the DOS function INT 21H function 3EH to delete the handle once the checking is done. The interrupt vector method involves checking the beginning of the INT 67H handler for expanded memory manager signature "EMMXXX0." If the signature is found, then the expanded memory manager is loaded.

2. Check if the expanded memory hardware is available.

This is done using the INT 67H function 40H. Also use INT 67 function 46H to check the Expanded Memory Manager version number to decide whether the functions to be used are supported by the expanded memory driver:

```
MOV     AH,40H
INT     67H
OR      AH,AH
JNZ     ERR_EXIT
```

3. Check the amount of expanded memory available.

Use INT 67H function 42H to determine the total number of expanded memory pages and how many are available. If the number of available pages is less than the application needs, the expanded memory cannot be used or the application needs to request fewer pages:

```
MOV     AH,42H
INT     67H
OR      AH,AH
JNZ     ERR_EXIT
MOV     TOTAL_PAGES,DX
MOV     AVAIL_PAGES,BX
```

4. Allocate expanded memory pages.

Use INT 67H function 43H to allocate a certain number of pages. Once allocated, the expanded memory manager returns a 16-bit handle associated with the allocated expanded memory pages. This handle must be used for all subsequent references to these pages:

```
MOV     AH,43H
MOV     BX,NUM_OF_PAGES
INT     67H
OR      AH,AH
JNZ     ERR_EXIT
MOV     HANDLE,DX
```

5. Get physical page frame address.

Use INT 67H function 41H to get the base address of the physical page frame address. The page frame contains a maximum of four 16 KB pages (P0-P3). Using these four pages, an application can access a maximum of four logical pages associated with the given handle. From the base address, a segment address is assigned to all four physical pages. For example, if the base address returned is C0000, then Page 0 address is C0000, Page 1 C4000H, Page 2 C8000, and Page 3 CC000:

```
MOV     AH,41H
INT     67H
OR      AH,AH
JNZ     ERR_EXIT
MOV     FRAME_ADDRS,BX
```

6. Map logical pages to physical pages.

Before accessing the expanded memory pages, the logical page must be mapped to the selected physical page in the page frame. This is done using the INT 67H function 44H call. For example, mapping logical page 2 to physical page 1 requires the following instructions:

```
MOV     AH,44H
MOV     AL,PHYS_PAGE_1
MOV     BX,LOGIC_PAGE_2
MOV     DX,HANDLE
INT     67H
OR      AH,AH
JNZ     ERR_EXIT
```

Once the pages are mapped, logical pages can be read or written using the segment number and the offset within the given physical page. For example, if the page frame address is C000H, then the segment numbers of physical page 0 through 3 are C000H, C400H, C800H, and CC00H. To access data in logical page 2 involves accessing the physical page 1 using segment number C400H and offset within the segment. To clear the logical page 2 requires the instructions shown in Figure 4.

```
MOV   AX,C000H    ; Ax = Page frame address
MOV   ES,AX       ; ES -> Page Frame
MOV   AX,4000H    ; 16 KB per page
MUL   1           ; calculate offset to physical page 1
MOV   DI,AX       ; DI ->offset to physical page 1
XOR   AL,AL       ; AL -> 0; data to be written
MOV   CX,4000H    ; Set count
REP   STOSB       ; write zeros to the mapped page
```

Figure 4. Accessing an Expanded Memory Page

Extended Memory Management

Extended memory is the memory available over 1 MB. Unfortunately, this area is accessible only under

protected mode. To access this area, the program has to switch from real to protected mode, and after using the extended memory, it returns to the real mode. This process involves creating the protected-mode descriptor tables. It accesses memory using the protected-mode selectors instead of segment ID as in the case of real mode. Fortunately, ROM BIOS provides two functions: INT 15H function 88H (Get Extended Memory Size) and INT 15H function 87H (Move Extended Memory Block). Using function 87H, blocks of data can be copied from conventional memory to extended memory, or from extended memory to conventional memory.

Unfortunately, the preceding BIOS extended memory access scheme has one major limitation: it does not protect against more than one application accessing the extended memory at the same time. Therefore, Microsoft, Intel, AST, and Lotus came out with an Extended Memory Specification (XMS), which is an extended memory management scheme that allows applications to access the extended memory area in a cooperative, hardware-independent manner. XMS provides functions to access the following memory areas:

- Upper Memory Blocks (UMB) – an area between 640 KB and 1 MB (1024 through 1088 KB).
- High Memory Area (HMA) – a 64 KB block of memory available at the beginning of the second 1 MB (1088 KB). The application can copy both data and code into this area. The area can be accessed using a special segment number F000H. The application must maintain HMA as a single segment.
- Extended Memory Blocks (EMB) – memory blocks above 1088 KB

The Extended Memory Manager (XMS) is an installable driver that can be installed during system loading and configuration using the DEVICE= command in the CONFIG.SYS file. For example, the Microsoft Extended Memory Manager (HIMEM.SYS) is loaded using the following command:

```
DEVICE = HIMEM.SYS
/HMAMIN=n
/NUMHANDLES=n
```

/HMAMIN=n specifies the minimum KB of the memory area a program intends to use. **n** = 0-63, default=0. This will assure the amount of memory to the application. Subsequent requests of an amount smaller than **n** by any other application will fail.

/NUMHANDLES=n sets the number of XMS handles that may be active at any one time. **n** = 0-128, default=32.

A summary of the functions provided by the Extended Memory Specification (XMS) is shown in Figure 5.

Next, we'll examine how an application program can access the extended memory area. The application should use the following steps to access the extended memory:

1. Check if the XMS driver is loaded.

This is done by loading AX with 4300H and executing INT 2FH. If the driver is available, the value 80H is returned in AL:

```
MOV     AX,4300H
INT     2FH
CMP     AL,80H
JNE     XMS_NOT_LOADED
```

2. Get the entry point of the XMS driver.

To get the entry point, load AX with 4301H and execute INT 2FH. The entry point address (Segment:Offset) will be returned in ES:BX. The program can then request the XMS functions by making a call to the driver entry point with the function code in AH:

```
MOV     AX,4301H
INT     2FH
MOV     WORD PTR XADDRS,BX
MOV     WORD PTR XADDRS+2,ES
CALL    XADDRS
```

For example, to allocate a 32 KB extended memory block requires the following instructions: Load AH with the function code and DX with the size of the extended memory block. Then call the XMS driver:

```
MOV     AH,09H
MOV     DX,32
CALL    XADDRS
```

DOS Extenders

An alternate way of solving the DOS 640 KB memory limitation is to use the DOS Extender. This product is available from several software manufacturers. The major advantage of using the extender is that unlike XMS and EMS, the DOS Extender allows applications to directly access the entire physical memory up to 16 MB without any special interface. If the application is using EMS or XMS, it accesses the memory indirectly using either EMS or XMS API calls. Under EMS, the application has to map logical pages to physical pages, and under XMS the application has to copy data from extended memory to base memory before it can be used.

The DOS Extender requires no special interface. It simply executes the application in the protect mode and allows the application to access

Function	Input	Output
Get XMS versions	AH = 00H	AX = version number
Allocate HMA area	AH = 01H DX = HMA space	AX = 0001H of HMA is assigned
Release HMA area	AH = 02H	AX = 0001H if HMA is released
Global enable A20 line	AH = 03H	AX = 0001H if A20 is enabled
Global disable A20 line	AH = 04H	AX = 0001H if A20 is disabled
Local enable A20 line	AH = 05H	AX = 0001H if successful
Local disable A20 line	AH = 06H	AX = 0001H if successful
Query state of A20 line	AH = 07H	AX = 0001H if successful
Query free Extended Memory	AH = 08H	AX = size of the largest extended memory in KB DX = total memory in KB
Allocate Extended Memory Block	AH = 09H DX = total memory in KB	AX = 0001H if successful DX = handle to allocated block
Free Extended Memory Block	AH = OAH DX = block handle	AX = 0001H if successful
Move Extended Memory Block	AH = OBH DS:SI points to move structure	AX = 0001H if successful
Lock Extended Memory Block	AH = OCH DX = block handle	AX = 0001H if successful DX:BX = address if locked block
Unlock Extended Memory Block	AH = ODH DX = block handle	AX = 0001H if successful
Get EMB Handle Information Block	AH = OEH DX = block handle	AX = 0001H if successful BH = block's lock count BL = number of EMB handles DX = block length in bytes
Reallocate Extended Memory Block	AH = OFH BX = new size BX = block handle	AX = 0001H if successful
Request Upper Memory Block	AH = 10H DX = block size requested	AX = 0001H if successful BX = segment of UMB DX = size of allocated block
Release Upper Memory Block	AH = 11H DX = segment of UMB	AX = 0001H if successful

Figure 5. XMS Functions

memory up to 16 MB. All that is needed is to link and bind the application using the special linker and binder provided by the DOS Extender. The binder takes the extender and the loader and binds it to the application, as shown in Figure 6. Even existing DOS applications can be easily accommodated under the DOS Extender with minimal modifications.

How does a DOS Extender work? Basically, the DOS Extender runs the application under protected mode, allowing the application to access the available physical memory up to a maximum 16 MB on 80286, 80386, and 80486 systems. The extender loads the application, switches the machine to protected mode, and manages the extended memory available over 1 MB.

Even though it executes the application under protected mode, when necessary, it can switch the machine to real mode in order to handle external interrupts as well as DOS and BIOS service calls from the application. For example, when an application running in protected mode issues a DOS or BIOS function call, the extender intercepts the call. It then switches the machine to real

mode and executes the function by calling the real DOS INT 21H handler. Upon completion, it switches the machine back to protected mode.

The preceding operation is transparent to the application. It even allows the application to access special memory locations, such as video buffers, by treating the real-mode segments of these locations as protected-mode selectors. For example, the real-mode video buffer address B8000H can be converted to protected-mode selector B800H for the application to access the real-mode video buffers.

Next, let's examine how a real-mode application is converted to a protected-mode application using DOS Extender DOS/16M from Rational Systems. All you have to do is to follow these steps:

1. Compile the application source file PROG.C using the Microsoft C 5.1 compiler to create the object file PROG.OBJ.

```
cl -AL  -Ox  -W3  Prog.c
```

2. Link the object file using the DOS/16M-supplied linker:

```
Link  /noe/map  \16M\preload
\16M\crt0_16M \16m\pml
prog,prog,prog;
```

- PRELOAD.OBJ provides placeholder for the application's protected-mode selectors.
- CRT0_16M is the DOS/16M replacement of Microsoft C 5.1 startup code (CRT0.OBJ).

3. Convert the program to a protected-mode executable file using the DOS/16M-supplied MAKEPM utility.

```
\16M\Makepm  prog
```

MAKEPM converts all real-mode segment IDs into protected-mode selectors.

4. Bind the DOS/16M loader program "LOADER.EXE" to the executable file (see Figure 6) using the DOS/16M-supplied bind utility "splice:"

```
\16M\Splice  prog.exe
prog.exp  \16m\loader.exe
```

LOADER.EXE is the DOS extender loader and kernel that loads the application and manages the interface between the protected mode and the real mode.

DOS Extender Restrictions

DOS extenders run only on 286-, 386-. or 486-based machines. This means that a DOS Extender will not work on systems based on 8086 and 8088 processors. Following is a list of other major restrictions on the application:

- Cannot perform real-mode address arithmetic
- Cannot access real-mode memory locations unless they are specially defined as protected-mode selectors
- Cannot misuse the segment registers while either loading data into code segment or using segment registers as temporary save areas

Conclusion

Each memory management scheme has its own advantages and disadvantages. Expanded Memory Management requires no special hardware on 386 or 486 machines, but special memory hardware is needed on 286 machines. It also requires special programming using Expanded Memory Specifications. Extended Memory Management requires no special hardware, but there must be special programming

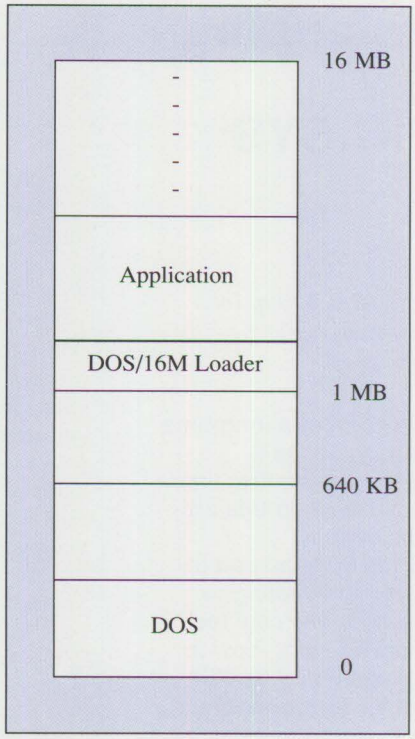


Figure 6. DOS Extender Memory Map

using Extended Memory Specifications. DOS Extenders provide protected-mode functions and allow applications to access large memory space without any special interface or hardware on 286 and 386 machines.

ABOUT THE AUTHOR

Pylee Lennil is a staff programmer at IBM Entry Systems Division in Boca Raton, Florida. He is presently involved in the development and support of PC DOS. Pylee, who joined IBM in 1983, received his B.S. in physics from Kerala University in India, and an M.S. in computer engineering from the University of Lowell in Lowell, Massachusetts.

Understanding an OS/2 CONFIG.SYS File

Paul Tremblett
Telesector Resources Group Inc., a NYNEX Company
New York, New York

A common reaction of a long-time DOS user viewing an OS/2 CONFIG.SYS file for the first time is one of intimidation. In this article, we will examine a CONFIG.SYS file in detail and attempt to change this feeling to one of confidence. The new feeling comes from knowing how to use a very powerful and flexible tool that can be used to tailor an OS/2 system to our individual needs.

The file we will be examining is a composite of my own CONFIG.SYS and several others I have seen in actual use. Before we begin, print a copy of your own CONFIG.SYS so you can check off each statement as it is presented. This way, after we have completed our examination, you will see that CONFIG.SYS is not as scary as it first appears.

The CONFIG.SYS statements are shown in Figures 1, 2, and 3, and are numbered for reference purposes only. They would not be numbered in the file.

Line 1: As the OS/2 start-up code reads the CONFIG.SYS file, it checks for errors. If any are encountered, the system pauses or continues, depending on whether YES or NO is specified for PAUSEONERROR. Because most of the time we will want to determine what the error is, YES is usually specified. If your CONFIG.SYS does not contain this statement, it is because PAUSEONERROR=YES is the default. You may be asking yourself why you would ever want to specify NO. Most likely, this question will be answered NO when your CONFIG.SYS file contains a statement (or many statements) that you do not wish to remove, but one that the start-up code determines to be in error. After the first few times you press Enter during the boot process, you will appreciate PAUSEONERROR=NO.

> Because you may be inclined to experiment as you read, before you begin, make a backup copy of your CONFIG.SYS file. Avoid such extensions as .BAK and .SAV. Instead, use the convention CFGmmdd.SYX where **mmdd** is the month and day.
>
> You should be aware that the CONFIG.SYS file is read only once during system startup. Therefore, if you make changes, the system must be restarted for these changes to take effect.
>
> Do not panic if your system cannot be booted as the result of a change you made to CONFIG.SYS. Boot from a copy of the installation diskette (note: a copy – never the original) and press ESC when the logo screen appears. Then simply transfer to C: drive, copy the backup file to CONFIG.SYS, and reboot.

Line 2: The PROTSHELL statement loads the user interface and OS/2 command processor. The statement indicates that the file containing the user interface is PMSHELL.EXE and is found on the C: drive in directory OS2. The remainder of the statement contains arguments that the user interface expects to receive. These arguments are the names of the Presentation Manager Configuration file, the Presentation Manager Program File, and the OS/2-mode command processor.

Now, you may wonder why the Presentation Manager has to be specified as the user interface, because everyone knows that it is. The purpose of the PROTSHELL statement is not to force you to specify the obvious, but to give you the opportunity to override it. When might you want to do such a thing? The best answer to this question is a real-life example. Look at the CONFIG.SYS file found on the OS/2 1.2 installation diskette. There you find the following statement:

```
protshell=sysinst1.exe
sysinst2.exe
```

It is clear that the installation process uses a different user interface, and those of us who have installed OS/2 certainly know that it does. While it wouldn't be an easy task to write your own user interface, should this be necessary, OS/2 gives you an easy way to install it. The key point here is just how flexible the system is, and how it can be customized.

Line 3: The SET statement is used to specify values for environment variables. In this case, the variable **COMSPEC** specifies the drive, directory, and file name of the command processor.

Line 4: The Installable File System (IFS) statement instructs OS/2 to load a file system other than the familiar File Attribute Table (FAT) system default. In this case, the alternate system is the High Perfor-

```
Line 1   PAUSEONERROR=YES
Line 2   PROTSHELL=C:\OS2\PMSHELL.EXE C:\OS2\OS2.INI C:\OS2\OS2SYS.INI C:\OS2\CMD.EXE
Line 3   SET COMSPEC=C:\OS2\CMD.EXE
Line 4   IFS=C:\OS2\HPFS.IFS -C:1536 /AUTOCHECK:CDEF
Line 5   RUN=C:\OS2\CACHE.EXE /LAZY:ON
Line 6   PROTECTONLY=NO
Line 7   RMSIZE=640
Line 8   SHELL=C:\OS2\COMMAND.COM /P
Line 9   DEVICE=C:\OS2\VDISK.SYS 4096 256 512
```

Figure 1. CONFIG.SYS Statements

mance File System (HPFS), which increases system performance. It does so by using cache, by keeping allocation for a file in contiguous sectors wherever possible, and by using a balanced directory tree that permits faster location of a file. Additional benefits of HPFS are long file names and extended attributes.

The -C parameter specifies that 1536 kilobytes of memory (1.5 MB) are used by the file system as cache. If no value is specified, 20 percent of available memory is used.

The /AUTOCHECK parameter specifies the disk drives (in this case C, D, E, and F) that the operating system should check during startup. If the file system for a drive is found to be in an unusable state, the system issues a CHKDSK command with the /F parameter, thus correcting any problems.

The two most common causes of an unusable file system are loss of electrical power and an improper shutdown. The latter can be eliminated by always using the SHUTDOWN option in the Desktop Manager. Even though OS/2 does intercept the Alt-Ctrl-Del key sequence and carries out appropriate housekeeping operations, you should make every attempt to form the habit of going through the normal shutdown procedure.

Line 5: Certain system programs, such as the CACHE.EXE that will be examined shortly, can be loaded and started during initialization. The RUN= statement is used to start such programs.

CACHE.EXE is a program that writes the cache used by HPFS to disk. The "LAZY" parameter (a better name would be the "deferred write" parameter) determines whether cache data is to be written immediately (/LAZY:OFF) or during idle time (/LAZY:ON). Up to three additional arguments (/MAXAGE, /DISKIDLE, and /BUFFERIDLE) may be used to control the program's actions. The defaults will generally satisfy most user requirements; they are documented in the online Command Reference Manual.

Line 6: In addition to its native (protected) mode of operation, OS/2 gives users the option of running existing DOS programs in real mode in the DOS compatibility box. Specifying PROTECTONLY=NO causes the compatibility box to be loaded during system startup. If PROTECTONLY=YES is specified, the DOS box is not loaded and the result is an OS/2-only environment. Because most users have certain favorite applications they would like to continue running under DOS, most CONFIG.SYS files contain PROTECTONLY=NO.

Line 7: The size of the DOS compatibility box is specified by this statement. If the statement is omitted, the total amount of low memory installed (either 512K or 640K) is used, less any amounts used by the OS/2 kernel and device drivers.

Line 8: Just as we used the PROTSHELL statement to load and start a user interface and command processor for OS/2 mode, we can use the SHELL statement to load and start a DOS command processor. In this instance, the default processor COMMAND.COM is started with the /P argument to indicate that the processor is to be retained in storage. As was the case for PROTSHELL, we could specify an alternate command processor.

Line 9: The DEVICE= statement is used to install a device driver. The statement shown here was taken from my own workstation. I develop a large amount of software, and because I have plenty of available RAM, I am willing to sacrifice four megabytes to be used as a virtual disk. On this disk I copy frequently accessed files, such as the

header files used for my numerous, daily C compilations. Because accessing a virtual disk involves no disk arm motion, the compilations are speeded up.

The OS/2 diskettes contain numerous device drivers that can be installed while using a DEVICE= statement. Some examples are:

- ANSI.SYS – Used in DOS mode for extended keyboard and display support.
- MOUSE.SYS – Used to implement support for pointing devices.
- NETBDD.SYS – Used when OS/2 applications programs require access to the NETBIOS Application Programming Interface (API).

Line 10: LIBPATH identifies the location(s) of dynamic link libraries. Whenever a dynamic link library module (DLL) is required, each of the directories specified in LIBPATH is searched until the required module is located. Certain software may be purchased or written that uses DLLs. The operating system cannot access the DLLs unless the LIBPATH statement is modified to include the names of the drive and directory where the software is installed.

As a technical coordinator, I often receive calls from users who have attempted to start a program and have received the message "Cannot find the file or directory. Be sure that the path(s) and file name(s) are entered correctly. (PMV1024)." The users swear that they *did* enter the name correctly, which is most often the case. Usually, the problem is that the message refers to a file which is not the name of the program, but rather the name of a DLL the program is attempting to load. Next time such a message is encountered, the source of the problem will probably be found in the LIBPATH statement in your CONFIG.SYS file. Don't forget that changes will take effect *only* after the system is rebooted.

Line 11: Here is another example of the SET statement which, as we have already seen, assigns a value to an environment variable. The **PATH** environment variable is used by the operating system to locate a .EXE or .CMD file. At this point, recent DOS converts will ask themselves "Wasn't this exact statement included in AUTOEXEC.BAT?" The answer is "yes," and some important clarification for using SET PATH is appropriate. The SET PATH directive in the CONFIG.SYS file sets the environment variable **FOR OS/2 SESSIONS ONLY**. If you are running the DOS compatibility box, create an AUTOEXEC.BAT file containing the directive that will set the environment variables for the DOS session. The first time you switch to DOS, the AUTOEXEC.BAT file is read and processed by the compatibility box.

When the environment variable **PATH** is set by CONFIG.SYS, a copy of this variable is made available to *each* window or full-screen session when the session is started.

You can issue the SET command from within a session to modify an environment variable. If you do so, realize that the change applies only to the session in which the SET command was issued.

It is quite possible (and in some cases, desirable, albeit confusing) to have a different environment in every active OS/2 session.

Line 12: The contents of the environment variable **DPATH** can be used by application programs to locate **DATA** files outside the current directory. DPATH is similar to APPEND. Searching a path specified by APPEND is automatic. With DPATH, searching is the responsibility of the application program.

Line 13: This statement is used to control what the user sees as a prompt. The values specified in this case are the default values supplied by the OS/2 installation process. The **$i** indicates that the "help" line is displayed. This is the line at the top of each session that reminds you that Ctrl+Esc = Task List, and that typing HELP will indeed provide help. To remove this line, simply remove the **$i**. The characters D, T, and V can be used to display date, time, or version, respectively.

Line 14: Long-time DOS users know that the last DOS command issued could be recalled in its entirety by pressing the PF3 key, or one character at a time by pressing PF1. Many users find it more productive to use one of the numerous available command-line editors to recall, optionally edit, and reissue commands. OS/2 gives us the choice of either of these methods of command-line retrieval.

Specifying KEYS=ON results in each command entered at the prompt being stored in a queue. Any commands in the queue may be recalled, edited, and reissued using special keys that are documented in the online command reference manual.

Specifying KEYS=OFF results in the well-known DOS method of command-line retrieval (that is, PF3 or PF1).

```
Line 10  LIBPATH=C:\OS212\DLL;C:\MUGLIB\DLL;F:\CMLIB\DLL.....
Line 11  SET PATH=C:\OS2;C:\MUGLIB;F:\CMLIB;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\.....
Line 12  SET DPATH=C:\OS2;C:\MUGLIB;F:\CMLIB;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\.....
Line 13  SET PROMPT=$i[$p]
Line 14  SET KEYS=ON
Line 15  IOPL=YES
Line 16  THREADS=255
Line 17  MAXWAIT=3
Line 18  PRIORITY=DYNAMIC
Line 19  TIMESLICE=45,125
Line 20  MEMMAN=SWAP,MOVE,NOSWAPDOS
```

Figure 2. CONFIG.SYS Statements

Line 15: In a multitasking environment, if every program were allowed to perform physical I/O, the result would be chaotic. Application programs normally run at privilege level 3 (the lowest) and are not permitted to perform physical I/O. Instead, they communicate with physical devices through the services provided by the operating system. The nature of some software is such that direct access to the registers and instructions required to communicate with a physical device is essential to proper operation. The input/output privilege level (IOPL) statement is used to permit this. A value of YES, shown here, permits any program that requests I/O privilege to be granted it. A value of NO denies such privilege. If IOPL is followed by a list of module names, only those modules will be granted I/O privilege.

Line 16: The basic unit of execution in OS/2 is the thread. All programs have at least one thread and some have many. If you type PSTAT (Process Status) in either a full-screen or windowed session, you will see a listing of all the processes currently running with the process ID (PID) listed in column 1, the process name in column 4, and the threads belonging to each of the processes in column 5. The value specified in the THREADS= statement is used to control the maximum number of threads that the system will start. A value between 16 and 512 may be specified. The default number supplied by the installation procedure will suffice in most cases; however, the number may be adjusted upwards or downwards to suit individual needs.

Line 17: This statement, which works in close conjunction with the PRIORITY and TIMESLICE statements, is designed to prevent a process from experiencing processor starvation. This occurs when other processes of higher priority issue excessive demands for processor services. Value 3 indicates that if a regular-class thread waits longer than three seconds to access the processor, it will be granted a temporary boost in priority.

Line 18: As mentioned earlier, the thread is the basic unit of execution. OS/2 schedules threads for execution based on priority classes and levels. PRIORITY=DYNAMIC can be used to instruct OS/2 to vary the priorities of threads at runtime. Specifying PRIORITY= ABSOLUTE instructs OS/2 not to vary priorities. Before altering this statement, obtain a good working knowledge of the concepts of priorities.

Line 19: Using this statement, the minimum and maximum amounts of processor time allotted to processes and programs for both OS/2 and DOS mode can be varied. The first value specifies the minimum time in milliseconds, and must be an integer with a value greater than or equal to 32. The second value specifies the maximum time in milliseconds, and must be greater than or equal to the minimum time and less than 65536. If the maximum time is omitted, it is assigned a value equal to the minimum time. This statement and the two previous ones are used to tune the system. Before experimenting with these statements, prepare yourself for a long, tedious process with numerous shutdowns and restarts. The default values supplied by the developers were chosen wisely based on experience and a knowledge of OS/2 that most of us will never gain.

Line 20: In some cases, system responsiveness must be guaranteed. One example is where time-critical tasks such as process control are running. The MEMMAN= statement can be used to disable storage

```
Line 21  SWAPPATH=C:\OS2\SYSTEM 512
Line 22  COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
Line 23  CODEPAGE=437,850
Line 24  DEVINFO=KBD,US,C:\OS2\KEYBOARD.DCP
```

Figure 3. CONFIG.SYS Statements

compaction (NOMOVE) or swapping (NOSWAP), or both, as a means of ensuring system responsiveness. The decision to disable either of these features should not be made lightly, and you should be aware of the penalties associated with this decision. Specifying NOSWAP obviously affects the number of processes started, and specifying NOMOVE can result in programs experiencing failures in dynamic requests for storage as storage becomes fragmented. The NOSWAPDOS parameter, although not documented at this time in either the printed or the online version of the manuals, prevents system memory allocated for DOS mode from being swapped out for use by OS/2 programs. Specifying this parameter, which is the system default, will not only contribute to system responsiveness, but will add to the effect of the previously mentioned NOSWAP parameter.

Line 21: By taking full advantage of the architecture of the 80286 (OS/2 1.2 uses the 80386 as an 80286), OS/2 presents each application with a virtual address space. Applications do not directly manipulate physical memory; instead, they work with virtual segments that are mapped to physical memory. The same physical memory locations can be shared by multiple programs, and OS/2 moves segments from physical memory to a disk file, and vice versa. The file in question is SWAPPER.DAT, and the SWAPPATH statement tells OS/2 the name of the drive and directory where the file will reside. In the previous example, the swap file is located on the C: drive in the SYSTEM subdirectory of the OS2 main directory. It is normal for SWAPPER.DAT to increase in size. To prevent it from increasing to the point where all available disk space is consumed, an additional parameter can be specified as part of the SWAPPATH directive. This parameter is the number of kilobytes of disk space (512, in our case) that the operating system must leave available.

Line 22: This statement is one of a series of interrelated CONFIG.SYS statements. It is used by the base operating system to determine how such items as date, time, decimal separator, character case map table, collating sequence table used by the SORT filter, and the environment vector used by the double-byte character set (DBCS) are to be handled. The country code for the United States is 001. Other values can be found in the online command reference manual.

Line 23: A code page is a defined character set. OS/2 supports a primary and a secondary code page. In this statement, the U.S. code page (437) is the primary and the Multilingual code page (850) is the secondary. The CODEPAGE statement works in conjunction with the COUNTRY statement.

Line 24: To properly translate keystrokes into the characters of each of the two supported code pages, users supply the system with a keyboard layout and the file name containing the keyboard translation tables. In this case, the keyboard layout specified is US and the tables are located in C:\OS2\KEYBOARD.DCP. This statement is closely related to both the COUNTRY and CODEPAGE statements.

In addition to the DEVINFO statement for the keyboard, there are similar statements for the display (SCR) and the printer (PRN). Full details are in the online command reference manual.

Summary

This brings us to the end of our examination of CONFIG.SYS. If you have been checking off the statements in your own CONFIG.SYS, the ones not checked are likely either SET or DEVICE statements similar to those we looked at. While our analysis was by no means exhaustive, it should have clearly demonstrated that CONFIG.SYS is not as scary as it first looked.

ABOUT THE AUTHOR

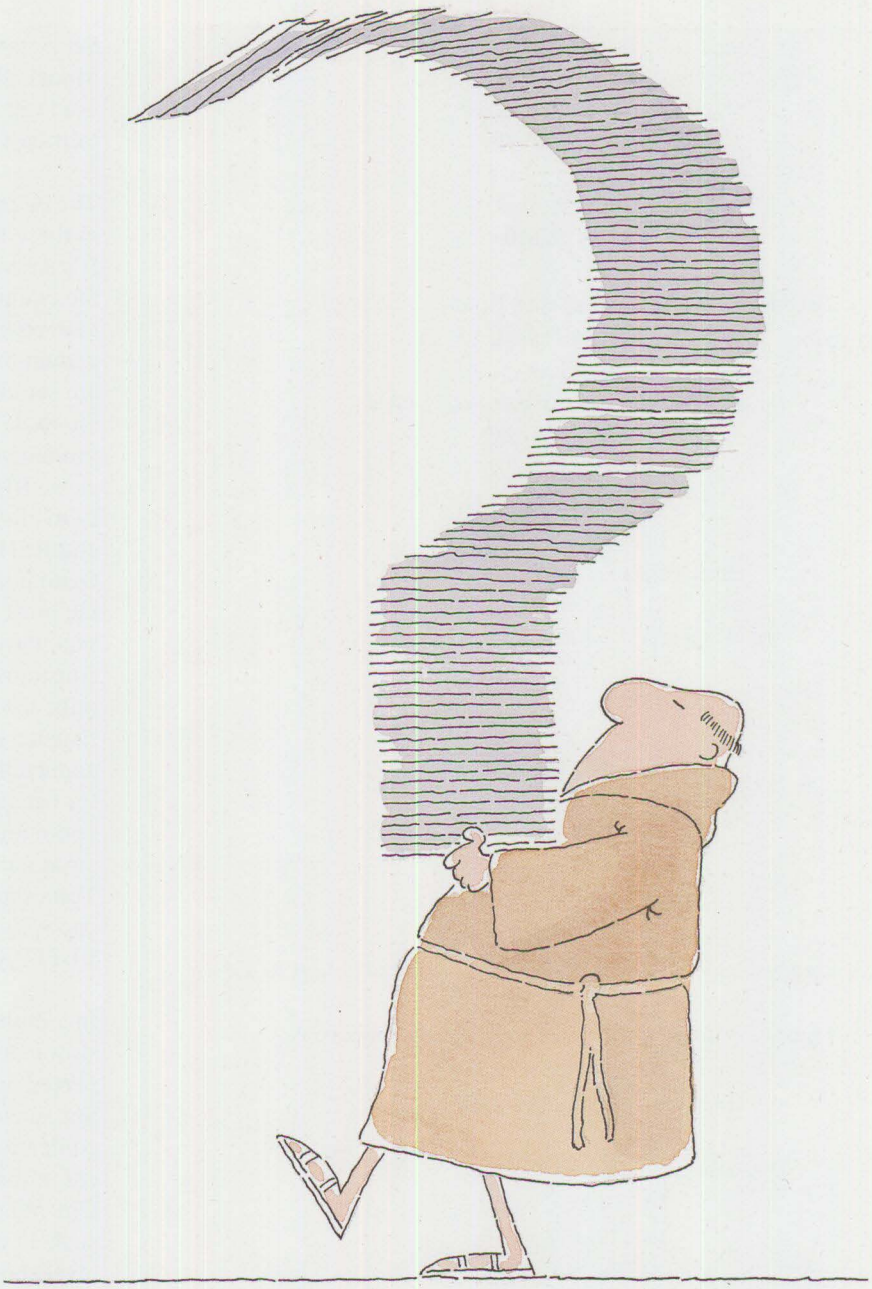
Paul Tremblett is a lead I.S. specialist for Telesector Resources Group Inc., a NYNEX Company. During his 20-year career, he has worked with minis, mainframes, and micros in numerous areas, including development of operating systems for realtime data acquisition. Paul has worked with OS/2 since the prerelease level of version 1.0, and has given technical presentations on program development under Presentation Manager at SHARE and GUIDE. His current responsibilities include helping software developers migrate from DOS to OS/2.

OS/2 EE 1.2 Database Manager Performance

Cesar R. Velasco and
N. James Smith
IBM Corporation
Austin, Texas

This article provides performance measurement results for frequently used OS/2 Extended Edition (EE) 1.2 Database Manager (DBM) utility and environment commands, as well as a range of single (atomic) data manipulation statements.

Editor's note: The performance data presented here was obtained in a controlled environment and may vary significantly from results obtained in other environments, even when the same test cases are used. Therefore, caution should be exercised when using the data.



Performance is always a difficult subject to discuss because there are several factors that affect it. The following discussion explains the results of a variety of test cases run in a controlled environment with the intention of gathering information on OS/2 EE DBM performance. Whenever possible, guidance is given about what can be done to improve performance.

Test Methodology

The test cases used consisted of simple and complex queries, data definitions, environment commands, and selected utilities. The tables consisted of six columns, and ranged in size from 100 to 100,000 rows. The results returned by a query varied from 1 to 10,000 rows. Graphs were included when possible to help illustrate comparisons between test cases or specific points discussed. The queries were executed as static SQL statements imbedded in C language application programs against a database consisting of seven tables.

Each test case was executed five times; response times for the first pass and the average of the last four passes are given. Descriptions of the database tables and SQL statements executed for each test are included in Figures 26 and 27.

For local execution, test cases were run on a PS/2 Model 80-A31 system; and for remote execution, a PS/2 Model 80-A31 Remote Data

Local Execution

PS/2 M80 - A31
RAM - 16 MB

Database Manager Configuration Parameter Settings:
All Parameters = Default Values

Database Configuration Parameter Settings:
BUFFPAGE = 250
LOGFILSIZ = 250
LOGPRIMARY = 5
LOGSECOND = 1
Other Parameters = Default Values

BIND Option: I = RR (Repeatable Read)

Remote Execution

RDS Server: PS/2 M80 - A31
RAM - 16 MB
RDS Requester: PS/2 M80 - 111
RAM - 16 MB
IBM Token-Ring Network 16/4 Adapter /A used in RDS Server and Requester

Database Manager Configuration Parameter Settings:
RSHEAPSZ = 100
Other Parameters = Default Values

Database Configuration Parameter Settings:
BUFFPAGE = 250
LOGFILSIZ = 250
LOGPRIMARY = 5
LOGSECOND = 1
Other Parameters = Default Values

BIND Option: I = RR (Repeatable Read)
Record Blocking = ALL
Transmission Service Mode = SQL LAN-Only Option (SQLLOO)

Figure 1. Equipment Configuration

Services (RDS) server and a PS/2 Model 80-111 RDS Requester were used (all running OS/2 Extended Edition 1.2).

The equipment configurations used in these tests are shown in Figure 1. It's a general practice to configure the equipment for testing with more than enough memory, and to make certain the configuration parameters are set high enough to ensure that the SQL function is tested in an environment with no constraints. Because BUFFPAGE, LOGFILESIZ, LOGPRIMARY, LOGSECOND, and RSHEAPSZ are among the configuration parameters that can make the most difference to an application, these were set to avoid any constraints. The best way to determine the proper configuration parameter setting for a particular application is to run a series of tests (or benchmarks) to determine the optimum setting, testing one parameter at a time at different levels. Tests show that 1 MB (250, 4 KB pages) is a good starting point for BUFFPAGE and LOGFILSIZ.

In a multiple requester system, slower response times should be expected due to factors such as locking, queuing delays, and resource contention, which may not be present in this single-requester test system. When writing applications for such an environment, the isolation level should be considered. Because Cursor Stability and Uncommitted Read relax the duration of the locks held, using one of these isolation levels may improve performance. In all cases, Record Blocking improves performance.

Another facility provided by Database Manager that may improve performance in this environment is the Database Application Remote Interface. This allows an ap-

ID Description	Local		Remote	
	First Pass	Avg Last 4 Passes	First Pass	Avg Last 4 Passes
01 START USING DATABASE	1.88	1.89	2.81	3.31
02 CREATE TABLE	1.06	1.08	1.06	1.04
06 DROP TABLE	0.91	0.92	0.94	1.04
07 STOP USING DATABASE	1.09	1.09	0.65	0.69

Figure 2. DBM Commands and Utilities

plication program that uses a remote database to execute a routine stored at the location of the database (remote server), which reduces the amount of network traffic between the RDS Requester and the RDS Server. The performance improvement achieved will be application-dependent.

Database Manager Performance Results

All the response times are given in seconds. The ID number corresponds to the query or function tested, as described in Figure 26. In some cases, the time for the First Pass Locally is longer than the time for the First Pass Remotely, which seems inconsistent with other timings. This difference is usually very slight and can probably be attributed to the fact that there is always an inherent error (however slight) in any timing. The position of the disk head is also different for each timing; this may also cause the difference. These timings were dependent on the resolution of the PS/2 Model 80 clocks. That is why averages were also included.

Database Manager Utilities

The timings shown in Figure 2 are for frequently used functions. Every application began by attaching to the database using the **START USING DATABASE** function and ended by using the **STOP USING DATABASE** function.

Figures 3 and 4 show the direct correlation between the number of rows being imported and the time needed. Import time depends on a number of factors: size and number of rows, size and number of columns, and whether indices are being built during the import.

Because it is important to execute REORG to eliminate the fragmentation that results from deletions, updates, and insertions of rows, Figure 5 shows a comparison of the time it takes to REORG a table with 10,000 rows versus the time it takes to REORG a table with 100,000 rows. Once REORG has been run, RUNSTATS execution should follow to keep the statistics on the

ID Description	Local		Remote	
	First Pass	Avg Last 4 Passes	First Pass	Avg Last 4 Passes
08 IMPORT DEL (Delimited ASCII)				
No. of rows = 1,000	15.4	15.4	20.3	20.3
No. of rows = 2,500	30.7	30.7	39.0	39.0
No. of rows = 5,000	57.9	57.9	71.0	71.0
No. of rows = 7,500	88.0	88.0	106.0	106.0
No. of rows = 10,000	130.0	130.0	150.0	150.0
No. of rows = 30,000	420.0	420.0	482.0	482.0
No. of rows = 50,000	741.0	741.0	843.0	843.0
No. of rows = 70,000	1092.0	1092.0	1235.0	1235.0
No. of rows = 100,000	1677.0	1677.0	1883.0	1883.0

Figure 3. IMPORT

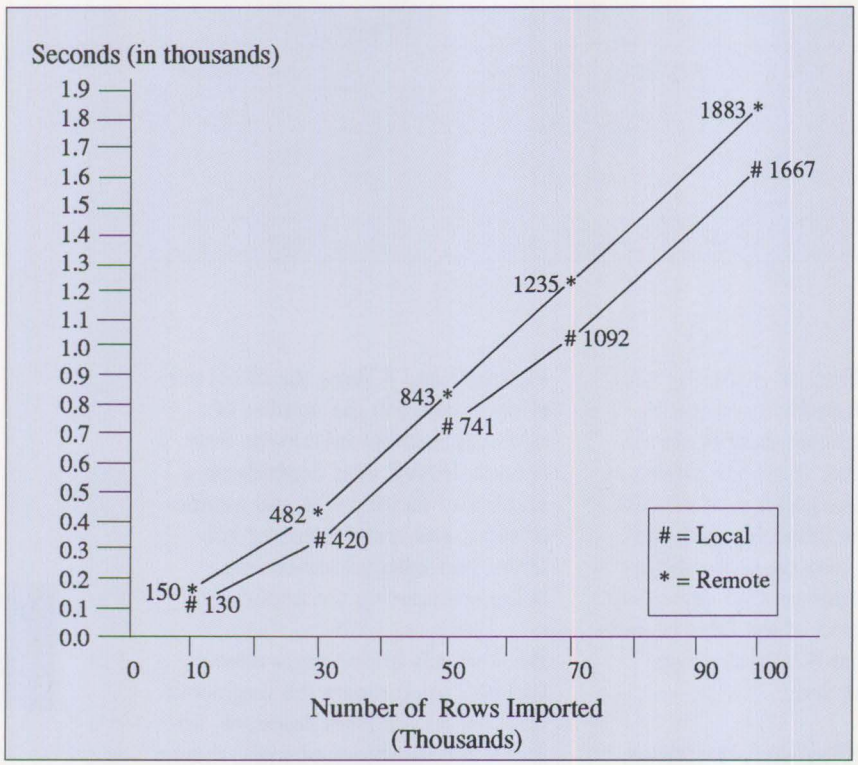


Figure 4. IMPORT

REORG	
No. of rows = 10,000	204 (3.4 minutes)
No. of rows = 100,000	2197 (36.6 minutes)
RUNSTATS	
No. of rows = 10,000	133 (2.2 minutes)
No. of rows = 100,000	1397 (23.3 minutes)

Figure 5. DBM Utilities

tables and indices current. These statistics are used by the Database Manager Optimizer to determine the most efficient way of accessing the data. Both functions may be time-consuming. How much time is actually required depends on the equipment configuration; number, size, and type of indices; and the number and size of the rows in the table.

REORG needs enough free space on the disk to make a temporary copy of the table plus enough log file space for logging all the transactions. The space needed for the temporary table should be less than the space occupied by the original table. Once the temporary table is created, the old table and old indices are deleted. RUNSTATS should be run because new indices are created. In case there is not enough free space on the disk where the table resides, in OS/2 EE 1.2, the log file can reside on another hard disk. If that still isn't sufficient, the temporary file created by REORG can also be redirected to another hard disk (not via Query Manager, but via a REXX or C language, for example).

If only the number of rows changes, the time will grow in proportion to the size of the table. In other words, in our tests (Model 80-A31, local mode), as the table grew by 10 times, the time to complete these functions also grew by about 10 times. Both tables contained four indices: all numeric – one each on columns 1, 2, and 3, and one multicolumn index on columns 1, 2, and 3.

Simple Queries

The SELECT A RECORD cases (Figure 6) select one row from near the middle of the table. In these

ID Description	Local		Remote	
Table A - 10,000 Rows (No Index)	**First Pass**	**Avg Last 4 Passes**	**First Pass**	**Avg Last 4 Passes**
10 SELECT A RECORD	5.53	5.34	5.47	5.46
10 SELECT SAME RECORD AGAIN	2.22	2.16	2.25	2.25
15 COUNT RECORDS (10 rows counted)	2.25	2.21	2.31	2.28
16 UPDATE RECORDS (20 rows updated)	2.84	2.80	2.85	2.85
Table A - 10,000 Rows (Index)				
26 SELECT A RECORD	0.13	0.13	0.19	0.23
26 SELECT SAME RECORD AGAIN	0.03	0.03	0.09	0.12
31 COUNT RECORDS (10 rows counted)	0.03	0.03	0.06	0.07
32 UPDATE RECORDS (20 rows updated)	0.53	0.58	0.63	0.61
Table J - 100,000 Rows (No Index)				
11 SELECT A RECORD	53.40	53.50	53.60	53.70
11 SELECT A RECORD AGAIN	52.30	53.10	52.60	52.80
15 COUNT RECORDS (10 rows counted)	53.40	53.40	55.20	52.90
16 UPDATE RECORDS (20 rows updated)	60.00	60.00	59.00	59.20
Table J - 100,000 Rows (Index)				
26a SELECT A RECORD	0.47	0.52	0.59	0.62
26a SELECT SAME RECORD AGAIN	0.03	0.03	0.12	0.12
31 COUNT RECORDS (10 rows counted)	0.12	0.12	0.16	0.16
32 UPDATE RECORDS (20 rows updated)	0.84	0.81	0.84	0.88

Figure 6. Simple Queries

cases, however, the queries could be satisfied by another row with the same key if it existed. In the NO INDEX case, Database Manager must scan the entire table to find all rows meeting the criteria, while the INDEX cases allow significant savings in processing and elapsed time. If an appropriate index exists and is used to process the query, rows that satisfy the query can be located much faster via an index scan. Because the index entries are arranged in ascending or descending order of the search-key values, the entire index file may not have to be scanned. The Database Manager Optimizer makes the choice of whether to use the available indices or not.

By using a tool called EXPLAIN, an IBM representative can tell whether or not Database Manager is using the indices.

Something else to note is the time difference between SELECT A RECORD/ SELECT SAME RECORD for TABLEA versus TABLEJ without an index:

	TABLEA	TABLEJ
SELECT	5.5	53.4
SELECT SAME	2.2	52.3

It takes only half as much time to SELECT the SAME record in TABLEA with 10,000 rows, because that record is already in the buffer pool. However, in TABLEJ with 100,000 rows (which don't fit in the buffer pool), the SELECT SAME operation begins reading from disk again. For SELECT SAME RECORD with index, the index and data record fit into the buffer pool for both tables. There are two things here to consider for improving performance: size of table versus size of BUFFPAGE and the advantages of using an index on large tables.

A small increase in elapsed time is seen as indexed tables increase in size due to the larger index files that must be searched.

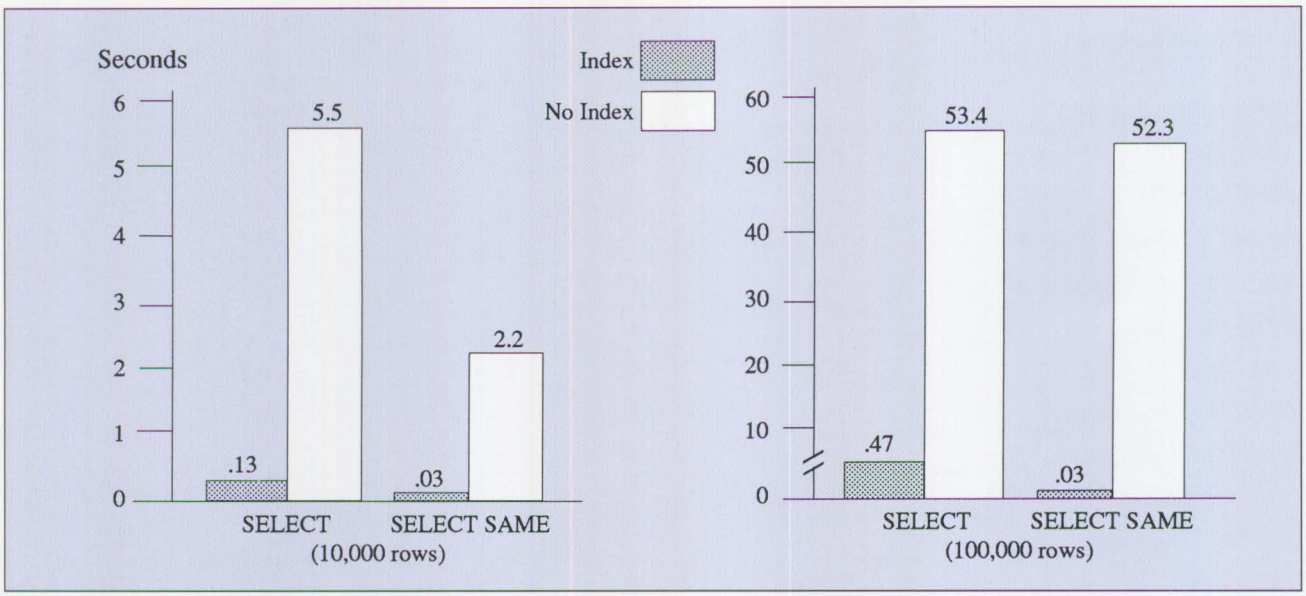


Figure 7. SELECT RECORD (Local)

Figures 7 and 8 illustrate the dramatic savings that can be obtained by using indices for selects in both local and remote environments.

The graph in Figure 9 shows a significant difference in processing time for a local update of a table with, versus without, an index.

Indices must be updated after records have been updated. For these tests, the time saved for updating the row more than compensates for the extra time needed to update the indices. Remote updates show a similar pattern.

Indexing is also clearly beneficial in join processing (Figures 10 and 11). As a general rule, indexing on a table is very beneficial as long as the appropriate indices are chosen and are used by the optimizer. You can, however, overuse indexing. The performance gains for queries and joins need to be weighed

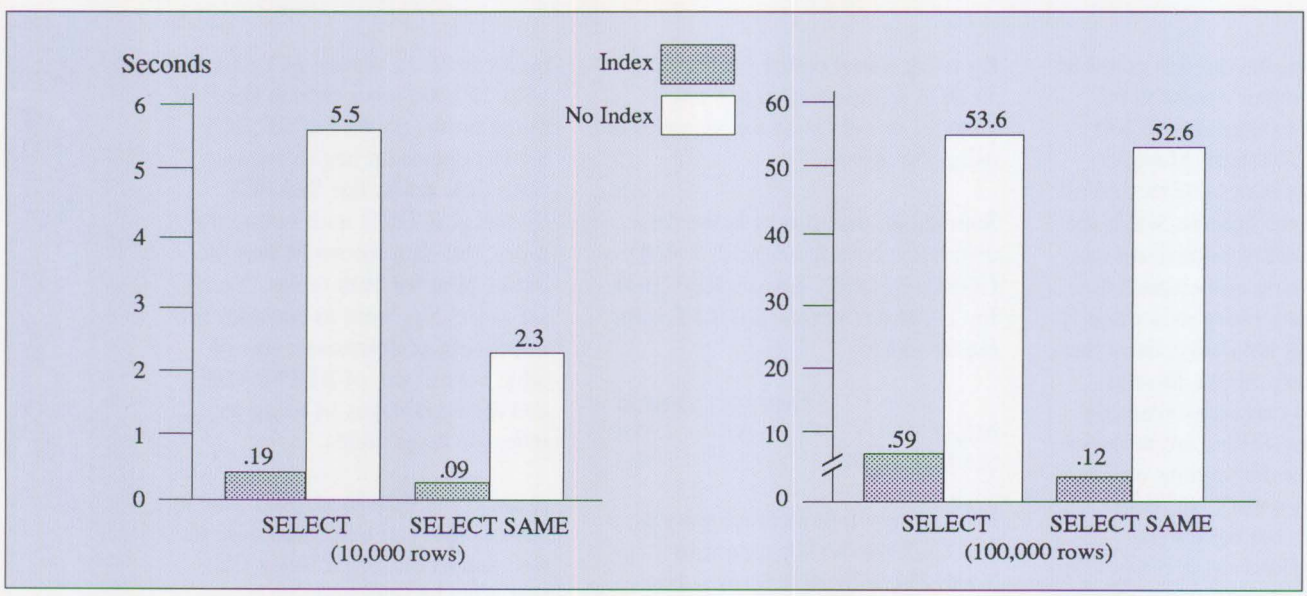


Figure 8. SELECT RECORD (Remote)

against the space and performance costs of maintaining the indices during updates, deletes, and inserts. As always, these decisions are based on the equipment configuration and the needs of the application program.

Creating Indices

As shown in Figure 12, the time needed to create indices depends on a number of factors:

1. The time required to create indices on multiple columns may depend on the order of the fields chosen in the create index statement (creating INDXA123 could be different from creating INDXA321). In our tests (INDXA123/INDXJ123), Field 1, which is a unique integer in ascending order, was chosen first. The times for this case are slightly higher than the times to create a single index (INDXA1/INDXJ1) on the same field, because there is a certain amount of checking that must be done when creating indices on multiple columns.

2. Creating an index on Field 2 (INDXA2/INDXJ2), which has the same data as Field 1, shows the effect of having the unique integers in random order. This took about twice as long as the time needed when the integers were in consecutive order.

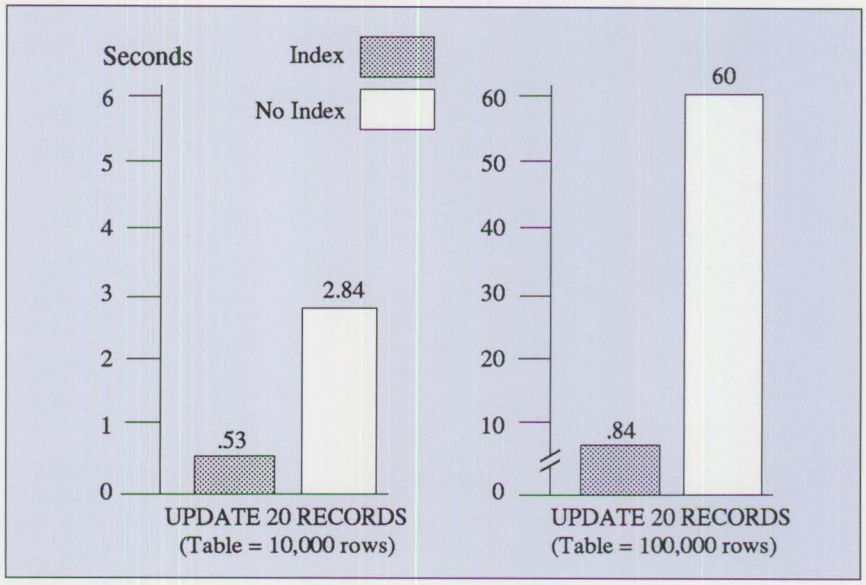


Figure 9. UPDATE RECORDS (Local)

3. The last test, creating an index on Field 3 (INDXA3/INDXJ3), which holds randomly dispersed integers that appear ten times each, shows that the time is almost doubled again when dealing with multiple duplicate values.

4. Creating the index on Field 1 is approximately linear with the number of rows in the table.

Figure 12 along with Figures 13 and 14 show how changing the value characteristics of the index field as previously described changed the create index time.

Figure 14 also shows how the creation time depends on the number of rows of data as well.

There has been much speculation about whether importing data into a table with an index processes more efficiently than importing the data first, and then creating the index. Our tests show that the time to create an index on an established table is less than the time to dynamically build that index during the table import (Figures 15 and 16). For larger tables, the savings gained by creating the index after import is greater (Figures 17 and 18).

ID Description		Local		Remote	
JOIN TABLE		First Pass	Avg Last 4 Passes	First Pass	Avg Last 4 Passes
13 No Index, Table B, C	100 rows	2.97	2.94	4.06	4.06
29 Indexed, Table B, C	100 rows	1.28	1.30	2.59	2.58
13 No Index, Table A, J	10,000 rows	547.00	554.00	654.00	665.00
29 Indexed, Table A, J	10,000 rows	165.00	165.00	271.00	268.00

Figure 10. JOIN TABLE

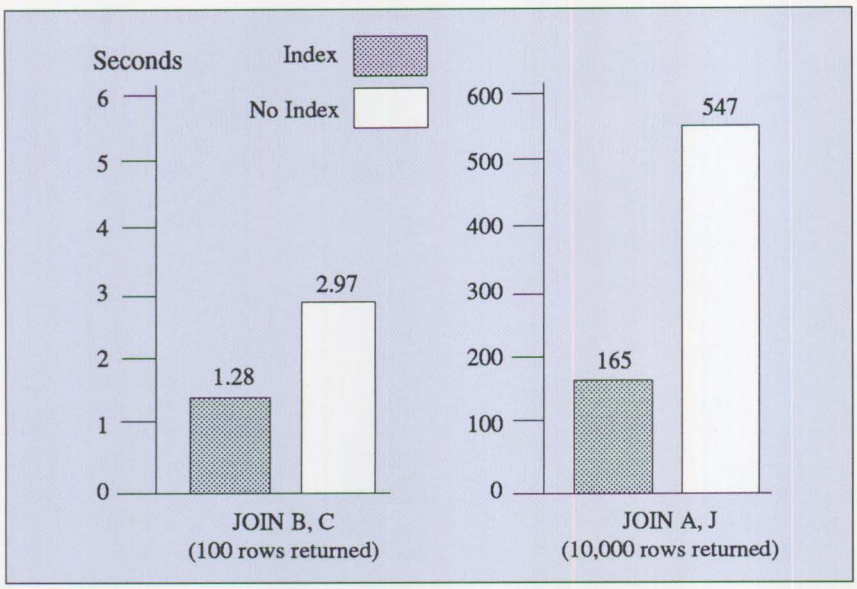


Figure 11. JOIN TABLE (Local)

Figure 19 shows information about INSERT, DELETE, and variations on the SELECT statements for TABLEA, TABLEC, and TABLEJ.

Complex Queries

As the queries get more complex, there are ways to fine-tune the SQL statements. Doing this minimizes the use of system resources and the time needed to access data in a very large table. Following these tips whenever possible will help improve performance:

- Specify only the columns that are needed in the SELECT list
- Avoid numeric conversions
- Index columns in DISTINCT, ORDER BY, and GROUP BY statements
- For joins, try to code a predicate that refers to indexed columns
- Use joins instead of correlated subqueries
- Use joins instead of non-correlated subqueries preceded by the IN predicate
- Use join predicates when joining tables

Figure 20 shows the results obtained from using more complex SQL statements on the various tables used in the tests.

Customizing DBM Parameters

During testing, it was shown that sort (ORDER BY) performance is roughly proportional to the number of rows sorted within the range tested (Figures 21 and 22). When the available real storage of the system is insufficient to contain the sort intermediate results, performance may be degraded. Consider

ID Description	Local		Remote	
CREATE INDEX	First Pass	Avg Last 4 Passes	First Pass	Avg Last 4 Passes
20 CREATE INDXA123 - Table A	23.0	22.9	23.0	22.8
20 CREATE INDXA1 - Table A	19.8	19.8	19.6	19.6
20 CREATE INDXA2 - Table A	37.2	37.2	36.7	36.7
20 CREATE INDXA3 - Table A	71.0	71.0	71.0	71.0
20 CREATE INDXJ123 - Table J	281.0	285.0	282.0	285.0
20 CREATE INDXJ1 - Table J	210.0	210.0	207.0	207.0
20 CREATE INDXJ2 - Table J	419.0	419.0	414.0	414.0
20 CREATE INDXJ3 - Table J	796.0	796.0	796.0	796.0
21 CREATE INDXB1 - Table B	1.53	1.56	1.57	1.54

Note: The following format was used: INDXxyyy where the table used is 'x,' and the columns used for creating the index is 'yyy.' Hence, INDXA123 is an index defined on columns 1, 2, and 3 of table A.

Figure 12. CREATE INDEX

increasing the SORTHEAP Configuration Parameter if the application seems to be slowing because of sorting. Using appropriate indices minimizes the use of SORTHEAP. Sorting time also depends on the number and size of rows sorted, and the number, type, and length of sequence fields.

In an earlier study, a Model 80-111 was used to determine the effects of changing the BUFFPAGE size. The results obtained are shown in Figures 23, 24, and 25.

Database records are read and updated in the buffer pool area of memory. Because data can be accessed much faster in the buffer pool than on a fixed disk, an increase in buffer pool size usually improves performance. While in some cases our tests show that nothing was gained by increasing the BUFFPAGE size, they do show that performance did not suffer with such an increase. On the other hand, especially with the Long Response Time Group, there were definite benefits to performance by increasing the BUFFPAGE size. As this parameter increases, the SQLENSEG (maximum number of shared segments) parameter may also require adjustment.

With larger tables and result sets, or when multiple Database Manager requests are being processed concurrently, providing BUFFPAGE size beyond 1 MB (BUFFPAGE=250) has been proven helpful. Testing the application with different BUFFPAGE sizes helps determine the optimum setting.

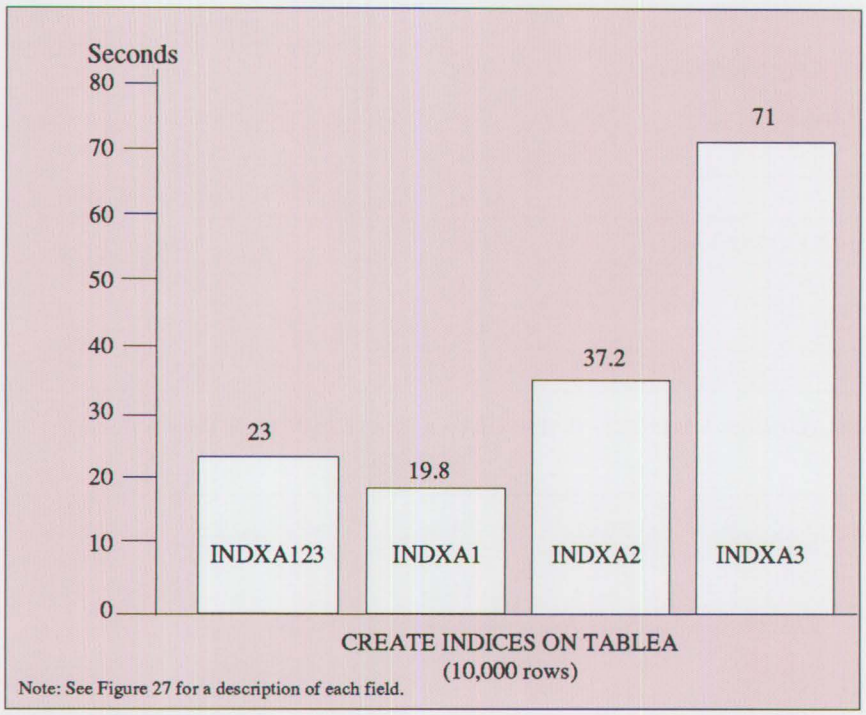


Figure 13. CREATE INDEX (Local)

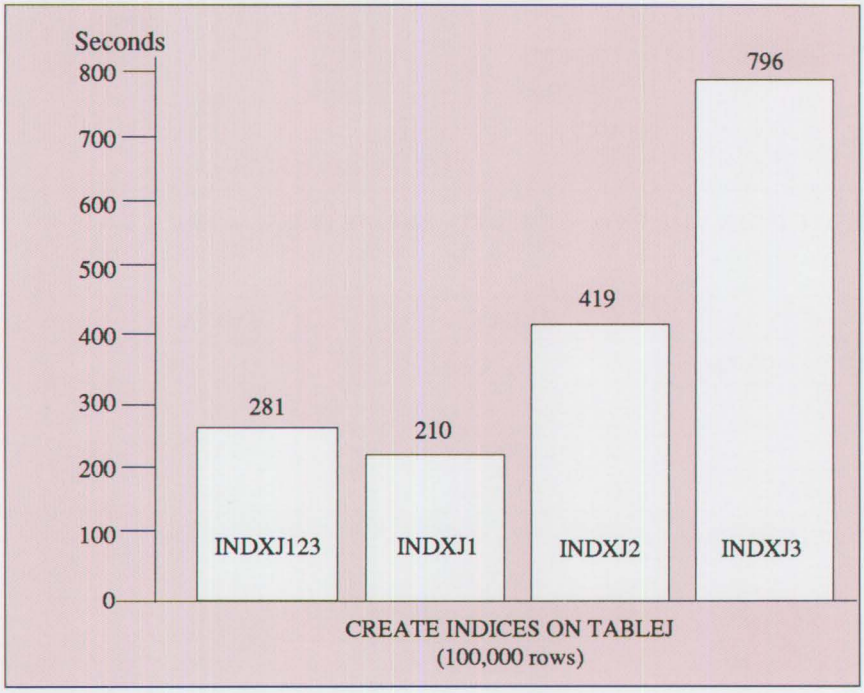


Figure 14. CREATE INDEX (Local)

	INDXA1	INDXA2	INDXA3
CREATE TABLEA (10,000 rows)	1.06	1.06	1.06
CREATE INDEX	0.94	1.00	0.97
IMPORT	159.00	181.00	210.00
Totals	161.00	183.06	212.03
CREATE TABLEA (10,000 rows)	1.06	1.06	1.06
IMPORT	130.00	130.00	130.00
CREATE INDEX	19.80	37.20	71.00
Totals	150.86	168.26	202.06

Figure 15. (CREATE INDEX + IMPORT) vs. (IMPORT + CREATE INDEX) on Table A (Local)

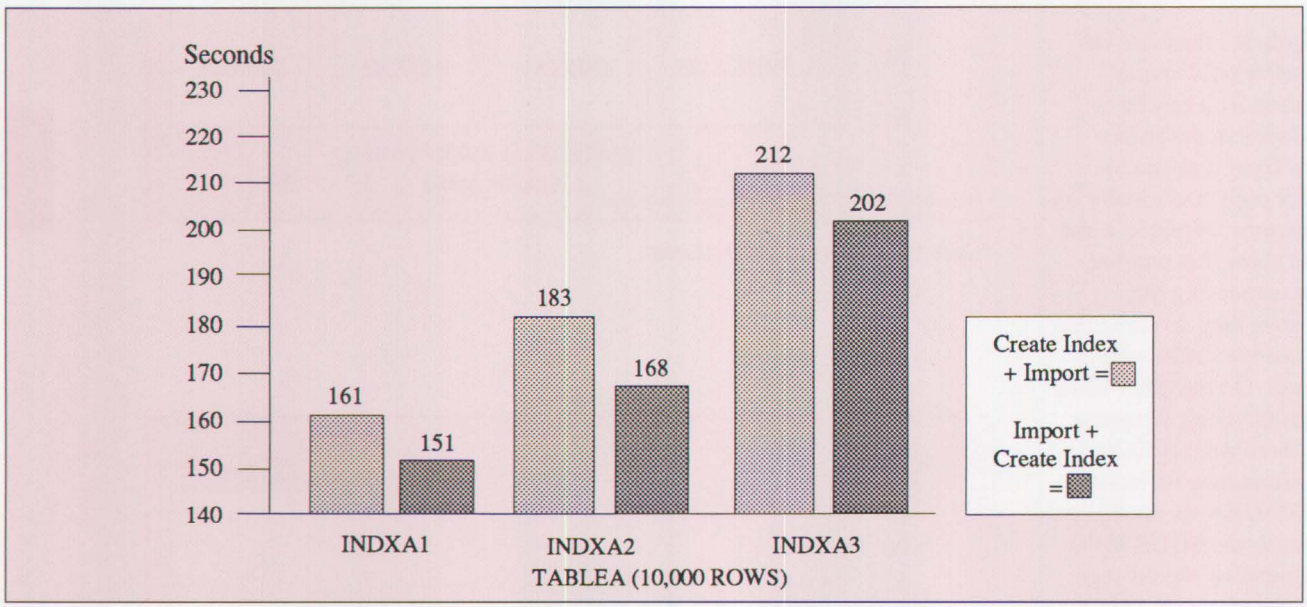


Figure 16. (CREATE INDEX + IMPORT) vs. (IMPORT + CREATE INDEX) on Table A (Local)

	INDXA1	INDXA2	INDXA3
CREATE TABLEJ (100,000 rows)	1.06	1.06	1.06
CREATE INDEX	0.97	1.00	1.03
IMPORT	2113.00	2333.00	2724.00
Totals	2115.03	2335.06	2726.09
CREATE TABLEJ (100,000 rows)	1.06	1.06	1.06
IMPORT	1677.00	1677.00	1677.00
CREATE INDEX	210.00	419.00	796.00
Totals	1888.06	2097.06	2474.06

Figure 17. (CREATE INDEX + IMPORT) vs. (IMPORT + CREATE INDEX) on Table J (Local)

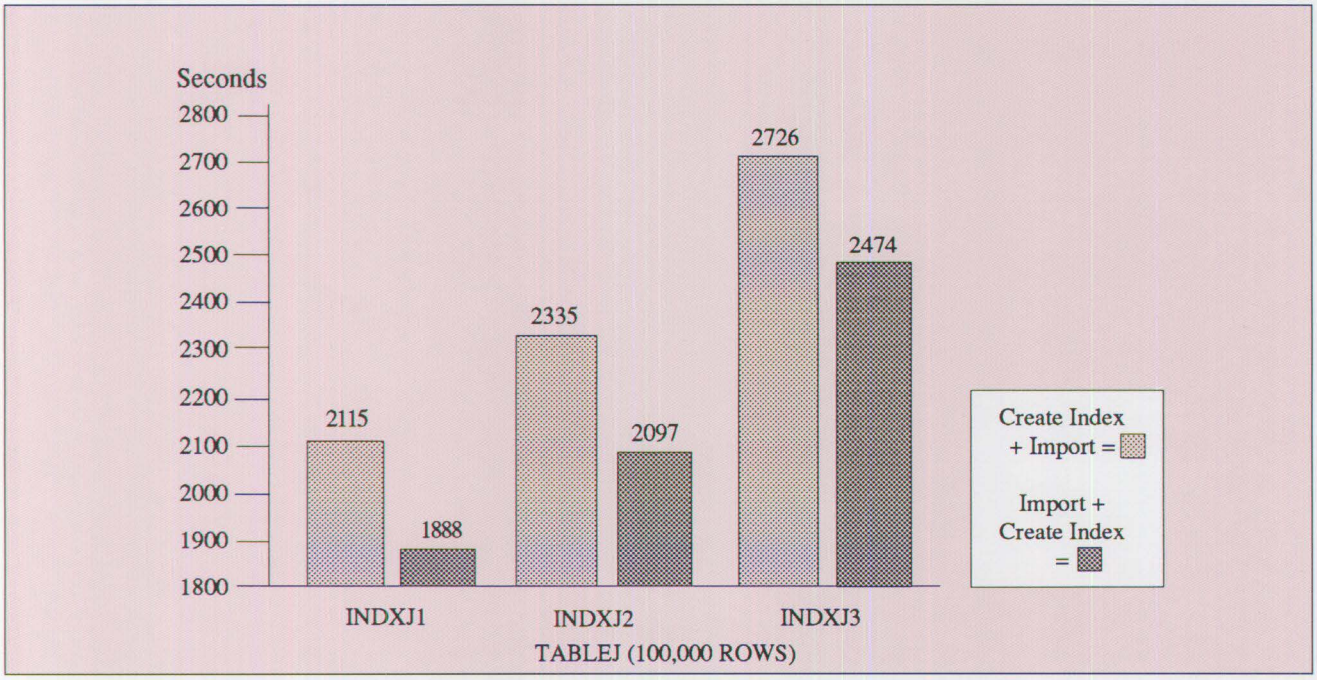


Figure 18. (CREATE INDEX + IMPORT) vs. (IMPORT + CREATE INDEX) on Table J (Local)

ID Description		Local		Remote	
TABLEA - Indexed		**First Pass**	**Avg Last 4 Passes**	**First Pass**	**Avg Last 4 Passes**
27 SELECT/ORDER BY -	20 Rows Returned	0.66	0.61	0.91	0.91
28 COMPLEX SELECT -	7 Rows Returned	0.25	0.31	0.44	0.44
33 DELETE RECORDS -	10 Rows Deleted	0.78	0.79	0.84	0.86
34 INSERT RECORDS -	10 Rows Inserted	0.16	0.15	0.25	0.24
Tables Used = A, C					
TABLEJ - Indexed					
27 SELECT/ORDER BY -	20 Rows Returned	1.13	1.17	1.32	1.40
28 COMPLEX SELECT -	7 Rows Returned	0.47	0.48	0.56	0.58
33 DELETE RECORDS -	10 Rows Deleted	1.25	1.28	1.38	1.39
34 INSERT RECORDS	10 Rows Inserted	3.56	3.61	3.62	3.64
Tables Used = A, J					

Figure 19. More SELECT, DELETE, and INSERT Results

ID Description		Tables Used	Local First Pass	Local Avg Last 4 Passes	Remote First Pass	Remote Avg Last 4 Passes
36 SELECT/SORT	100 Rows Returned	A	3.5	1.0	4.1	1.6
37 SELECT 5	5 Rows Returned	A	0.3	0.2	0.4	0.4
38 SELECT/SORT, SUBQUERY	9 Rows Returned	A,C	0.5	0.3	0.6	0.4
39 SELECT/SORT1, 2-T JOIN SUBQUERY	99 Rows Returned	A,G	26.4	26.2	27.6	27.4
40 SELECT/SORT, 2-T JOIN SUBQUERY	99 Rows Returned	A,G	27.4	26.3	28.6	27.5
41 SELECT/SORT, 3-T JOIN	9580 Rows Return.	B,G,H	104.0	104.0	146.0	147.0
42 SELECT/SORT1, 3-T JOIN SUBQUERY	99 Rows Returned	A,G,H	78.0	77.0	79.0	78.0
43 SELECT/SORT2, 3-T JOIN SUBQUERY	99 Rows Returned	A,G,H	79.0	77.0	80.0	78.0
44 SELECT, 4-T JOIN	100 Rows Returned	B,C,G,H	29.0	28.6	29.6	29.3
45 SELECT, 2-T JOIN	999 Rows Returned	A,G	17.1	16.6	25.6	24.1
46 INSERT INTO EMPTY TABLE	1000 Rows Inserted	A,I	17.3	21.4	17.4	21.6
47 UPDATE	1000 Rows Updated	A	14.5	13.5	14.7	13.2
48 DELETE	1000 Rows Deleted	A	23.4	19.4	23.2	19.7
49 INSERT	1000 Rows Inserted	A,I	35.9	26.6	37.0	26.7
50 SELECT, 3-T JOIN SUBQ (No Index)	98 Rows Returned	A,B,C	128.0	125.0	130.0	127.0

Figure 20. Complex Queries on Tables with Indices

ID Description	Local		Remote	
54 SELECT/ORDER BY - Table J	First Pass	Avg Last 4 Passes	First Pass	Avg Last 4 Passes
No. of rows = 1,000	132	129	144	143
No. of rows = 2,500	162	159	193	193
No. of rows = 5,000	211	209	280	280
No. of rows = 7,500	264	262	366	366
No. of rows = 10,000	321	321	463	462
No. of rows = 30,000	550	545	826	827
No. of rows = 50,000	802	797	1157	1152
No. of rows = 70,000	1091	1087	1577	1579
No. of rows = 100,000	1516	1530	2218	2232

Figure 21. SELECT/ORDER BY

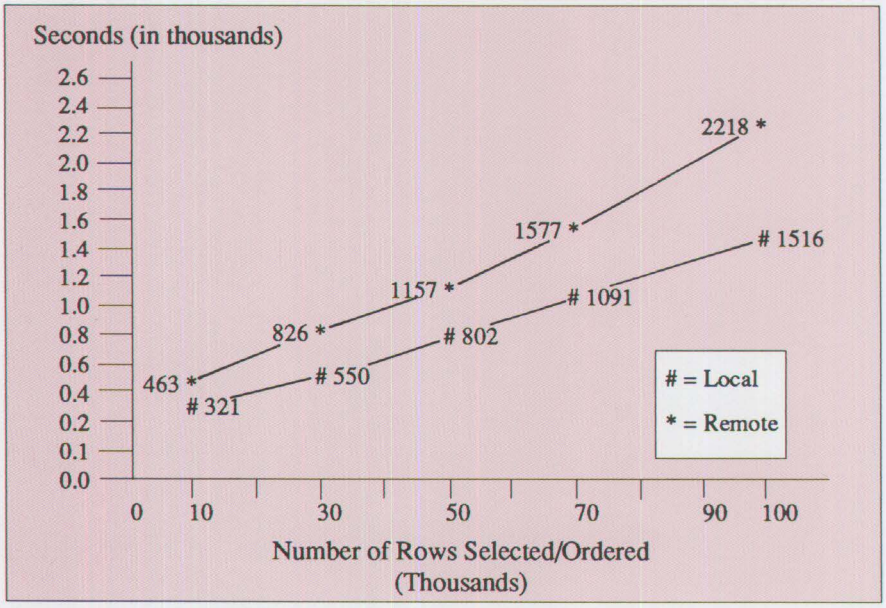


Figure 22. SELECT/ORDER BY

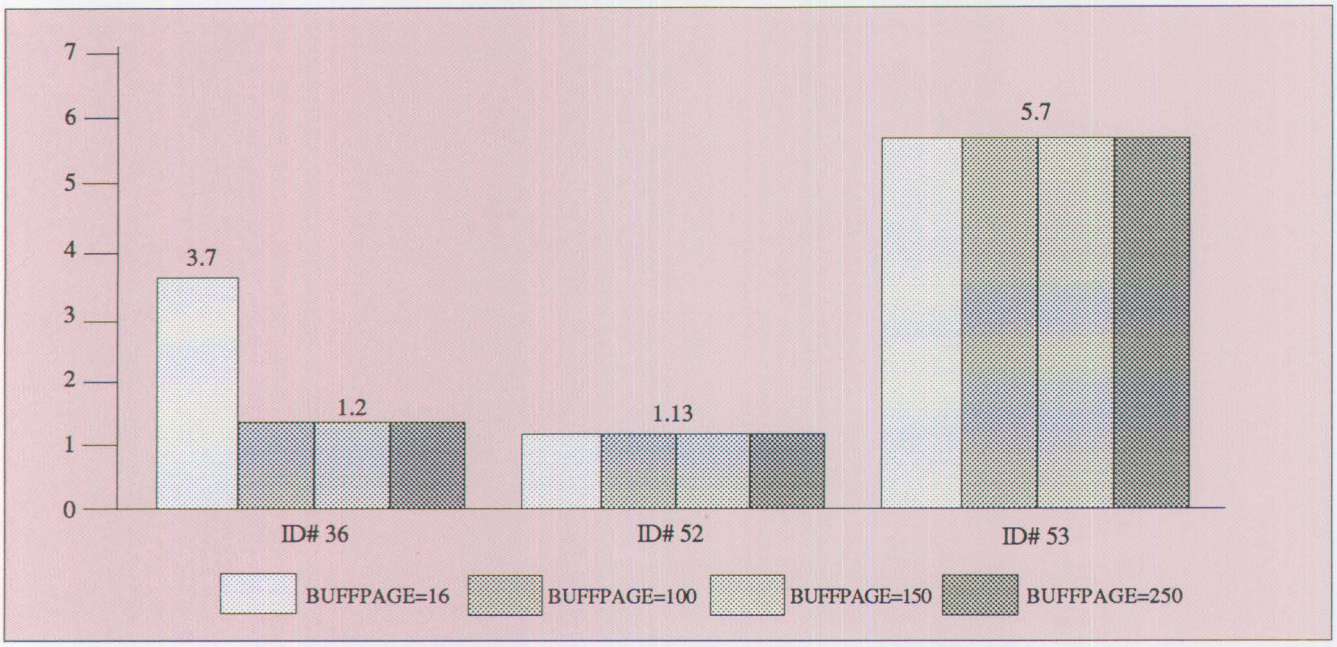


Figure 23. Short Response Time Group (Local)

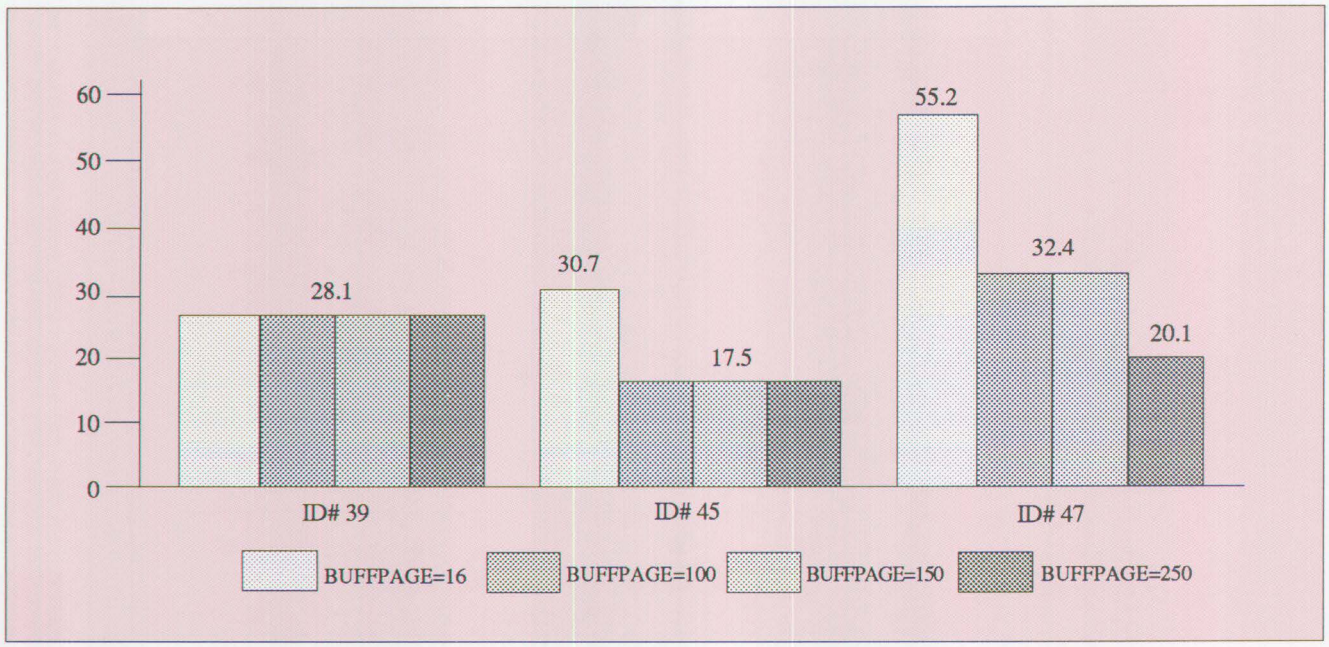


Figure 24. Moderate Response Time Group (Local)

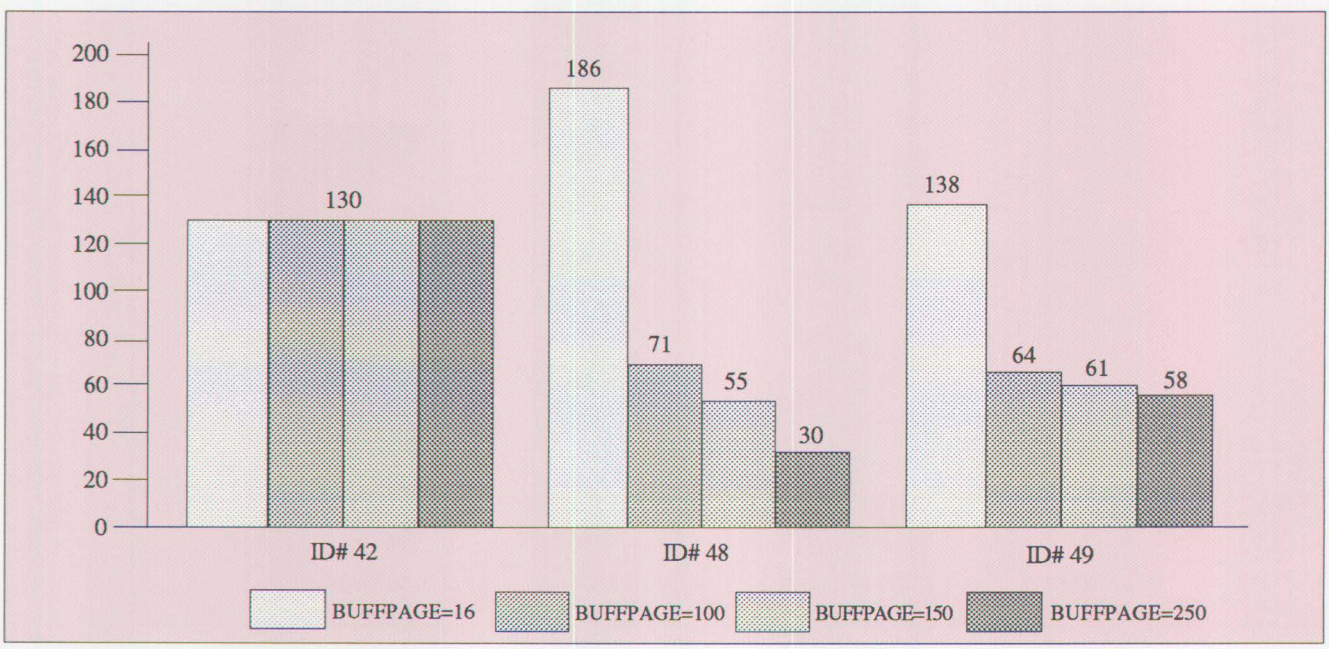


Figure 25. Long Response Time Group (Local)

Performance Tips

The following list of factors mentioned in this article will help improve OS/2 EE 1.2 Database Manager Performance. The advantages of each are discussed in detail in the *IBM Operating System/2® Extended Edition Version 1.2 Database Manager Administrator's Guide*, Chapter 7 – "Performance Monitoring and Tuning" (S01F-0267).

- Have adequate hardware
  - Processing power (in RDS environment, database queries can use Application Remote Interface process on the server)
  - RAM size (no swapping on server)
- Design efficient program/database
  - Table definition and normalization
  - Static versus dynamic SQL
  - Tuning queries
  - Index structure (identify performance and space costs)
  - Isolation level
- Tune configuration parameters
- REORG after extensive inserts, updates, or deletes
- Run RUNSTATS to update table statistics after
  - REORG
  - Loading new tables
  - Extensive modification of table contents
  - Creating indices
- Place log file on different physical drive
  - Avoids disk heads moving between data and log recordings
  - Facilitates overlapped I/O between two drives
  - Increases capacity for database and log file
- Use appropriate features to enhance Remote Data Services
  - Record blocking (default for Dynamic SQL is OFF)
  - Application Remote Interface, if possible
- Prototype and test (benchmark) to determine best configuration

There are many factors involved in designing database applications. The information in this article should provide guidance in making design decisions, which will result in a database with good performance.

ABOUT THE AUTHORS

Cesar R. Velasco is an advisory programmer in IBM's Entry Systems Division in Austin, Texas. Cesar was hired by IBM in 1962, and among his many assignments, he has developed simulation models for a number of systems, and received an IBM Outstanding Contribution award for one of them. Presently, he works in performance measurement, evaluation, analysis for the OS/2 Extended Edition Database Manager. He has authored a number of IBM papers. He received a B.S. in geodetic engineering from the University of the Philippines, and an M.S. in mathematics from Michigan State University. Cesar was a Fulbright and Smith-Mundt fellow in mathematics.

James Smith is an advisory engineer/scientist in IBM's Entry Systems Division in Austin, Texas. He started with IBM in 1966 in Huntsville, Alabama, where he worked on Apollo lunar missions and other government projects. Presently, he is responsible for developing OS/2 EE performance data. James received an IBM Outstanding Contribution Award for his work on a computer-controlled Target Drone Formation Control System, which IBM developed for White Sands Missile Range. James received a B.S. from Mississippi State University, an M.S. from Georgia Institute of Technology, and a Ph.D. from Southern Methodist University, all in electrical engineering.

Test Cases

Simple Queries (non-indexed)

The following statements were executed five times as a group.

ID

```
01    sqlestrd (dbname,"",'S',&sqlca) = START USING DATABASE
02    CREATE TABLE tablename
      (field1 integer NOT NULL,
      field2 integer NOT NULL,
      field3 integer NOT NULL,
      field4 char(21) NOT NULL,
      field5 char(21) NOT NULL,
      field6 char(21) NOT NULL)
06    DROP TABLE tablename
07    sqlestpd (&sqlca) = STOP USING DATABASE
08    sqluimp (dbname,"",datafile1,&dcoldata1,&tcolstrg1,"del",NULL, msgfile1,callact,&sqlca) = IMPORT
```

Note: **Each execution of this statement imported a different number of rows into an empty table.**

```
10    SELECT * FROM tablea WHERE field1 =  5000
11    SELECT * FROM tablej WHERE field1 = 55000
13    SELECT * FROM tableb,tablec WHERE tableb.field2 = tablec.field3
13    SELECT * FROM tablea,tablej WHERE tablea.field2 = tablej.field1
15    SELECT COUNT (*) INTO numrows FROM tablea/j WHERE field3 = 77
16    UPDATE tablea/j SET field6 = 'ZZZZZZZZZZZZZZZZZZZZ' WHERE field3 = 55 OR field3 = 69
```

Simple Queries (Indexed)

The following statements were executed five times as a group.

```
20    CREATE INDEX indxa123 ON tablea/j (field1,field2,field3)
20    CREATE INDEX indxa1 ON tablea/j (field1)
20    CREATE INDEX indxa2 ON tablea/j (field2)
20    CREATE INDEX indxa3 ON tablea/j (field3)
21    CREATE INDEX indxb1 ON tableb (field1)
26    SELECT * FROM tablea WHERE field1 =  5000
26a   SELECT * FROM tablej WHERE field1 = 55000
27    SELECT * FROM tablea/j WHERE field3 = 99 OR field3 = 999 ORDER BY field3, field2
28    SELECT * FROM tablea/j WHERE (field3 = 77 OR field3 = 99) AND field2 <=5000
29    SELECT * FROM tableb,tablec WHERE tableb.field2 = tablec.field3
29    SELECT * FROM tablea,tablej WHERE tablea.field2 = tablej.field1
31    SELECT COUNT (*) INTO numrows FROM tablea/j WHERE field3 = 77
32    UPDATE tablea/j SET field6 = 'ZZZZZZZZZZZZZZZZZZZZ' WHERE field3 = 55 OR field3 = 69
33    DELETE FROM tablea/j WHERE field3 = 0
34    INSERT INTO tablec/j SELECT * FROM tablea WHERE field3 = 55
```

Complex Queries (Indexed)

Each of the following statements was executed five consecutive times.

```
36 SELECT DISTINCT * FROM tablea WHERE field1 >= 0 AND field3 <= 9 ORDER BY field1
37 SELECT * FROM tablea WHERE field2 in (10,20,30,40,50)
38 SELECT DISTINCT sum(field2),field1,field2,field3,field4, field5,field6 FROM tablea WHERE field1 in
   (select field3 FROM tablec WHERE field3 >= 0 and field3 <= 9) GROUP BY
   field1,field2,field3,field4,field5,field6 ORDER BY tablea.field4
39 SELECT DISTINCT sum(field2),field1,field2,field3,field4, field5,field6 FROM tablea WHERE field1 in
   (select tableg.field3 FROM tablea,tableg WHERE tablea.field3 = tableg.field3)
   GROUP BY field1,field2,field3,field4,field5, field6 ORDER BY tablea.field4
```

Figure 26. Test Cases

```
40 SELECT DISTINCT sum(field2),field1,field2,field3,field4, field5,field6 FROM tablea WHERE field2 in
   (select tableg.field3 FROM tablea,tableg WHERE tablea.field3 = tableg.field3)
   GROUP BY field1,field2, field3,field4,field5, field6 ORDER BY tablea.field4
41 SELECT DISTINCT tableg.field1,tableb.field4,tableg.field5 tableh.field6 FROM tableb,tableg,tableh WHERE
   tableb.field3=tableg.field3 and tableb.field3=tableh.field3 ORDER BY tableg.field1
42 SELECT DISTINCT sum(field2),field1,field2,field3,field4, field5,field6 FROM tablea WHERE field1 in
   (select tablea.field3 FROM tablea,tableg,tableh WHERE tablea.field3 = tableg.field3
   and tablea.field3=tableh.field3) GROUP BY field1,field2,field3,field4,field5,field6 ORDER BY tablea.field5
43 SELECT DISTINCT sum(field2),field1,field2,field3,field4, field5,field6 FROM tablea WHERE field2 in
   (select tablea.field3 FROM tablea,tableg,tableh WHERE tablea.field3 = tableg.field3
   and tablea.field3=tableh.field3) GROUP BY field1,field2,field3,field4,field5,field6 ORDER BY tablea.field5
44 SELECT DISTINCT tableb.field5 FROM tableb,tableg,tableh,tablec WHERE tableb.field3=tablec.field3 and
   tableb.field3= tableh.field3 and tableb.field3=tableg.field3
45 SELECT * FROM tablea,tableg WHERE tablea.field1=tableg.field2
```

Each of the following statements was executed once. (*Note:* TABLEI grows from 0 to 5000 rows.)

```
46 INSERT INTO tablei SELECT * FROM tablea WHERE field2 <= 999
46 INSERT INTO tablei SELECT * FROM tablea WHERE field2 > 999 and Field2 <= 1999
46 INSERT INTO tablei SELECT * FROM tablea WHERE field2 > 3999 and Field2 <= 4999
46 INSERT INTO tablei SELECT * FROM tablea WHERE field2 > 6999 and Field2 <= 7999
46 INSERT INTO tablei SELECT * FROM tablea WHERE field2 > 8999 and Field2 <= 9999
```

Each of the following statements was executed once.

```
47 UPDATE tablea SET field3 = 1000 WHERE field2 <= 999
47 UPDATE tablea SET field3 = 2000 WHERE Field2 > 999 and Field2 <=1999
47 UPDATE tablea SET field3 = 4000 WHERE Field2 > 3999 and Field2 <= 4999
47 UPDATE tablea SET field3 = 7000 WHERE Field2 > 6999 and Field2 <= 7999
47 UPDATE tablea SET field3 = 10000 WHERE field2 > 8999 and Field2 <= 9999
```

Each of the following statements was executed once. (*Note:* TABLEA shrinks from 10000 to 5000 rows.)

```
48 DELETE FROM tablea WHERE field3 = 1000
48 DELETE FROM tablea WHERE field3 = 2000
48 DELETE FROM tablea WHERE field3 = 4000
48 DELETE FROM tablea WHERE field3 = 7000
48 DELETE FROM tablea WHERE field3 = 10000
```

Each of the following statements was executed once. (*Note*: TABLEA grows from 5000 to 10000 rows.)

```
49 INSERT INTO tablea SELECT * FROM tablei WHERE field2 <= 999
49 INSERT INTO tablea SELECT * FROM tablei WHERE field2 > 999 and Field2 <= 1999
49 INSERT INTO tablea SELECT * FROM tablei WHERE field2 > 3999 and Field2 <= 4999
49 INSERT INTO tablea SELECT * FROM tablei WHERE field2 > 6999 and Field2 <= 7999
49 INSERT INTO tablea SELECT * FROM tablei WHERE field2 > 8999 and Field2 <= 9999
```

The following statement was executed five consecutive times.

```
50 SELECT * FROM tablea WHERE field4 in (select tablea.field5 FROM tablea,tableb,tablec WHERE
   tablea.field5=tableb.field5 and tablea.field5=tablec.field5) ORDER BY tablea.field1
52 SELECT * FROM tablec ORDER BY field4
53 SELECT * FROM tableb ORDER BY field4
54 SELECT * FROM tablej WHERE (predicate) ORDER BY field2
```

***Note*: Each execution of this statement selected and sorted a different number of rows. Selection predicate referenced an indexed field.**

Figure 26. Test Cases

Database Tables

All the database tables used the same table definitions and contained the same number of fields. The differences in the tables were in the number of rows per table and the range of information in the fields. The fields may be described as follows, where N is the number of rows in a table:

Table X contains N rows:

Column	Width	Type	Content
FIELD1	4	Num	Unique integer, 1 to N, in ascending order from row to row
FIELD2	4	Num	Unique integer, 1 to N, in random order from row to row
FIELD3	4	Num	Integer, 1 to N/10, each integer repeated 10 times and distributed randomly from row to row
FIELD4	21	Char	N unique character strings with the format: AXXXXXXXXXXAXXXXXXXXXA where the A's are varied to make the strings unique and orderable; this field is in ascending order from row to row
FIELD5	21	Char	Same character string as in FIELD4 but distributed randomly from row to row
FIELD6	21	Char	N/10 unique character strings, each repeated 10 times and distributed randomly from row to row

The table sizes were then defined as follows:

```
TABLEA =     10,000  Rows
TABLEB =        500  Rows
TABLEC =        100  Rows
TABLEG =      1,000  Rows
TABLEH =      1,000  Rows
TABLEI - Initially empty; used for inserts
TABLEJ =    100,000  Rows
```

Figure 27. Database Tables

OS/2 EE 1.2 Competitive Performance

Comparisons for OS/2-based SQL servers were published in the Software Digest® *RATINGS REPORT*® June 1990, *LAN TIMES*™ July 1990, *DATA COMMUNICATIONS* September 21, 1990, and *PC World*™ October 1990. The information in these publications was provided by National Software Testing Laboratories (NSTL). Products compared by NSTL were Microsoft SQL Server™, SQLBase Server®, Oracle Server for OS/2®, and IBM OS/2 EE 1.2 Database Manager.

The Software Digest *RATINGS REPORT* gave a rating of "fair" to the OS/2 EE DBM product. Even though NSTL generally attempted to use performance enhancement features for all the products under comparison, the implementation of their benchmark for OS/2 EE DBM did not make use of some of its performance enhancement features.

IBM has since demonstrated that performance of the NSTL benchmark is significantly improved when the Record Blocking and Application Remote Interface (ARI) features of OS/2 EE DBM are properly used. The results of these measurements by IBM led NSTL to publish the Software Digest *BUYER'S ALERT*® Volume 7 Number 13, which stated, "...IBM OS/2 Extended Edition significantly outperforms the other SQL server products. In virtually every test, IBM's performance is comparable or superior to the competing products. NSTL verified IBM's results and ensures that IBM's implementation meets guidelines and definitions for each benchmark..." and, "With record blocking alone (without the ARI), IBM's performance becomes competitive on all tests." The article concludes, "IBM Extended Edition is a bargain... For one-third the price of other SQL servers (which require the purchase of LAN and operating system software), IBM provides a fully implemented product with some of the most sophisticated development tools available in a product of its kind. IBM Extended Edition's tools offer skilled developers the potential for performance comparable or superior to that available in much more expensive products."

An Intelligent Front-End EASEL Application

Alex Cabanes
IBM Corporation
Dallas, Texas

During the evolution of the keyboard, which was first created for terminals and later for PCs, the "standard" keypad layout for program function keys (PF keys) changed dramatically. Today's users invariably find PF keypad layouts different from the ones they learned on. With the introduction of graphical interfaces, a software solution has been found for this "hardware" problem – an application that can mimic and enhance a previous hardware limitation, resulting in an overall system that is easier to use.

PFKEYS, a small and simple front-end program for OS/2 Communications Manager™ (CM), utilizes EASEL 1.1 and OS/2 Extended Edition 1.2 as the graphical interface. This program demonstrates some of EASEL's new features and can be modified to meet the user's additional needs. The program is shown at the end of this article as Figures 6a through 6h. The figure lists the code in the left column, with comments on the right.

Because host sessions are predominantly character-based, and the OS/2 environment is predominantly graphics-based, a mouse can now be used to access both host and OS/2 sessions in a consistent manner.

Features

PFKEYS appears on the Presentation Manager desktop as a dialog box next to a host session window. The PF keys are displayed as push buttons within the dialog box. With a mouse, users can "point-and-click" on the button. This program can communicate with up to twenty-six logical 3270 host sessions using any of five "typical" PF keypad layouts. The actual number of host sessions will be limited by the user's particular Communication Manager (CM) configuration. This intelligent program searches for active host sessions, skipping past unavailable 3270 sessions. Using a PS/2 printer, it can also print an image of the host window.

Five different PF keypad layouts are supported, referred to as *box* layouts (Figure 1) and *line* layouts (Figure 2). The four types of *box* layouts organize the PF keys in a 4 X 3 matrix, whereas the *line* layout organizes PF keys in a 2 X 12 matrix. PF keypad layouts are shown in Figure 3.

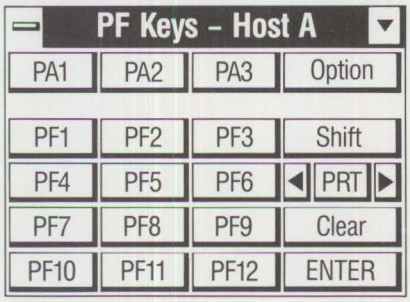


Figure 1. ***Box*** **Layout**

Default values are stored in the Layout_Id, StartX, StartY, and Ring variables. Current defaults are set to Layout_Id="A" (box 12), StartX=300, StartY=100, and Ring="ABCD...XYZ". By storing these values in variables, they can be changed upon invoking the EASEL program. Using the new EASEL 1.1 command line option ("-d" parameter), users can customize the program without having to recode the application. Variables and values are case-sensitive and must be typed *exactly* as shown in Figure 4.

For example, to invoke PFKEYS, use layout C on host sessions D and A (D being the first default) as shown in Figure 5.

Potential User Modifications

Intelligent Keys: A Dynamic Link Library (DLL) could be written to notify EASEL when a screen has been modified. The host screen would be scanned to determine if a PF key profile is available. If a profile was found, the text appearing in the PF key could display the associated program function. Printed keypad templates could be eliminated, because the PF keypad would always reflect the key's function.

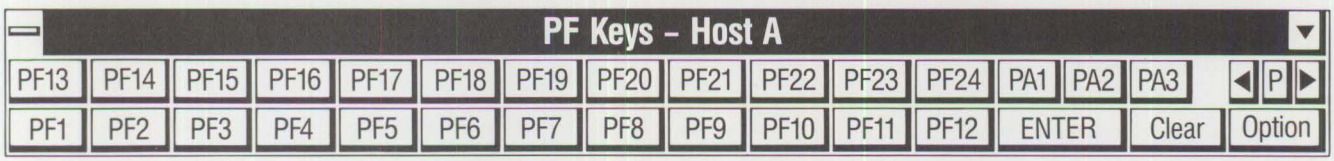


Figure 2. ***Line*** **Layout**

A) BOX 12 — Displays 12 PF keys, in a 4 x 3 pad, with PF12 in the lower right-hand corner. (Figure 1)

A.1) SHIFT — Same as above, except that PF24 is in the lower right-hand corner.

B) BOX 24 — Displays 12 PF keys, in a 4 x 3 pad, with PF24 in the upper right-hand corner.

B.1) SHIFT — Same as above, except that PF12 is in the upper right-hand corner.

C) LINE 24 — Displays 24 PF keys, in a 2 x 12 pad, with PF24 in the upper right-hand corner. (Figure 2)

Figure 3. PF Keypad Layouts

```
EASEL          PFKEYS          PFKEYS
Switch         Variable        Value
_______        ________        _______
   -d          Layout_Id       A
   -d          Layout_Id       B
   -d          Layout_Id       C

   -d          Ring            ABCD...XYZ

   -d          StartX          300
   -d          StartY          100
```

Figure 4. Program Variables

```
C:> easel pfkeys -d Layout_Id C -d Ring DA
```

Figure 5. Invoking PFKEYS

An additional program could simply have Up, Down, Left, and Right keys that could be redefined with the appropriate PF key sequence, depending on the screen profile.

Modifying this program to include these intelligent features would improve end users' interaction with the host environment. Furthermore, an additional DLL could be coded to capture the actual Up, Down, Left, and Right keystrokes from the keyboard and remap the functions to correspond to the screen profile. This addresses the problem in the host environment where the meanings of the PF keys change depending on the underlying host application, causing confusion among the end-user community.

Touchscreen: EASEL 1.1 has expanded support for touchscreen interactions. The sample program could be modified to support an expanded point-and-shoot interaction.

5250 Support: EASEL 1.1 now supports 5250 terminal emulation. Although the current version of CM does not support windowed 5250 sessions, the sample program could be modified to support the new emulation environment. This would bring the same benefits and features to the 5250 environment.

Summary

The use of intelligent software on workstations allows us to produce front-ends to the traditional host environments that are easier to use and incorporate the latest graphical interfaces. With programming products such as EASEL and OS/2 EE, applications that until recently took large programming staffs to produce can now be developed quickly.

ABOUT THE AUTHOR

Alex Cabanes is a market support representative in IBM's Product Support Operations in Dallas, Texas. Currently, he is the team leader for Lotus® 1-2-3/M™ support. Alex received a B.S. in information and computer science and an M.B.A. from the University of California at Irvine. In the past 10 years, Alex has held various positions in the computer industry. Prior to joining IBM in 1989, Alex was a systems analyst for a large hospital supply company.

```
screen size 640 480

include accel.inc
include pmptr.inc

module M3270

string variable  Session_Id is "?"
                 Layout_Id is "A"
                 Object
                 TitleBar
                 Ring is "ABCDEFGHIJKLMNOPQRSTUVWXYZ"        Ring contains valid host sessions.
                 TempRing

integer variable Count is 1
                  X Y T T1 T2
                  DoubleClick is 1000                       Timeout value for "double click" of PRINT key.

                  StartX is 300                             Starting X, Y positions on the PM desktop.
                  StartY is 100

string variable Dummy

integer PosX[12] is 00,30,60,00,30,60,00,30,60,00,30,60     Array stores X position for PF key layout in box layout.
integer PosY[12] is 30,30,30,20,20,20,10,10,10,00,00,00     Array stores Y position for PF key layout in box layout.

function TIMER() returns integer                            TIMER DLL returns the OS/2 system clock in milliiseconds.
   library "timlib"                                         Note: This DLL can be omitted and will only affect the "dou-
                                                            ble click" feature for the PRINT key.
###
###    PF Key Definitions
###

enabled invisible primary dialog region PFKey               Define the PF keypad.
   size 130 55 at StartX StartY
   in desktop
   border
   title bar ""
   system menu
   minimize button using "pfkeys.ico"                       You can create your own icon using the OS/2 EE 1.2 icon ed-
                                                            itor. A sample icon is provided with the article's title.

enabled invisible cancel push button Cancel                 Define PA keys (PA1, PA2, PA3).
   size 40 10 at 90 0
   in PFKey
   parameter is "Cancel"

enabled visible push button PA1_Key
   size 30 10 at 0 45
   in PFKey text "PA1"

enabled visible push button PA2_Key
   size 30 10 at 30 45
   in PFKey text "PA2"

enabled visible push button PA3_Key
   size 30 10 at 60 45
   in PFKey text "PA3"

enabled visible push button Option                          Used to choose between the different PF key layouts.
   size 40 10 at 90 45
   in PFKey text "Option"
```

Figure 6a. PFKEYS Program

```
enabled visible push button Shift
   size 40 10 at 90 30
   in PFKey text "Shift"

enabled visible push button Left
   size 10 10 at 90 20
   in PFKey text "◀"

enabled visible push button Print
   size 20 10 at 100 20
   in PFKey text "PRT"

enabled invisible push button ShadowPrint
   size 20 10 at 100 20
   in PFKey text "PRT"

enabled visible push button Right
   size 10 10 at 120 20
   in PFKey text "▶"

enabled visible push button Clear
   size 40 10 at 90 10
   in PFKey text "Clear"

enabled visible push button Enter
   size 40 10 at 90 0
   in PFKey text "ENTER"

enabled visible push button PF1
   size 30 10 at 0 30
   in PFKey text "PF1"

                .
                .
                .

###
###    Options Dialog
###

enabled invisible modeless dialog box OptionDialog
   size 75 60 at StartX StartY
   in desktop
   dialog border
   title bar "Options"
   system menu

enabled invisible cancel push button OCancel
   size 40 10 at 55 5
   in OptionDialog

enabled visible radio button A_Layout
   size 55 10 at 7 30
   in OptionDialog
   parameter is "A"
   group is LayoutGroup
   text "Box 12"

enabled visible radio button B_Layout
   size 55 10 at 7 20
   in OptionDialog
   parameter is "B"
   group is LayoutGroup
   text "Box 24"
```

Used to toggle between low-order and high-order PF keys.

Used to scan the host session ring in a "reverse" direction. Create the ◀ symbol by entering the OS/2 editor, holding down the Alt key, and typing 017.

Used to initiate the PRINT key sequence by taking a beginning timestamp.

ShadowPrint key is used to implement a double-click for the print key.

Used to scan the host session ring in a "forward" direction. Create the ▶ symbol by entering the OS/2 editor, holding down the Alt key, and typing 016.

Define the CLEAR key.

Define the ENTER key.

In the interest of brevity, PF2 through PF24 have been excluded from the article. The push buttons associated with PF2 through PF24 should use the same form as PF1 with appropriate modifications. Use the same size for all PF keys, and increment the position by 30 horizontally and 10 vertically.

Define selection buttons for the different PF keypad layouts.

Figure 6b. PFKEYS Program

```
enabled visible radio button C_Layout
   size 55 10 at 7 10
   in OptionDialog
   parameter is "C"
   group is LayoutGroup
   text "Line 24"

enabled visible group box Layouts
   size 60 40 at 4 10
   in OptionDialog
   text "Layout"

###
### Action Definitions
###

class LoPFKeys is   PF1   PF2   PF3   PF4   PF5   PF6
                    PF7   PF8   PF9   PF10  PF11  PF12

class HiPFKeys is   PF13  PF14  PF15  PF16  PF17  PF18
                    PF19  PF20  PF21  PF22  PF23  PF24

class SenseKeys is  PF1   PF2   PF3   PF4   PF5   PF6
                    PF7   PF8   PF9   PF10  PF11  PF12
                    PF13  PF14  PF15  PF16  PF17  PF18
                    PF19  PF20  PF21  PF22  PF23  PF24
                    Enter       Clear
                    PA1_Key     PA2_Key     PA3_Key

action DisableMouse is
   make point disposition mouse drop
   set pointer to SPTR_WAIT

action EnableMouse is
   set pointer to SPTR_ARROW
   make point disposition mouse normal

action WaitForKey is
   action DisableMouse
   action DefineWatch
   action WatchForNoX
   action WatchAndWait
   if (WatchGaveUp) then
      wait 0
   end if

   action EnableMouse

action RepositionBox is
   for each member of LoPFKeys loop
      extract from member
         skip by "PF"
         take number X
      if (Layout_Id = "A") then
         copy X to Y
      else
         copy (13-X) to Y
      end if
      change member position to PosX[X] PosY[Y]
      change member size to 30 10
   end loop
```

Low-order PF keys (01 through 12).

High-order PF keys (13 through 24).

Any keys that user can click on.

Generates the hourglass pointer to denote "Computer Busy."

Restores the mouse pointer to the normal arrow pointer.

Hook for future enhancements that may require capturing the host screen. Future enhancements are outlined at the end of the article.

Change PF key position for the *box* layout. Start by repositioning the low-order PF keys.

If the B layout was chosen, reverse the order of the PF keys.

Change the actual position and size of the PF key.

Figure 6c. PFKEYS Program

```
    for each member of HiPFKeys loop                    Let's compute the new positions for the high-order PF keys.
       extract from member
          skip by "PF"
          take number X
       copy (X-12) to X
       if (Layout_Id = "A") then
          copy X to Y
       else
          copy (13-X) to Y                              If the B layout was chosen, reverse the order of the PF keys.
       end if
       change member position to PosX[X] PosY[Y]        Change the actual position and size of the PF key.
       change member size to 30 10
    end loop

    change PFKey size to 130 55                         Resize the dialog region. This is a new EASEL 1.1 feature.
                                                        In the prior version, this would "stretch" the keys.

    change Left position to 90 20                       Reposition and resize the other non-PF keys. The ability to
    change Right position to 120 20                     resize a push button is a new EASEL 1.1 feature.

    change Print size to 20 10 change Print position to 100 20
    change Print text to "PRT"
    change PA1_Key size to 30 10 change PA1_Key position to 0 45
    change PA2_Key size to 30 10 change PA2_Key position to 30 45
    change PA3_Key size to 30 10 change PA3_Key position to 60 45
    change Option size to 40 10 change Option position to 90 45
    change Clear size to 40 10 change Clear position to 90 10
    change Enter size to 40 10 change Enter position to 90  0
    change ShadowPrint size to 20 10 change ShadowPrint position to 100 20
    change ShadowPrint text to "PRT"

    make Shift visible                                  Enable the SHIFT key. This allows the user to shift between
                                                        the low-order and the high-order PF keys.
action RepositionLine is
    for each member of LoPFKeys loop                    Change the PF key position for the line layout. Start by re-
       extract from member                              positioning the low-order PF keys.
          skip by "PF"
          take number X
       copy ((X - 1)  * 25) to Y
       change member size to 25 10                      Change the actual position and size of the PF key.
       change member position to Y 0
    end loop

    for each member of HiPFKeys loop                    Let's recompute the positions of the high-order PF keys.
       extract from member
          skip by "PF"
          take number X
       copy ((X - 13) * 25) to Y
       change member size to 25 10                      Change the actual position and size of the PF key.
       change member position to Y 10
    end loop

    change PFKey size to 400 20                         Resize the dialog region. This is a new EASEL 1.1 feature.
                                                        In the prior version, this would "stretch" the keys.

    change Left position to 370 10                      Reposition and resize the other non-PF keys. The ability to
    change Right position to 390 10                     resize a push button is a new EASEL 1.1 feature.

    change Print size to 10 10 change Print position to 380 10
    change Print text to "P"
    change PA1_Key size to 20 10 change PA1_Key position to 300 10
    change PA2_Key size to 20 10 change PA2_Key position to 320 10
    change PA3_Key size to 20 10 change PA3_Key position to 340 10
    change Option size to 30 10 change Option position to 370 0
    change Clear size to 30 10 change Clear position to 340 0
```

Figure 6d. PFKEYS Program

```
    change Enter size to 40 10 change Enter position to 300 0
    change ShadowPrint size to 10 10 change ShadowPrint position to 380 10
    change ShadowPrint text to "P"

    make Shift invisible                                    Disable the SHIFT key. This prevents the user from shifting
                                                            between the low-order and the high-order PF keys.

action ProcessShift is                                      Let's shift the PF keys. If the low-order PF keys are visible
    if (visibility of PF1) then                             (PF1), then show the high-order PF keys; else show the low-
        make HiPFKeys visible                               order PF keys.
        make LoPFKeys invisible
    else
        make LoPFKeys visible
        make HiPFKeys invisible
    end if

action ProcessKeys is                                       What key sequence should we send to the host session?
    switch (Object) is
        case char "PA1_Key" is                              PA1 key was clicked.
            copy PA1 to KeyCode
            action PressKey
            action WaitForKey
        case char "PA2_Key" is                              PA2 key was clicked.
            copy PA2 to KeyCode
            action PressKey
            action WaitForKey
        case char "PA3_Key" is                              PA3 key was clicked.
            copy PA3 to KeyCode
            action PressKey
            action WaitForKey
        case char "Clear" is                                The CLEAR key was clicked.
            copy CLEAR to KeyCode
            action PressKey
            action WaitForKey
        case char "Enter" is                                The ENTER key was clicked.
            action PressENTER
            action WaitForKey
        default is                                          Anything else should be a valid PF key.
            extract from Object
                skip by "PF"
                take number PFkeyNumber
            action PressPFkey
            action WaitForKey
        end switch

    action ChangeTitleBar is                                This will change the title in the titlebar to reflect the at-
        copy "PF Keys - Host " SessionName to TitleBar      tached host session.
        change PFKey text to TitleBar
        wait 0

    action LookLeft is                                      Start reverse scan; that is, start looking at the end of the
        for T = 1 to length of Ring loop                    ring and going left (reverse).
            extract from Ring
                skip to last
                skip by -1
                take to last Session_Id
                skip to 1
                take to Session_Id Ring

            copy Session_Id Ring to Ring                    Place the extracted host session ID at the beginning of the
                                                            host session ring.
```

Figure 6e. PFKEYS Program

```
        if (Session_Id != SessionName) then
           copy Session_Id to SessionName
           action ChangeTitleBar
           action Connect
           if (A3270ErrorCode = E_SESSNAME) then
              copy "?" to SessionName
              action Disconnect
           else
              action SetEmulatorWindow
              action Disconnect
              leave loop
           end if
        end if
     end loop

action LookRight is
     for T = 1 to length of Ring loop
        extract from Ring
           take to 2 Session_Id
           take to last Ring

        copy Ring Session_Id to Ring


        if (Session_Id != SessionName) then
           copy Session_Id to SessionName
           action ChangeTitleBar
           action Connect
           if (A3270ErrorCode = E_SESSNAME) then
              copy "?" to SessionName
              action Disconnect
           else
              action SetEmulatorWindow
              action Disconnect
              leave loop
           end if
        end if
     end loop

action ClearShadow is
     make ShadowPrint invisible
     make Print visible

action ProcessLayout is
     switch (Layout_Id) is
     case char "C" is
        action RepositionLine
        make LoPFKeys visible
        make HiPFKeys visible
        check C_Layout
     case char "B" is
        action RepositionBox
        make LoPFKeys invisible
        make HiPFKeys visible
        check B_Layout
     default is
        copy "A" to Layout_Id
        action RepositionBox
        make HiPFKeys invisible
        make LoPFKeys visible
        check A_Layout
     end switch
```

If not already connected to that host session, try to establish a connection.

Unable to establish connection. Let's try again!

Okay, connection established. Disconnect and leave the loop. Found a good host session.

Start forward scan; that is, start looking at the beginning of the ring and go to the right (forward).

Place the extracted host session ID at the end of the host session ring.

If not already connected to that host session, try to establish a connection.

Unable to establish connection. Let's try again.

Okay, connection established. Disconnect and leave the loop. Found a good host session.

This resets the shadow PRINT key.

User selected *line* 24 layout.

User selected *box* 24 layout.

User selected *box* 12 layout.

Figure 6f. PFKEYS Program

```
    make OptionDialog invisible                    Hide options menu and redisplay the PF keypad.
    action ChangeTitleBar
    change PFKey position to StartX StartY
    make PFKey visible
    action ClearShadow
    wait 0                                         Refresh the PF keypad.

###
### Response Definitions
###

response to start                                  Startup process.

    change PFKey position to StartX StartY         Reposition dialog regions.
    change OptionDialog position to StartX StartY
    copy Session_Id to SessionName
    action Init3270                                Initiate 3270 host sessions.

    copy EW_ACTIVATE to EmulatorWindowCommand       Make CM host session active, just as if the user had clicked
                                                   on the host session using the mouse.

    if (length of Ring < 2) then                   If the host session ring was overridden, and a single host ses-
       copy Ring Ring to Ring                      sion was defined, then duplicate the defined host session so
       disable Left in PFKey                       the initial forward scan will find the host session.
       disable Right in PFKey
    end if
    action LookRight                               Forward scan the host session ring.
    action ProcessLayout                           Process the PF keypad layout in case the initial default was
                                                   overridden.
response to SenseKeys
    copy object to Object                          Temporary fix for EASEL.
    action Connect                                 Connect to the host session, just for duration of the key
    if (A3270ErrorCode = E_SESSNAME or             transmittal.
       A3270ErrorCode = E_INUSE) then
       action Disconnect
    else
       action ProcessKeys                          Send clicked key to host session.
       copy EW_ACTIVATE to EmulatorWindowCommand    Activate host session, just like the user had clicked on the
       action SetEmulatorWindow                    host session. In this manner, if the keyboard is used to enter
                                                   some host commands, EASEL will be out of the way, and
                                                   the user will interact directly with the host session.

       action Disconnect                           By disconnecting from the host session immediately after
    end if                                         the keys have been sent to the host session, EASEL will not
                                                   interfere with other PC/host programs (like file transfers).

    action ClearShadow                             Clear the shadow PRINT key, because if it was activated, it
                                                   was a mistake.

response to Option                                 User wishes to change PF keypad layout.
    copy xposition of PFKey to StartX
    copy yposition of PFKey to StartY
    change OptionDialog position to StartX StartY
    make PFKey invisible
    make OptionDialog visible
    action ClearShadow
    wait 0                                         Refresh PF keypad.

response to Shift
    action ProcessShift
    action ClearShadow
    wait 0                                         Refresh PF keypad.
```

Figure 6g. PFKEYS Program

```
response to Print                                  User clicked on the PRINT key.
   copy TIMER() to T1                              Let's take a timestamp to see if the user really meant to
   make ShadowPrint visible                        print.
   make Print invisible

response to ShadowPrint                            Okay, user clicked on the shadow PRINT key.

   copy TIMER() to T2                              Let's compare this timestamp to the previous timestamp to
   copy (T2 - T1) to T1                            see if the user really wanted this as a double click or
                                                   whether this procedure should time out.

   if (DoubleClick > T1) then                      The user really wanted this as a double click. Connect,
      action Connect                               print, and disconnect.

      action Print3270Screen                       The Print3270Screen is a new EASEL 1.1 feature.
      action Disconnect
   end if
   action ClearShadow                              Clear the shadow PRINT key. The screen print either
                                                   worked, or timeout occurred.

response to A_Layout in OptionDialog               User selected box 12 PF keypad layout.
   copy parameter of object to Layout_Id
   action ProcessLayout

response to B_Layout in OptionDialog               User selected box 24 PF keypad layout.
   copy parameter of object to Layout_Id
   action ProcessLayout

response to C_Layout in OptionDialog               User selected line 24 PF keypad layout.
   copy parameter of object to Layout_Id
   action ProcessLayout

response to Left                                   Reverse scan of host session ring.
   copy EW_ACTIVATE to EmulatorWindowCommand
   action LookLeft
   action ClearShadow

response to Right                                  Forward scan of host session ring.
   copy EW_ACTIVATE to EmulatorWindowCommand
   action LookRight
   action ClearShadow

response to Cancel
   action Stop3270
   exit

response to OCancel
   action Stop3270
   exit

response to PFKey on close
   action Stop3270
   exit
```

Figure 6h. PFKEYS Program

Enabling Software for National Language Support

Jennifer Wilson
IBM Corporation
Boca Raton, Florida

This article discusses where focus should be placed for National Language Support (NLS), and how to design enablement into a software product.

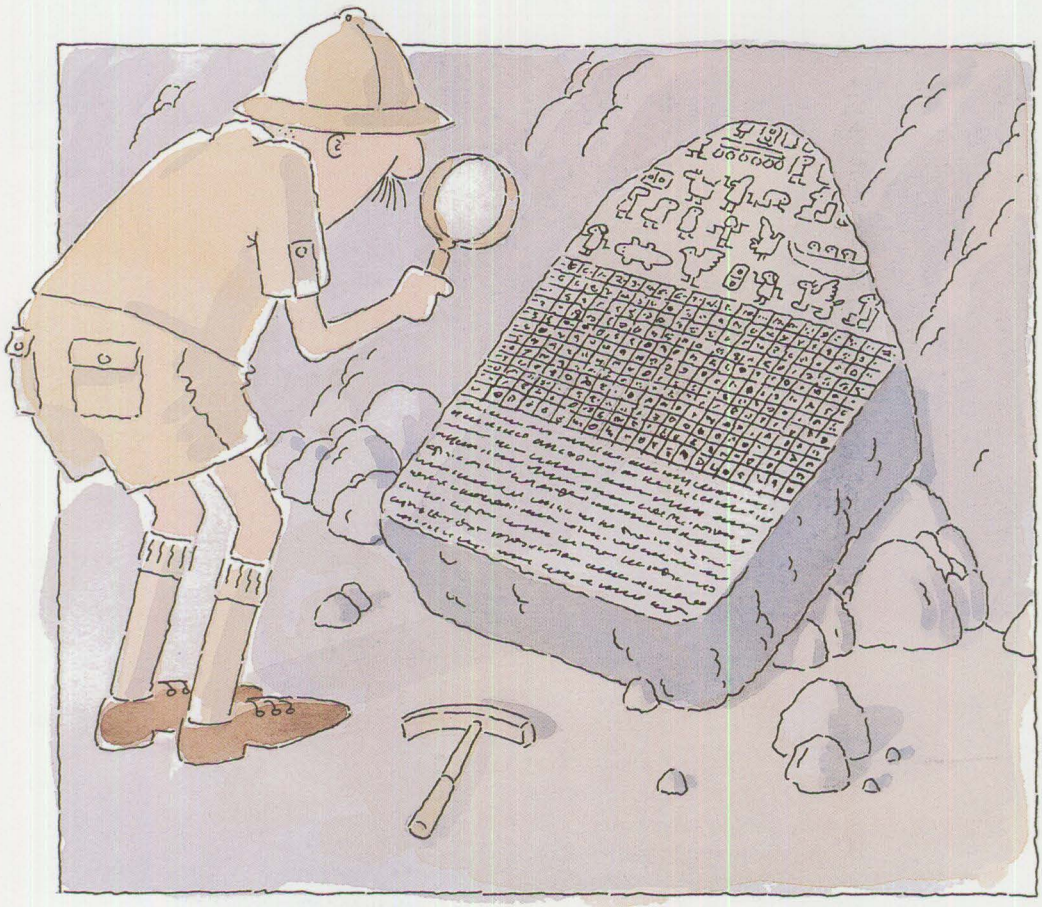
Europe '92 (the 12-nation European Community) is just around the corner, and technological advances from Japan and other foreign countries are abundant. Competition among computer software products has become fierce. To compete in the increasingly important global market, United States software companies must be able to easily transform their software for use in other countries.

Enabling Software

Any software product can probably be rewritten to be used in a country with its unique language, characters, and formats. However, to be able to do this quickly and easily for any country, the product must be *enabled* for translation. This means that the design for the base U.S. product is created with consideration for the functions and uses required for the other country versions. Implementing the national language occurs when functions for a specific country are added to an enabled base product.

Enabling and implementing a product is much easier, less costly, and less time-consuming than trying to change a product that was not originally enabled in its design.

In order to enable a product, code should be structured to make modifi-



Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	ð	Ó	-
-1	☺	◄	!	1	A	Q	a	q		ü	æ	í	▓	┴	Ð	ß	±
-2	☻	↕	"	2	B	R	b	r		é	Æ	ó	▓	┬	Ê	Ô	=
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	Ë	Ò	¾
-4	♦	¶	$	4	D	T	d	t		ä	ö	ñ	┤	─	È	õ	¶
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	Á	┼	ı	Õ	§
-6	♠	▬	&	6	F	V	f	v		å	û	ª	Â	ã	Í	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ù	º	À	Ã	Î	þ	¸
-8	◘	↑	(	8	H	X	h	x		ê	ÿ	¿	©	╚	Ï	Þ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ö	®	╣	╔	┘	Ú	··
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Û	·
-B	♂	←	+	;	K	[	k	{		ï	ø	½	╗	╦	▉	Ù	1
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▬	ý	3
-D	♪	↔	-	=	M	]	m	}		ì	Ø	¡	¢	═	¦	Ý	2
-E	♫	▲	.	>	N	^	n	~		Ä	×	«	¥	╬	Ì	-	■
-F	☼	▼	/	?	O	_	o	⌂		Å	ƒ	»	┐	¤	▀	´	

Figure 1. Multilingual Code Page

cation easy. Areas to focus on for enabling a single-byte, left-to-right language include character sets and code page support, machine-readable information (MRI), and national language-dependent functions.

Bidirectional and DBCS Languages

There are additional enabling requirements when a product is to be translated into a language that is bidirectional or uses double-byte characters. Bidirectional languages are those in which most text is read from right to left, but some characters and numbers are read from left to right. Many Middle Eastern languages are read this way.

Double-byte character sets (DBCS) are used by some Far East nations, such as Japan and China. Their languages use many different ideographic symbols, causing the need for two bytes per symbol to allow for the larger number of symbols. Because single-byte characters can also be used, careful design consideration is necessary.

This article discusses enabling for single-byte, left-to-right languages only, used by many American and European countries.

Design Considerations

Some basic design considerations for NLS should be implemented early in the design phase. Code should be modular so it can be modified easily. Components that will be modified for NLS (such as code page and date format support) should be in modules and files separate from the base code.

Character Sets and Code Pages: Every country has a set of characters that form the words of the language. These characters are alphabetic, numeric, and special, such as punctuation characters. Each character is assigned a unique code point, which is a hexadecimal code. The tables that assign code points to characters are called code pages. Figure 1 shows a code page. The character "J" in that code page is assigned code point 4A. Many countries use several different code pages, and some countries use several languages. French, German, and Italian are all accepted languages in Switzerland.

Hex Code	Swedish Code Page Character	French Code Page Character
5B	Å	$
6A	ö	ù
82	b	b
84	d	d
87	g	g
89	i	i
92	k	k
93	l	l
95	n	n
B1	£	#
C0	ä	é
D0	å	è

Swedish phrase: Ål åbäkig ökänd £100

Same phrase with French code page: $l èvékig ùkénd #100

Figure 2. Misspelling From Using Different Code Pages

Because so many different character sets and mappings were being used by different countries, the Latin 1 character set was created. This is a combination of the character sets of many areas, including the Americas and Western Europe. A code page that uses the Latin 1 character set is called a Country Extended Code Page (CECP). Any country-unique characters are added to the Latin 1 characters to form the code page.

In addition to a character set and code page, an *encoding scheme* must be used. This definition tells what code point values are assigned to control characters and graphic characters. Any code point that is not assigned in code pages must not be assigned to any characters by an application. Two encoding schemes are EBCDIC and ASCII. EBCDIC is used primarily with large systems, while ASCII is mostly for small systems.

In order for applications to work correctly in different countries, the same code page and encoding scheme should be used. If they are not, the numeric code points will be interpreted differently, and the messages displayed and entered will appear as nonsense, as shown in Figure 2. Applications should allow users to select the character set and code page they want to use.

Not all languages have both upper- and lowercase characters. Where languages have both types of characters, the associated case characters must be definable. Any search must account for both the lower- and uppercase characters.

Some languages use characters with symbols, such as tildes. These characters are treated uniquely, so the order in which the characters are sorted must be selectable. The order is how a dictionary, for instance, would order the letters. As shown in Figure 3, some country names are not always sorted on the first letter of the last name. The ability to sort using the country's method must be allowed.

Some characters have special control uses, so they perform a function rather than representing a character. The apostrophe is often used to indicate the beginning and end of a text string. In some languages, the apostrophe is used in words that could be in the text string. To avoid confusion, users must be able to substitute other characters for control characters whenever these situations occur.

Some printers or screens may not have the capacity to display all the characters of a character set. For any character that cannot be printed

In some countries, the names:

del Zoppo
van der Vliet
van der Hoff
d'Marco

would not be sorted on the first letter:

van der Hoff
d'Marco
van der Vliet
del Zoppo

In France, the characters:

e, é, è, ê

are sorted:

é, è, ê, e

Figure 3. Sorting Order

For English Phrases	Allow Expansion
up to 10 chars	101 - 200%
11 - 20	81 - 100
21 - 30	61 - 80
31 - 50	41 - 60
51 - 70	31 - 40
over 70	30

Figure 4. Translation Expansion

or displayed, a substitute character must be definable. This process is called *folding*.

Any non-character graphical symbols, or *icons*, that are used must be universally understood. An example is the hourglass, which represents time passing while action is taking place.

When keyboards do not have the capacity to allow one key per character, multiple definitions are given. In this situation, users must be given the option to choose which keyboard definition to use. They should have the option of selecting a logical keyboard layout where the desired characters are available.

Machine-Readable Information: Some of the most obvious differences between various countries' versions of the same product are seen in the text displayed on the screen. This text, which includes messages, prompts, user keyboard input, help information, menus, and text in graphics images, is called *machine-readable information* (MRI).

For MRI, two areas to focus on are content and packaging. Content is information displayed or entered, and packaging is the location of the MRI in the code and on the diskettes.

One of the most important things to remember when writing MRI is to keep it short and simple. The translation of an English phrase into another language can double or even triple the number of characters in that phrase. See Figure 4 for an example of how much space should be reserved.

When writing MRI, you should use proper grammar and syntax, without slang, jargon, jokes, or abbreviations. Consistency is also important. Try to use the same words when conveying the same idea. If technical or unusual words are used, a glossary or description of the keywords and concepts may help translators understand the meanings. Differences in culture make it difficult to translate concepts and ideas that differ among countries.

English: &1 returned by function &2

German: Die funktion &2 endete mit dem Fehlercode &1

&1 = Return code
&2 = Function name

Figure 5. Message with Variable Order Changed

Any text should be a complete entity to be translated. Individual words or parts of words (such as "day" in the days of the week) should not be translated literally or assumed to be in a certain order in the sentence, as shown in Figure 5. Different languages use unique sentence structures to convey an idea.

For this reason, if a variable is used as part of a message, it must be able to be positioned wherever necessary. For example, a variable may be used if a message contains an error code number.

Keywords, messages, and responses must never be hard-coded. In English, the responses "Y" and "N" for "Yes" and "No," for example, must be translatable. User input also must not be required to be in upper- or lowercase.

The way MRI is packaged can make a big difference in the time and money necessary to implement a national language version of a product. All MRI should be separate from any executable code. If the product requires several disks, ideally all the MRI should be on one disk. This allows each country to need to change and produce as few diskettes as possible.

By isolating the MRI from the code, MRI translation and code development can occur simultaneously, allowing the national language products to get to market more quickly. Isolation also protects the code from being accidentally changed by the translators. Many translators are not programmers and would find it difficult and time-consuming to locate and change text strings in code. By separating the MRI from the code, a higher-quality translation probably results as well.

Time:
```
16:42
 4:42 p.m.
```

Date:
```
1990-04-29
   4.29.90
  90-29-04
   4/29/90
  29/04/90
  29.04.90
  90/04/29
```

Calendar:
```
Gregorian
Islamic
Chinese Lunar
Buddhist Era
Japanese
Hebrew
```

Negative number format:
```
(10)  [10]  -10  10-
```

Currency:
```
Albania    Lek  12.345,67
Canada (Eng.)$  12,345.67
       (Fr.)    12 345.67 $
Italy       L.  12.345
Netherlands F   12.345,67
Portugal        12.345,67 Esc.
Norway     kr   12.345,67
Sweden          12.345,67 kr
```

Keywords:
```
Yes  Oui  Si  Ja
```

Figure 6. National Language-Dependent Functions

The following presentation controls also must follow MRI rules:

- Highlighting coordinates
- Protected fields
- Centering
- Label fields
- Mouse selection
- Fields
- Function key areas
- Scrolling

Imagine a function-key area with F1=Help and F3=Exit as two hard-coded fields, each seven characters long, with two spaces between them. After the product is translated, the Help field is now 11 characters long. If the mouse pointer is clicked on the last character of the first word, the Exit function, rather than Help, is now performed. Because these items are related to the MRI and are code, they should be separated from both the MRI and the rest of the code. This is so the base code does not have to be changed when implementing national language products.

Code function itself should not be dependent on any translated information. For example, if one function is a prerequisite for another, the function should not have a hard-coded name associated with it. A number or flag should be used instead.

Another way to ease translating is to number all messages and panels. All messages could be put in one master message file. Numbering messages and panels makes translation easier, especially if the person testing the translated version does not speak the language.

National Language-Dependent Functions: Some items unique to certain countries should be selectable for both keyboard entries and screen displays. These include the following items:

- Time symbol and format
- Date symbol and format
- Calendar scheme
- Holidays
- Number of days per year
- Number of months per year
- Numeric punctuation, including decimal and thousandths separator, and format for negative numbers
- Rounding rules
- Currency symbol, format, and field size
- Measurement system, including weight, volume, and distance
- Keywords, such as Yes/No, Female/Male
- Sorting order
- Paper size

Examples of some of the representations used are shown in Figure 6.

Additional details and a complete set of rules and guidelines are in *National Language Information and Design Guide Volume 1, Designing Enabled Products, Rules and Guidelines*, SE09-8001. By following these rules when designing software products, you will be more successful in the global marketplace.

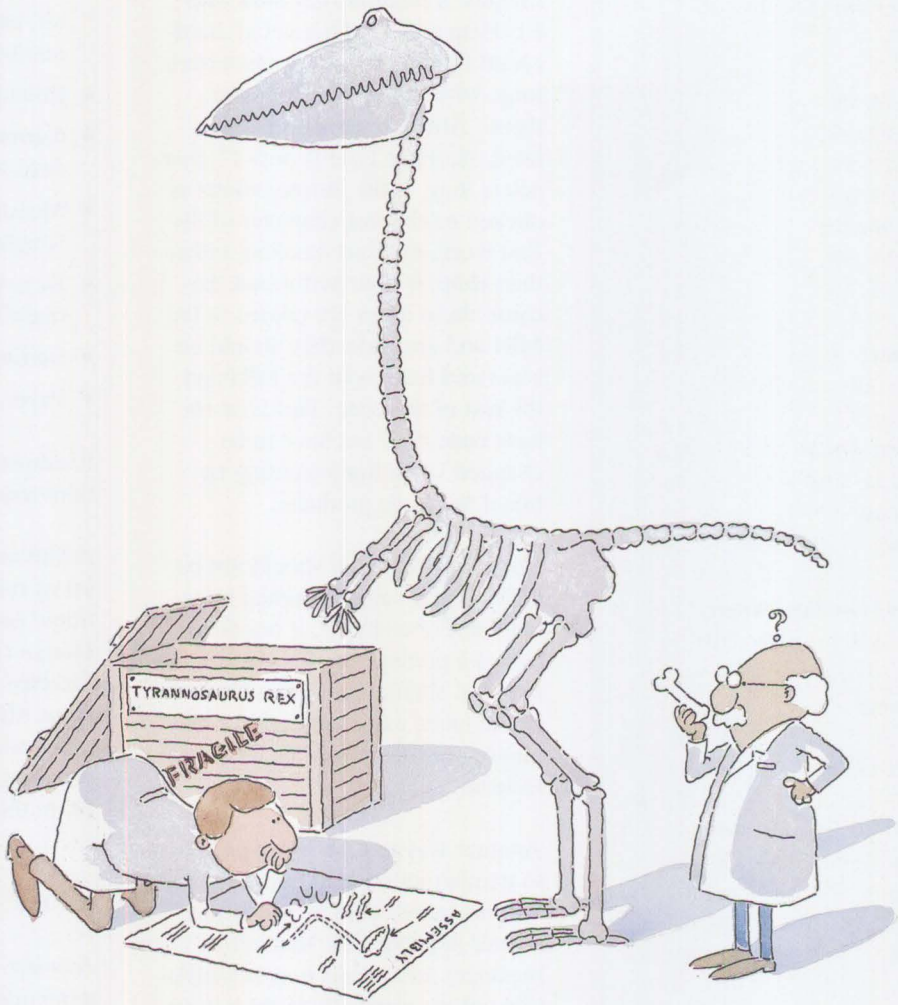
ABOUT THE AUTHOR

Jennifer Wilson is a senior associate programmer in IBM's Entry Systems Division (ESD) in Boca Raton, Florida, working in OS/2 Systems Assurance. She recently finished a temporary assignment working in National Systems Division OS/2 support. Since joining IBM in 1985, she has also worked in the design and development of DOS and interface tools, and educational software testing. Jennifer has a dual degree in computer science and mathematics from Pennsylvania State University, and an M.B.A. from Nova University.

SNA Definitions for 3270 Emulators

William J. Wen
Houston, Texas

The thigh bone is connected to the hip bone; the hip bone is connected to the backbone. But do you know what needs to be connected to what, before your 3270 emulator will communicate with a System/370 host? In this two-part article, we will explore which parameters need to match between the 3270 emulator and the System/370 host, as well as between the intermediate communication facilities.



Do you remember the first time you needed to configure a PC 3270 emulator to communicate via a token ring network (TRN) through a 3174? Or, how about through a Network Control Program (NCP) TRN gateway, such as a 3720, 3725, or 3745? Do you remember wondering which parameters needed to match which other parameters in the various packages (PC 3270 emulator configuration options, 3174 configuration, NCP, and VTAM definitions)?

Systems Network Architecture™ (SNA™) communications provide a versatile means of connectivity, adaptive to many possible communications scenarios. However, with flexibility comes complexity, and, for those unfamiliar with SNA, a degree of confusion.

Documentation wasn't necessarily lacking; you can get access to references detailing how to configure any of the specific packages. But wouldn't it be great if you could find an SNA overview and a subsequent application of these SNA concepts to actual communication scenarios?

The objective of this article is to give readers a comprehensive view of specific configurations. It is presented in two parts. Part I overviews the concepts behind Systems Network Architecture (SNA) communications. Part II, which will appear in the next issue, contains specific configuration examples.

Scope

There is a myriad of possible host connectivity options. To attempt to cover all possible connectivity options in one article would definitely be impractical. Consequently, the scope of this article is limited to only the following conditions:

1. The SNA environment.

2. The SDLC protocol for remote communications.

3. Distributed Function Terminal (DFT) mode 3270 emulators.

4. 3270 sessions through the TRN.

Condition 1 is present because most 3270 configurations in the TRN environment require SNA; the one exception to this is covered in the section "LAN Connectivity in 3270 Emulation in the Non-SNA Environment," which begins on page 72.

Condition 2 is present because of implementation options. SNA supports several remote communications protocols: SDLC, X.21, and X.25. SDLC is the most widely used protocol in the United States, and the hardware necessary for this protocol is available both at the PC level (for 3270 emulators) and in some of the components to which the 3270 emulators are attached. Consequently, almost all connectivity questions involving remote communications deal with the SDLC protocol.

Condition 3 is present because most 3270 configurations in the TRN environment involve DFT-mode emulation. Control Unit Terminal (CUT)-mode emulation is covered in the section "CUT-Mode Emulation in the TRN Environment," which begins on page 74.

Physical Units (PUs) and Logical Units (LUs)

Systems Network Architecture (SNA) defines a seven-layer division, as shown in Figure 1.

The seven layers define specific functions. Each layer has specific responsibilities upon which the adjacent layers depend. The division of responsibilities to specific layers helps in its implementation and in isolating problems to specific layers

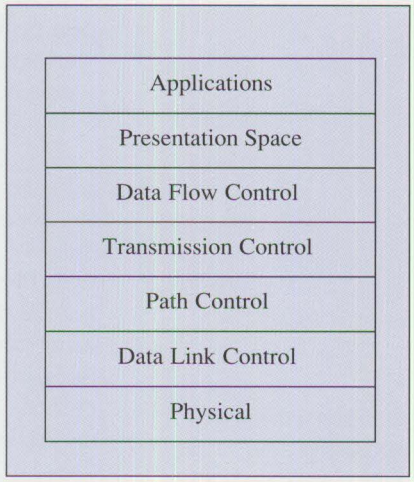


Figure 1. Seven Layers of SNA

(by relating the problem to specific functions). Because functions are isolated to layers, problems discovered in one layer may be corrected without affecting other layers. For more details about other SNA advantages, refer to *Systems Network Architecture Technical Overview*, GC30-3073, and *Systems Network Architecture Concepts and Products*, GC30-3072.

Layers are implemented using LUs and PUs. Three SNA layers are implemented by LUs: presentation space, data flow control, and transmission control. PUs together with their communications hardware implement the lower three SNA layers: path control, data link control, and physical. The top layer of application is the host application at the host side, and the user at the terminal side.

Depending on which features an LU or PU is designated to perform, various LU and PU types are defined. The LU types defined are: 1, 2, 3, 4, 6.1, 6.2, and 7. This article mainly deals with LU type 2, which is the LU type designated for 3270 displays. LU types 1 and 3, which are 3270 printer LUs, are also discussed. The PU types defined are: 2.0, 2.1, 4, and 5. This article deals mainly with PU type 2.0, which is required by LU types 1, 2, and 3. Because LU types 1, 2, and 3 require PU support, these LU types are always defined under a PU.

Basic Attachments to the Host

There are essentially three methods of attaching to the host: one locally and two remotely. These methods are specific about how a PU communicates to the host where the PU is defined. A PU 2.0 is defined at one host only; therefore, all LUs under PU 2.0 are also defined at the same host, and only at this host. It is important to note that such a configuration does not restrict LUs to communicate only with the host where the LUs are defined. An LU may use the host's cross-domain facility to communicate to another host through the host network.

One method is a local attachment to the host, as shown in Figure 2.

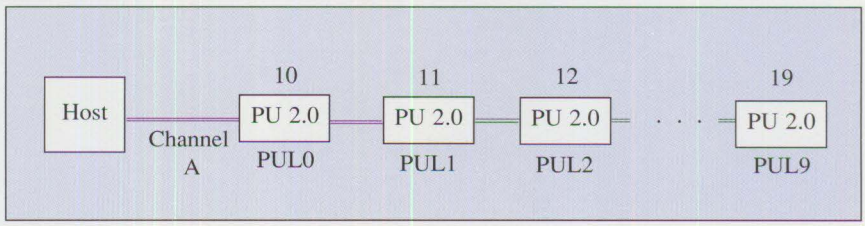


Figure 2. Local Host Attachment (bus and tag)

Diagram		VTAM Definitions		
PUL0 ---------->	DEFPUL0	PU	CUADDR=A10	
			:	
PUL1 ---------->	DEFPUL1	PU	CUADDR=A11	
			:	
PUL9 ---------->	DEFPUL9	PU	CUADDR=A19	
			:	

Figure 3. Local PU Attachment to VTAM Definitions

The local attachment is also called the channel attachment, because the local connection is attached to the host channel. The method of attaching multiple PUs on the same channel using a daisy chain is called a bus-and-tag connection. Because there is more than one PU on the same channel, there must be a way to distinguish one PU from another on the same channel. This method is implemented by designating each PU with a unique address on the channel, called the subchannel address. The subchannel address of the individual PU is included in that PU's customization. The subchannel address may range from X'00' to X'FF', though in Figure 2, only subchannel addresses of X'10' to X'19' are used. At the host's communications definitions, or Virtual Telecommunications Access Method™ (VTAM™) definitions, these PUs are distinguished using the CUADDR= field. This field is three bytes long: the most significant byte is the channel address, and the remaining bytes designate the subchannel address. For Figure 2, the PUs are associated to the VTAM definitions as shown in Figure 3.

Two methods of remote attachment are available: switched point-to-point and dedicated multidropped. A dedicated, remote communications line means a data-grade line with no phone switches between the connection; therefore, no phone number needs to be dialed, because the target device is always on the same dedicated line. A switched connection means a voice-grade line where a phone number is required to make the connection to the target device. A dedicated connection, sometimes called a nonswitched connection or leased line, is shown in Figure 4.

Like the local attachment, the nonswitched connection may have more than one device per communications line; hence, the term multidropped for the remote, nonswitched connection. The VTAM definition that distinguishes one PU from another on a dedicated line is called the ADDR= field, sometimes referred to as the SDLC station address. The ADDR= field may range from X'01' to X'FE', though Figure 4 uses only X'20' to X'29'. The PUs in Figure 4 are associated to the VTAM definitions as shown in Figure 5.

It is important to note that you may choose to connect only one PU on a nonswitched line. However, this single PU still requires an SDLC station address, because the nonswitched environment requires the VTAM ADDR= field to be defined.

The alternate method for remote attachment is a switched connection, as shown in Figure 6.

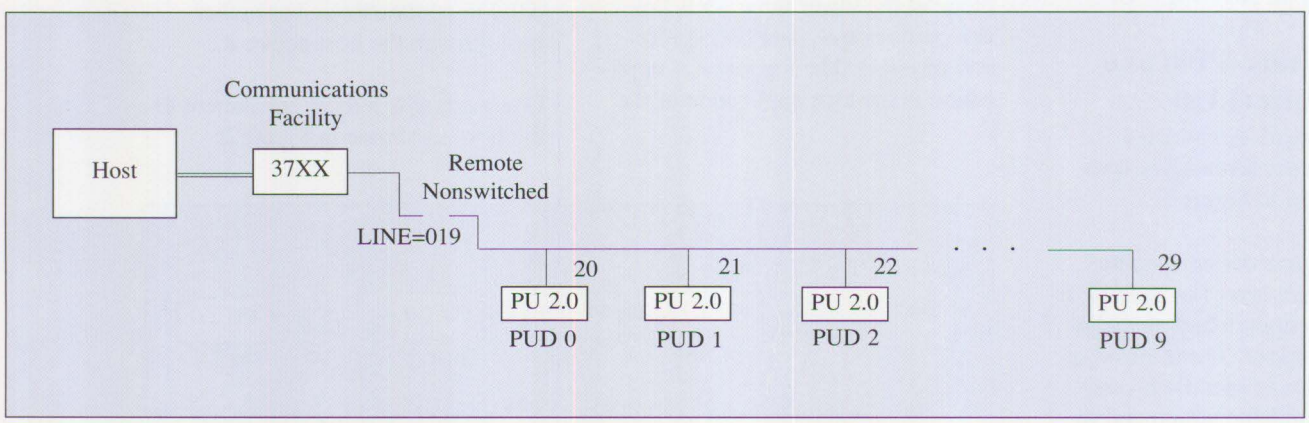


Figure 4. Remote, Dedicated Connection

On a switched connection, only one PU 2.0 is on a line. Multiple, switched PUs require an equal number of lines. Because only one PU is on a switched line, no SDLC addressing scheme is needed to uniquely identify the PU on a line. The host communications facility may have multiple switched lines, and it would be impractical to limit a specific PU to a specific switched connection only. Consequently, a pool of definitions, called a VTAM Switched Major Node, is defined. A switched PU's specific definition in this pool is determined by the program identifier and the Physical Unit Identifier (PUID) of the specific PU. In the VTAM Switched Major Node, each PU definition includes an IDBLK= field and an IDNUM= field. When a PU and the host are initiating contact, the PU's program identifier must match a PU definition IDBLK= field, and the PU's PUID must match the same PU definition IDNUM= field. Unless both matches occur, the contact will fail. Therefore, the PUs are associated to the VTAM Switched Major Node definitions, as shown in Figure 7.

Diagram	VTAM Definitions		
	DEFLIN01	LINE	ADDRESS=019
			:
PUD0 ---------->	DEFPUD0	PU	ADDR=20
			:
PUD1 ---------->	DEFPUD1	PU	ADDR=21
			:
PUD9 ---------->	DEFPUD9	PU	ADDR=29
			:

Figure 5. Remote, Nonswitched Connection Definition

3270 Emulators

We will concentrate on four 3270 emulators:

- PC 3270 Emulation Program Version 3 (3270EMV3)
- 3270 Workstation Program (WSP11)
- Personal Communications/3270 (PCOM3270)
- Operating System/2 Extended Edition Versions 1.1 and later Communications Manager's Part for 3270 Emulation (OS23270)

One other 3270 emulator is available and supported – PC 3270 Emulation Program Entry Level Version (3270EMEL). However, 3270EMEL does not access its host session through the LAN, and is discussed as a CUT-mode emulator only (see "CUT-Mode Emulation in the TRN Environment").

Also included are customization questions of the 3174 controller gateway (Z4-01 to Z4-08). These menus are accessed using a CUT-mode terminal (in TEST mode), coax-attached to the 3174 gateway while the gateway was still active and online. The menu sequence is the same as shown when the administrator views the configuration, which happens when the CUT terminal is in TEST mode. While this article does not contain details on how to customize the 3174 control-

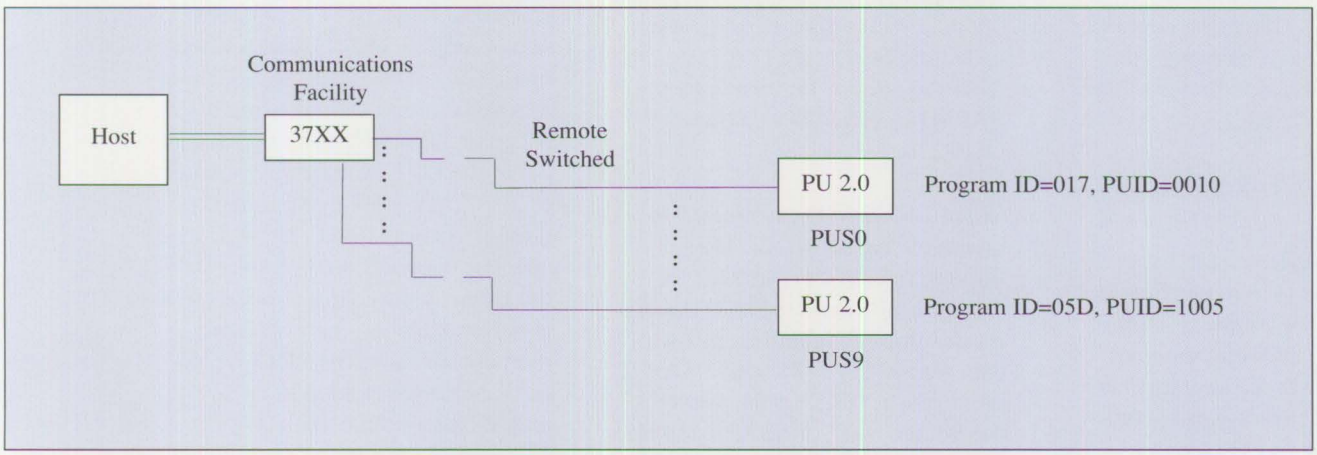


Figure 6. Remote, Switched Connection Definition

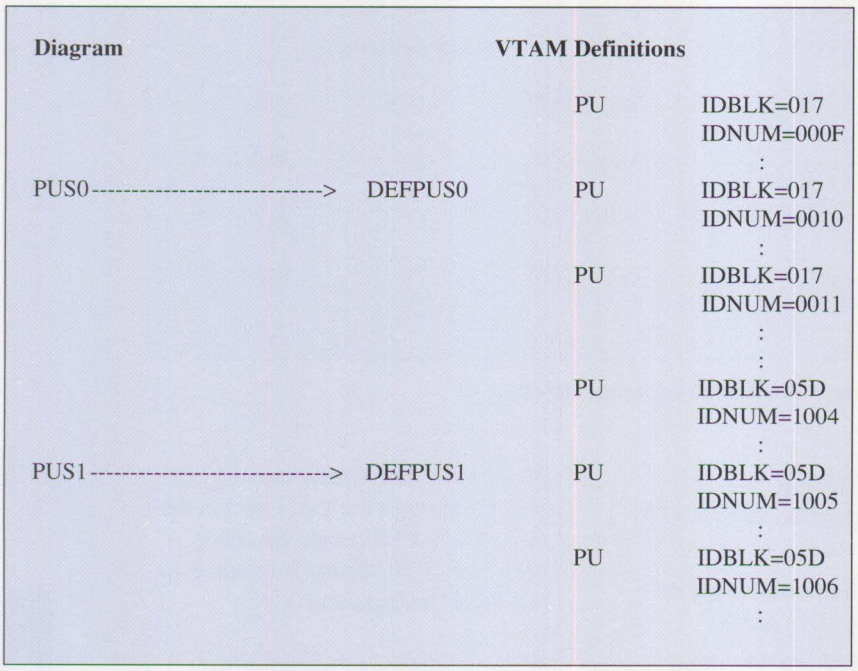


Figure 7. Remote, Switched Definitions

ler gateway, it does cover the relationship of the addressing scheme from the 3270 emulator up to the host. Consequently, the 3174 controller gateway's configuration questions are included to better illustrate how the emulator's definitions are related to those of the 3174 gateway, and subsequently how the controller gateway's definitions are related to the host's. Those interested in details of the 3174 customization, either as a controller gateway or one of its many other configurations, should refer to the 3174 planning guide.

SAPs in the Token Ring Network

In the LAN environment, many different types of communication facilities may use the common LAN media. In the TRN environment, these different types may include SNA components of various functions (PU type 5, PU type 4, PU type 2, and so forth), as well as a multitude of other facilities, such as NETBIOS, TCP/IP, and APPC.

Applications using these facilities communicate via the TRN through specific application programming interfaces, or APIs. Instead of requiring every API to process all incoming data, these interfaces (especially those that are coded to, or write to the IEEE 802.2 interface) may specify processing those incoming data with a specific indicator, called the Service Access Point (SAP) identifier. For example, the SAP ID of the 3174 is 04, as is the OS23270.

When OS23270 is customized to communicate across the TRN through the 3174 controller gateway, the API that OS23270 communicates through identifies its own data sent to the 3174 with a SAP ID of 04 (the SAP ID of the sender is the Source SAP, or SSAP), and also identifies the SAP ID of the 3174 (the SAP ID of the receiver is the Destination SAP, or DSAP). In such a configuration, the API that OS23270 uses processes only incoming data with a DSAP ID of 04.

For more information on SAPs, interfaces, token ring network protocols, and general TRN definitions and architecture, refer to the *IBM Token-Ring Network Architecture Reference*, SC30-3374.

LAN Connectivity in 3270 Emulation in the Non-SNA Environment

All TRN controller gateways currently implemented require an SNA environment. Consequently, this article has concentrated only on the SNA environment. Though controller gateways are not implemented in the non-SNA environment, it is possible to implement a PC gateway using 3270EMV3. The configuration would be attached as shown in Fig 8A).

To both the 3X74 and the non-SNA host, there is no distinguishable difference between the PC gateway and a DFT terminal that supports up to five host sessions.

It seems conceivable to have a configuration, such as in Figure 8, where the local 3X74 is replaced with a remote 3X74. However, the non-SNA environment implements remote communications using binary synchronous (BSC) line discipline; the DFT PC gateway feature of 3270EMV3 is not supported through a remote BSC 3X74. Consequently, the configuration of a DFT PC gateway through a remote 3X74 is not possible in a non-SNA environment. Because the SDLC protocol is not implemented in a non-SNA environment, a 3270EMV3 configured as an SDLC PC gateway is also not possible.

In the SNA environment, the DFT PC gateway is supported through both a local 3X74 and a remote 3X74. The relationships of the addressing scheme of the DFT PC gateway through the 3X74 to the host definitions are shown in Figure 9.

The Network Station Names (NSNs) are positionally related to the port sessions such that NSN 02 is related to the primary session, NSN 03 is related to the first secondary session, and so on. The values filled into the port session are the local addresses of the 3X74, and these local addresses are defined at the host as the LOCADDR= field.

Remember that the DFT PC gateway appears to both the 3X74 and the host as a DFT terminal. Consequently, the DFT PC gateway is not

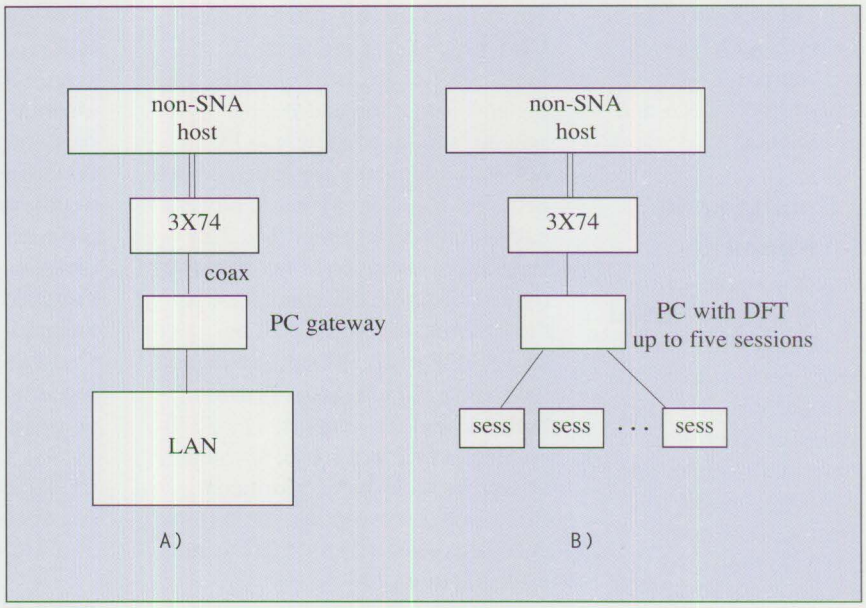


Figure 8. Non-SNA Gateway

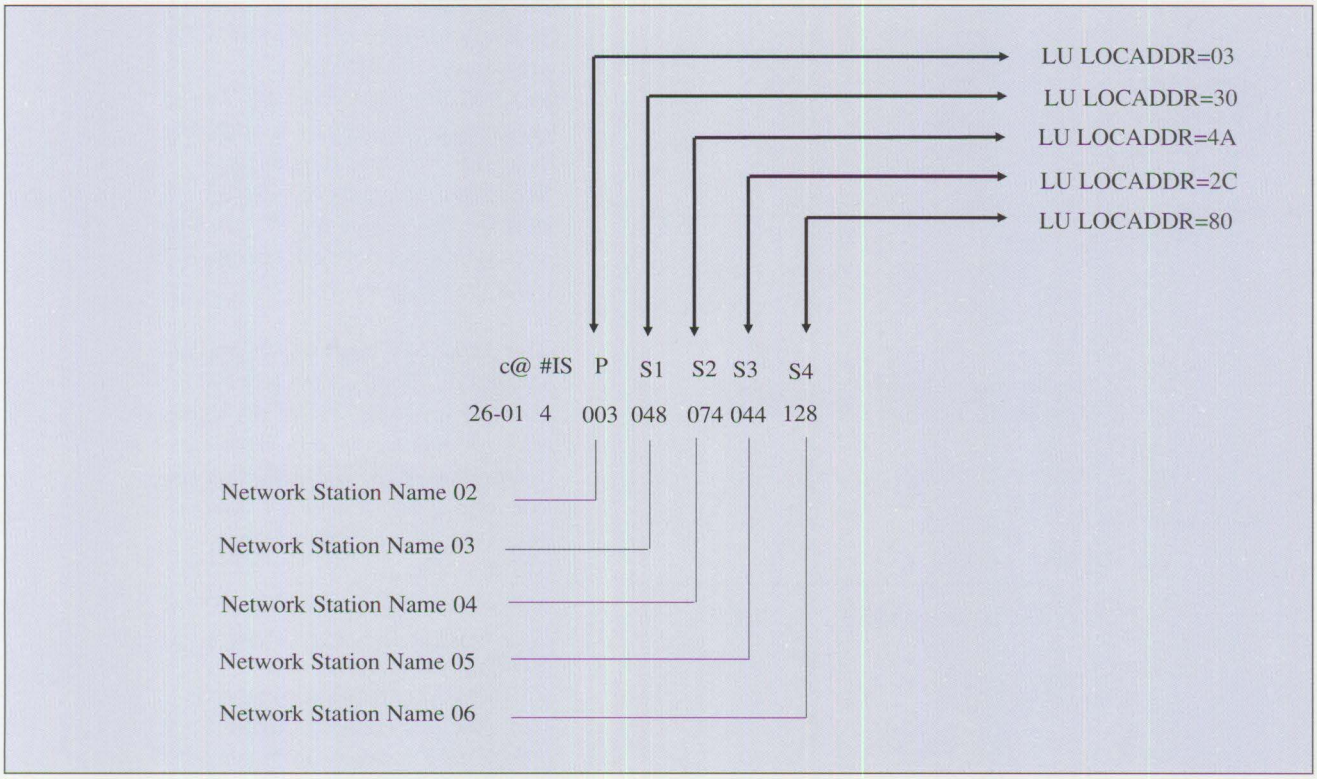


Figure 9. Network Station Name to Local Address Association

defined as a PU, but as a set of LUs under the PU of the 3X74. The NSN-to-LU associations, as shown in Figure 9, apply to both SNA and non-SNA environments.

CUT-Mode Emulation in the TRN Environment

In its original inception, the terminal was attached to a controller, and the keystroke and screen handling of the terminal are performed by the controller. All of the terminal's keystrokes are sent to the controller, and all of the terminal's screen updates are done by the controller. The terminal operates in Control Unit Terminal (CUT) mode. As technology advanced, a different implementation allowed the terminals to handle their own keystrokes as well as screen updates. These terminals operate in Distributed Function Terminal (DFT) mode.

The distinction between DFT and CUT implies that the terminal and the controller are two separate components. From the addressing scheme's point of view, the PU is not only a logically separate component, but also a component, physically separate from the LUs. This separation, however, is not necessarily true for the 3270 emulators. In the example cases that will be shown in Part II, all configurations have the 3270 emulators providing both LU and PU support. The 3270 emulators are, in a sense, terminals as well as their own controllers; from such a perspective, the difference between DFT and CUT is not as distinctive.

However, PCs running 3270 emulation are not the only LAN devices capable of accessing their host sessions via the TRN through a controller gateway. Some models of the 3174 (for example, models 53R and 03R) may be configured as downstream PUs communicating through a controller gateway. For the terminals coax-attached to the TRN downstream 3174s, there is no distinguishable difference among the different ways that the 3174 communicates upstream to the host (SDLC switched or nonswitched, channel bus-and-tag, X.25, and so forth). Consequently, both CUT- and DFT-mode terminals may be coax-attached to these TRN downstream 3174s; a PC running 3270EMEL, a CUT-mode 3270 emulator, may be coax-attached to a 3174 configured as a TRN downstream PU.

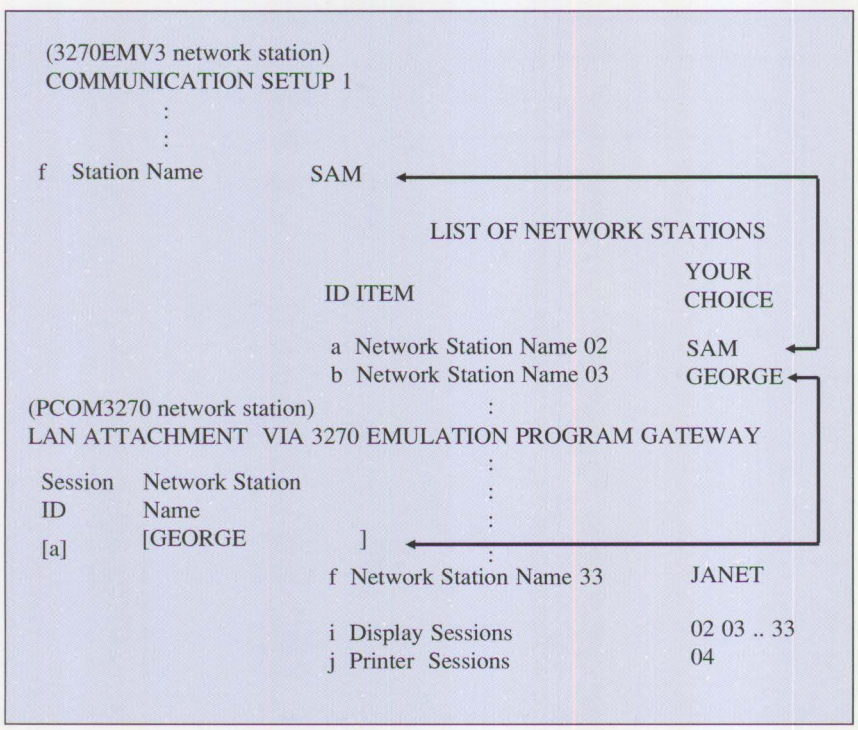


Figure 10. 3270EMV3 Name Match

3270 Emulator Gateway Functions

With the advent of the gateway features provided by the 3270 emulator, users need to be aware of another set of parameters through the PC gateway. There are three PC gateway features provided by 3270 emulators: 3270EMV3, PCOM3270, and OS2EE12. The concept of the gateway is relatively straightforward, as long as users keep separate points of view on how the host views the PC gateway versus how the network stations view the PC gateway.

The 3270EMV3 gateway is implemented differently from other PC gateway products. 3270EMV3 gateway communicates to its network stations using the NETBIOS protocol. Two types of network stations can communicate through the 3270EMV3 gateway: PCOM3270 network stations specifically configured to communicate through a 3270EMV3 gateway, and 3270EMV3 network stations. Network stations going through a 3270EMV3 gateway are matched up to LU definitions through the net-

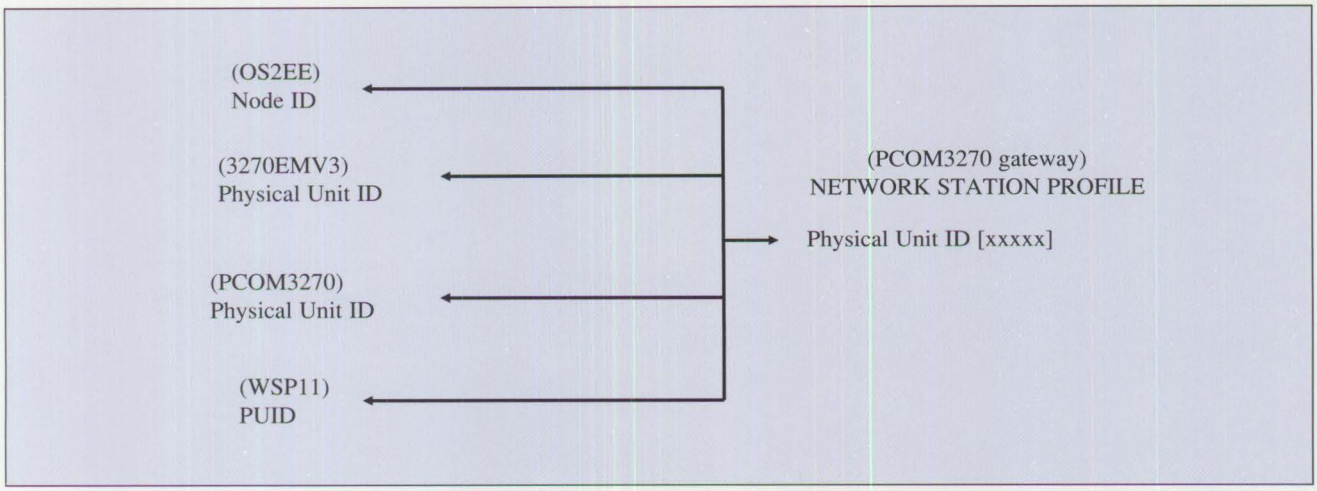


Figure 11. PUID Match

work station names, as shown in Figure 10.

Unlike the 3270EMV3 gateway, the PCOM3270 and OS2EE12 gateways communicate to their network stations using the 802.2 protocol instead of the NETBIOS protocol. The 802.2 protocol is also used by 3270 emulators communicating to controller gateways over the TRN. Consequently, a PC gateway that communicates to its network station via 802.2 is more versatile, as such a gateway can theoretically support other 3270 emulators that have previously been communicating directly to the TRN controller gateway.

The PCOM3270 gateway can define network stations in two ways: as explicitly defined or as a default. When a network station is contacting the PC3270 gateway, the PCOM3270 gateway first looks for a match in its list of explicitly defined network stations. If no match is found and if the PCOM3270 gateway has been configured with a default network station, then the calling network station uses the default network station profile. Explicitly defined network stations are defined using the PUID, as shown in Figure 11.

The LUs defined under each of the network stations are either preallocated or pooled. The preallocated LUs are referenced with their specific local address, while the pooled LUs are defined to a pool and are referenced according to the specific pool to which they are defined. The PCOM3270 gateway as defined to VTAM is shown in Figure 12.

The PCOM3270 gateway is defined to VTAM as a single PU with many LUs. However, from the network station's point of view, the PCOM3270 gateway appears similar to an NCP TRN controller gateway, because, from the network station's perspective, the PCOM3270 gateway supports multiple PUs. Each PU (if the PCOM3270 gateway is configured for explicitly defined network stations) identifies itself to the PCOM3270 gateway using the PUID. For explicitly defined network stations, the PCOM3270 gateway definitions allow users to optionally mandate a specific TRN address for a specified PUID. How these two perspectives (one as a single PU and the other as a NCP TRN gateway) are resolved is through the PCOM3270 gateway definitions; these definitions match the LUs defined for the PCOM3270 gateway at VTAM to the LUs defined for the network stations.

The OS2EE12 gateway is similar in many ways to the PCOM3270 gateway, though there are differences in some implementations as well as terminology. The main differences are as follows:

1. Unlike for the PCOM3270 gateway, network stations are defined to the OS2EE12 gateway through the network stations' TRN addresses. (Remember that network stations are defined to the PCOM3270 gateway through the network station's PUID.) The PCOM3270 gateway may optionally be configured to require a TRN address match for any explicitly defined network station.

2. Though the PUID of the network station is not used and therefore irrelevant during the process of the

(PCOM3270 gateway)
HOST PROFILE FOR GATEWAY

Press F10 to scroll through options
Press PgDn to see more

Host Name HOST1

				PUG17	PU	
Gateway LU address	**LU type**	**Pool ID**				: :
02	[Local]	[01]	<--------->	LUG1702	LU	LOCADDR=02
03	[Local]	[01]	<--------->	LUG1703	LU	LOCADDR=03
04	[Preallocated]	[01]	<--------->	LUG1704	LU	LOCADDR=04
05	[Preallocated]	[01]	<--------->	LUG1705	LU	LOCADDR=05
06	[Pooled]	[01]	:			:
07	[Pooled]	[01]	:			:
08	[Pooled]	[01]	:			:
09	[Pooled]	[01]	:			:
0A	[Pooled]	[01]	:			:
0B	[Pooled]	[01]	<--------->	LUG170B	LU	LOCADDR=11
0B	[Pooled]	[01]	<--------->	LUG170B	LU	LOCADDR=11

Figure 12. PCOM3270 Gateway to VTAM

network station contacting the OS2EE12 gateway, it is used when the network station is sending alerts (Network Management Vector Transport, or NMVTs) to the host. NMVTs not only allow monitoring, but also control nodes throughout a SNA network. Sending NMVTs is part of the configuration in OS/2 EE, versions 1.1 and 1.2.

3. LUs with the same characteristics may be grouped together as a pool in a PC gateway. A specific pool of LUs defined under the PCOM3270 gateway is referenced with a specific pool ID; such a definition allows network stations accessing the PCOM3270 gateway to request a certain number of sessions from a pool ID. Subsequently, the PCOM3270 gateway can dynamically allocate available sessions on request. The same principle applies to the OS2EE12 gateway, though the terminology changed. A pool defined under the PCOM3270 gateway is designated with a pool ID; under the OS2EE12 gateway, a specific pool is designated with a pool class.

4. LUs defined under a PCOM3270 gateway may also be designated as preallocated. LUs designated as preallocated are explicitly defined to a network station via a specific local address. Generally, such LUs are defined for network stations requiring unique LU definitions different from other network stations. An example of such a requirement may be access to administrative functions based on a session's local address (the LOCADDR= field under an LU definition in VTAM). The same principle applies to the OS2EE12 gateway, though under this gateway, these LUs are designated as dedicated.

5. Unlike the PCOM3270 gateway, the OS2EE12 gateway does not provide a default network station profile. Each network station must be explicitly defined to the OS2EE12 gateway.

The definition of the various LU pools in the OS2EE12 gateway will be shown in Part II of this article.

Frame Sizes and Segmentation

The topic of this article covers matching different addresses from

the various communication components. However, there are other considerations for successful 3270 emulation. Matching frame sizes for the various communication components is one of the more important considerations.

Before we get into this topic in detail, we need to introduce the following terminology:

- RH - Request/Response Header
- RU - Request/Response Unit
- BIU - Basic Information Unit = RH + RU
- TH - Transmission Header
- PIU - Path Information Unit = TH + RH + RU
- BTU - Basic transmission unit, one or more PIUs

The RH and RU are the basic units of information that send requests and replies across the SNA network. The transmission header defines specific formats, and these different formats define how the requests and replies should be routed through the SNA network.

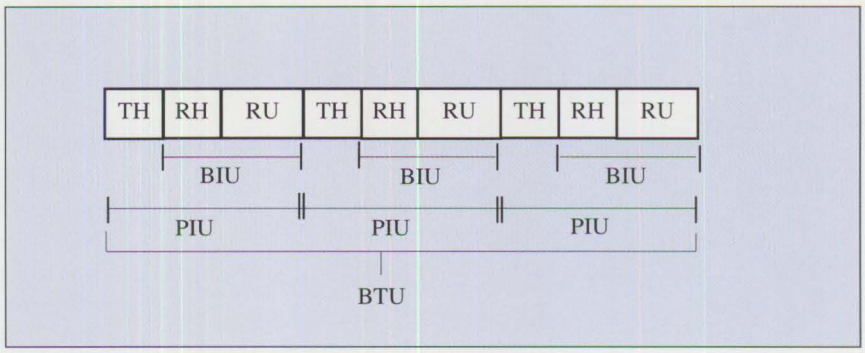


Figure 13. Format of BTU Blocking

A BIU does not always fit neatly into a BTU. Sometimes a BIU is smaller than a BTU, and sometimes larger. When BIUs are smaller than a BTU, and requests and replies are sent between two System/370 hosts, the multiple BIUs are combined into single BTUs for more efficient use of the communications facilities. The process of combining multiple BIUs into single BTUs is called blocking, as depicted in Figure 13. Blocking can be more accurately described as combining multiple PIUs into single BTUs. Because blocking does not occur except for communications between hosts, we can imply that the terms BTU and PIU are equivalent for the communications scenarios covered. The PIUs (also known as BTUs) are sometimes called frames. In the case of the 3174, the PIUs are also called Information-frames, or I-frames.

When the BIU is larger than a PIU, the communications facility divides

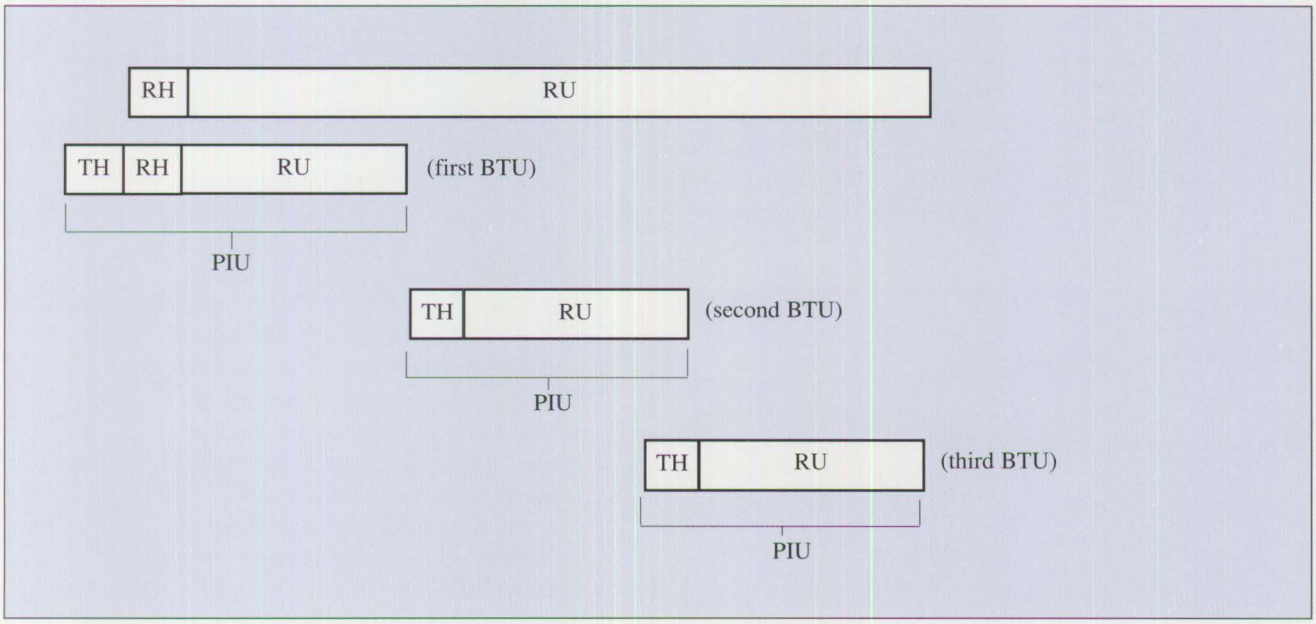


Figure 14. Format of BTUs During Segmentation

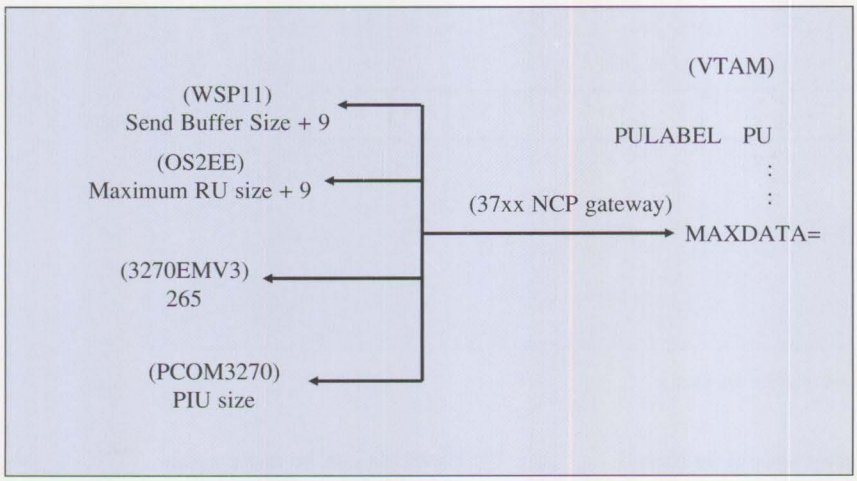


Figure 15. PIU Size Match

the BIU into smaller PIUs, as shown in Figure 14. This process is called segmentation. The process of segmentation is a common phenomenon in the 3270 environment. For more details about segmentation, refer to the *SNA Technical Overview*, SC30-3073.

After defining these fundamental terms, we can go through some basic configurations and identify which sizes need to match. When a PU 2.0 communicates over the TRN through an 37XX NCP gateway, the maximum PIU size defined at the PU 2.0 needs to match the MAXDATA= field of the PU macro in VTAM, as shown in Figure 15.

The parameter for setting OS2EE's PIU size requires some additional explanation. For 3270 emulation, the "Maximum RU size" field as defined under the SNA IBM Token-Ring Network data link control (DLC) profile of OS2EE, actually refers to PIU size less nine bytes. The 3270 emulation under OS2EE supports segmentation, while APPC under OS2EE does not. If segmentation is not supported, then the maximum RU size is the PIU size less the TH and less the RH, or PIU size minus nine bytes. Because 3270 emulation under OS2EE is implemented as part of the APPC subsystem, the PIU size under OS2EE is defined using terminologies that imply no segmentation support. Therefore, the PIU size under OS2EE is the "Maximum RU size" field plus nine bytes.

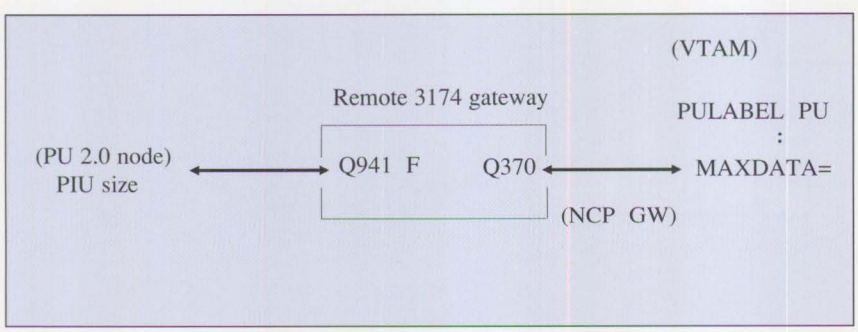


Figure 16. Same Sizes Through 3174

There is an additional limitation on the PIU size when communicating through a NCP TRN gateway. The NCP gateway allows whatever PIU size to go through it as long as this PIU size is smaller than the MAXTSL= parameter. The MAXTSL= parameter is defined under the LINE macro of NCP.

A PU 2.0 communicating through a remote 3174 gateway adds a new level of isolation for frame sizes. The frame sizes that need to match are shown in Figure 16.

Note that in the preceding configuration, the MAXDATA= field in VTAM does not need to match the PIU size of the PU 2.0. If there is a difference in frame sizes between the "F" field of Q940 and Q370, the 3174 gateway performs resegmentation of the PIU. However, it is recommended that the PIU size of the PU 2.0 node match Q370 of the 3174 gateway. While the 3174 will perform resegmentation of the PIU when necessary, it will not perform blocking even if there is a PIU size difference: the 3174 will forward a PIU, regardless of how small it is, if the PIU is smaller or equal to the smaller of Q941 F and Q370. The "F" field of Q941 uses a single-digit number to represent the different I-frame sizes: 0 for 265, 1 for 521, 2 for 1033, 3 for 2042, and 4 for 4105. Unlike Q941, Q370 is filled in with the actual I-frame size instead a single-digit representation of the I-frame size. The 3174 establishment controller planning guides have more information on this as well as other customization questions.

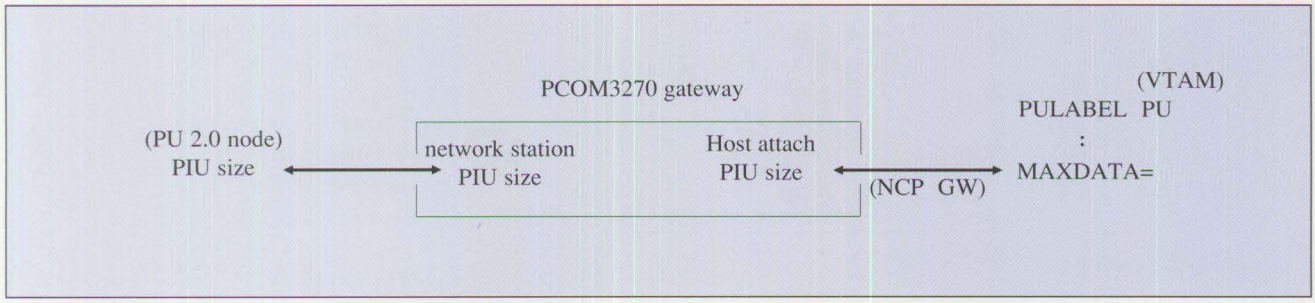


Figure 17. PIU Size Match Through PCOM3270

Note: The 3174 planning guides differ, depending on the microcode level used on the 3174: one for *Configuration Support A and S 5.0*, GA27-3844, and one for *Configuration Support B*, GA27-3862.

The PCOM3270 gateway functions also provide a level of isolation of frame sizes. The PIU size of network stations going through the PCOM3270 gateway does not match MAXDATA= in VTAM. Instead, the PIU size match is through the PCOM3270 gateway as shown in Figure 17.

Although it is recommended, the PIU size of the PU 2.0 node does not need to match the MAXDATA= field in VTAM.

If a PU 2.0 communicates through a local 3174 TRN gateway, then the PIU size matches are as shown in Figure 18.

Because either the OS2EE12 or the PCOM3270 gateway is viewed as a PU type 2.0 node from upstream, the PU 2.0 node can be replaced in the preceding figure with either of the PC gateways.

The important thing to remember is that PIU sizes should be matched in stages. For example, to have an OS2EE node communicate through a PCOM3270 gateway, the PCOM3270 gateway would communicate through a remote 3174 TRN gateway, the remote 3174 TRN gateway would communicate through an NCP Communications Controller, and the NCP Controller would be channel-attached to the host. The first matching stage is between the OS2EE node and PCOM3270 gateway, as shown in Figure 19.

The next matching stage is between the PCOM3270 gateway and the remote 3174 TRN gateway, as shown in Figure 20.

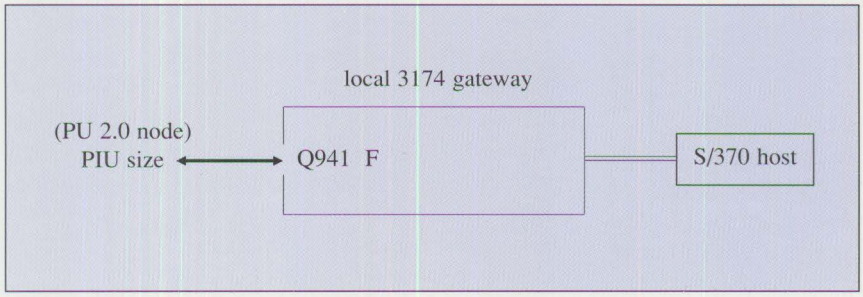


Figure 18. PIU Size Match Through Local 3174

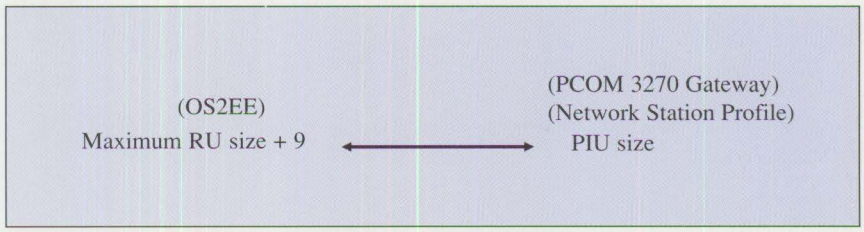


Figure 19. PIU Size Between OS2EE and PCOM3270

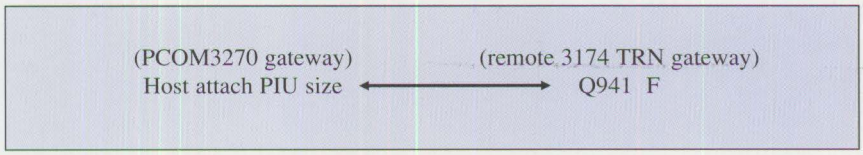


Figure 20. PIU Size Between PCOM3270 and 3174

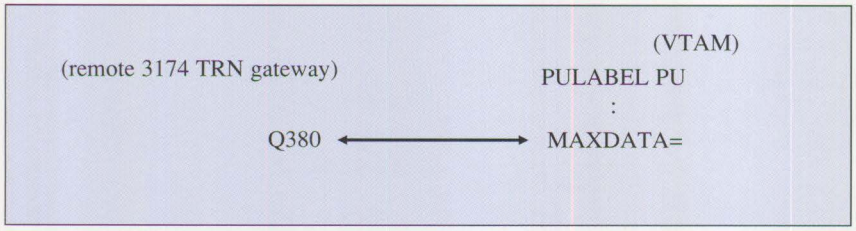


Figure 21. PIU Size Between 3174 and Host

The next matching stage is between the remote 3174 TRN gateway and the NCP Switched Major Node definitions in VTAM, as shown in Figure 21.

Matching Frame Sizes

The network administrator needs to be aware that a PU 2.0 that can successfully establish contact with the host does not guarantee that frame sizes are matched between the PU 2.0 node, the host, and any intermediate devices. Frame sizes are statically defined in the PU 2.0 and in the host: on the host, MAXDATA= dictates what the host knows is the largest frame its downstream device can receive and send; on the PU 2.0 node, the PIU size in all its different nomenclature dictates what the PU 2.0 node knows its upstream device can handle. For communications between the host and a PU 2.0 node, the two nodes do not negotiate or exchange information on the largest frame size that each other can send and receive. If there is a mismatch of PIU sizes at any one of the PIU size matching stages (as shown with the example with OS2EE communicating to the host via PCOM3270, remote 3174 TRN gateway, and NCP), the connection will work as long as PIUs transferred are smaller than or equal to the smaller of the PIU size in the mismatched stage. Once a PIU size exceeds the smaller of the PIU size in the mismatched stage, the connection will be dropped because of a frame reject. As previously noted, be aware that a PIU size that is different from one matching stage to another will work fine; however, a mismatch of PIU size in the same matching stage causes the connection to drop as soon as the smaller of the mismatched PIU size is exceeded.

Preface to Part II

We have now covered the necessary concepts for PC 3270 emulator-to-host communications. Specific configuration examples will be covered in Part II.

ABOUT THE AUTHOR

William J. Wen received a bachelor's degree in electrical engineering from the University of Houston – Central Campus. He worked for six co-op terms as a supplemental employee in the IBM National Technical Support Center in Dallas. Bill is currently employed as a systems engineer, working on system integration and performance analysis of PC systems.

Diskette Failures Caused by Contamination

John Holsteen
IBM Corporation
Boca Raton, Florida

This article shows the different kinds and sources of contamination affecting diskette operations, and methods that can be employed to reduce diskette failures.

Until recently, it was never fully understood why diskette drives fail to read environmentally contaminated diskettes. Failures result from drives becoming contaminated after cooling air is drawn through the front panel and expelled out the back. Part of the airflow impregnates the magnetic read heads within the flexible media drive.

Dust, dirt, and fiber particles pick up a minute electrical charge, which adheres to the recording head surfaces. Heads are exposed if no diskette is loaded into the drive; therefore, many particles settle and adhere to the surfaces of the lower and upper heads.

The head has two rails or contact sliders that come in contact with the recording surface upon inserting a diskette. The rails of the upper head are directly opposite the rails of the lower head. The impact of inserting a dusty diskette can easily damage heads or diskette surfaces.

Four ways that contaminating particles damage the media recording surface are:

- Tiny pock marks may occur at head impact point due to contaminating particles being lodged between the head and diskette surfaces during head load.

Light marks are often seen but usually do not constitute a read or write problem. Heavy damage marks may be so severe as to damage the surface enough to interfere with the recording process.

- The contaminant sticks to the head long enough to develop a surface scratch on or into the diskette magnetic coating. Usually, this results in a circumferential scratch deep enough or wide enough to adversely affect the recorded signal.
- The contaminating particle may bond itself to the media recording surface during head load. The head now bounces over the area and cannot read or write properly due to the head-to-diskette gap being too great. Occasionally, if the contaminant is soft, it may smear out or pull loose entirely after several revolutions of the diskette, resulting in recovery of normal recording operations. Thus, a failing diskette may reformat correctly. The wipe material liner within the diskette cartridge will pick up any loose debris.
- A particle that has bonded itself to the recording surface may strip the coating from the MYLAR® diskette base if the coating-to-MYLAR bond is weak. This exposes the MYLAR base and destroys the diskette. This condition is more often observed on poor-quality, less-expensive diskettes.

Diskette failure primarily results from:

1. Heads collecting dust.

Because the lower head is exposed upward, debris adheres to it. When the diskette is inserted into the drive, the same debris crushes against the diskette, possibly damaging the magnetic coating. It may result in merely a tiny pock mark. Occasionally, it results in severe magnetic-coating destruction such as a crush mark, tearout, scratch, or combination of all three.

In most cases, the position most often used for head load can be identified; that is, outside cylinders due to DIRECTORY or FAT table update prior to the last operation completion and head unload.

2. All magnetic media may have manufacturing imperfections, contamination problems, or both.

This is evident by small, rather isolated, coating protrusions observable under a microscope. These protrusions develop from one or more of the following: 1) surface defects on the diskette MYLAR base material, 2) contaminating particles sandwiched between the MYLAR surface and the magnetic coating, or 3) contamination within the magnetic coating material itself.

If these protrusions are present on a diskette with poor coating-to-MYLAR bonding, the head could tear loose a portion of the magnetic coating. Some are crushed by head impact, destroying a small area of the diskette recording surface. Others tear out and may strip the magnetic coating from the MYLAR base. In either case, the diskette is destroyed.

Low-quality diskettes are more likely to have this problem because they don't undergo the rigid testing required for high-quality diskettes. Low-quality diskettes are also susceptible to poor adhesion of the coating to the MYLAR. Generally, these low-cost diskettes lack a manufacturing brand.

Testing Contaminated Diskettes

A six-week "customer simulation" test was conducted in a lab with a field-returned planar and one of the returned diskette drives. Both items were contaminated with an exorbitant amount of dust and dirt that was not removed for the test. Both the planar and the drive were used as shipped to the lab.

Ten new IBM 2 MB diskettes were selected, and the recording surfaces were observed under microscope. Four diskettes were used to run a daily backup program. The remaining six diskettes were formatted daily, except Sundays. This test simulated how systems are often used in the field.

During two weeks in the middle of the test, this procedure was not followed. However, during the entire six weeks, the test system was used daily for formatting other diskettes and for general lab use. The system was powered down approximately 50 percent of the time when not in direct use.

At the conclusion of the test, the diskettes were again inspected under the microscope.

Test Results

During the six-week test, there was absolutely no reason to suspect any problem with either the planar, the diskette drive, or any of the diskettes used in this system.

The 10 controlled test diskettes performed flawlessly. Re-examination did not disclose any notable changes in the recording surfaces except for very minute pock marks. Again, these marks were attributed to contamination collecting on the lower head, which caused very slight marks on the diskette surface during head load procedures. In most cases, it was impossible to identify the head load area even though it was monitored during the testing.

Conclusions

Field-returned diskette failures result from two factors:

- Large-scale use of media of questionable quality.
- Operation of the computer system in a less-than-desirable environment; that is, extremely dusty conditions.

Recommendations

1. Leave a blank diskette inserted into the diskette drive when the PC is not in use. This will reduce contamination from dust particles that collect on heads during inactive periods.

2. To reduce diskette failures, use only high-quality media.

3. Relocate systems operating in a dusty or dirty environment, or replace them with a system designed and rated for industrial use. Avoid placing system units near carpeting.

Diskettes should operate with no problems in a clean environment.

Summary

New diskette products can better withstand contamination. Improvements being introduced, such as soft head load and restricted airflow, increase drive/diskette contamination tolerance. All of these improvements result in better diskette/drive performance under the same adverse operating conditions.

ABOUT THE AUTHOR

John Holsteen is a senior laboratory specialist at IBM's Entry Systems Division in Boca Raton, Florida. Presently, he supports diskettes drives and controller development. John joined IBM in 1965 in San Jose as a final test technician for the 2321 Data Cell Drive, and in 1967 he joined the development laboratory to work on the first diskette drive. John has published two invention disclosures.

Little Solutions

Here are some tips and hints that may solve a problem for you. If you have helpful information to share, send it to us in care of the editor.

Using Microsoft Windows on 4019 Printer With Serial Sharing Device

Note: Before following these instructions, Microsoft Windows 3.0 must be properly installed.

To perform steps 1 through 7 of the following instructions, you will need an ASCII editor capable of editing the Windows 3.0 WIN.INI file. The Notepad or System Editor features that come with Windows 3.0 may be used. To invoke the Windows 3.0 System Editor, perform steps 1 through 5. If another editor is used, load the WIN.INI file, and skip to step 6.

1. From the Program Manager, select "File," then "Add;" from the Add dialog, enter the program description.
2. For the program name and location, enter the following:

```
C:\WINDOWS3\SYSTEM\SYSTEM.EXE
```

The drive and directory containing your Windows program are shown where "C:" and "\WINDOWS3" are.

3. Save this new program and return to the Program Manager.
4. From the Application Group, select the new System Editor icon.

Both .INI files as well as the the CONFIG.SYS and AUTOEXEC.BAT files will be loaded automatically.

5. Click on the "WIN.INI" window to bring it to the foreground.
6. Scroll down the page until you come to the LPT3 line.

   Go to the end of this line, and press Enter or click on your mouse twice.
7. At the beginning of the line just created, type LPT1.PRN:=

   Go to the upper left-hand corner of the menu and select:

   - FILE
   - SAVE
   - FILE again, to exit this screen

Continue, and completely exit Microsoft Windows.

8. The next screen is "Exit Windows."

   Choose OK, which saves this session.

You should now be back in DOS.

9. Create a file called AUTOEXEC.BAT (if you don't already have one).

   If you have DOS 4.0 or later, add the command CALL SSDI to this file.

   If you have an earlier version of DOS, add the command SSDI to the last line of the file.
10. Reboot the system by either holding down the Ctrl+Atl+Del keys, or turn the computer off, then on, and type WIN.

You now are back in Microsoft Windows.

11. Select the "Program Manager" icon if the "Program Manager" menu does not appear.
12. In this menu, select the "Main" icon, and then select the "Control Panel" icon.
13. When the "Control Panel" screen appears, choose the "Printers" icon.
14. Select the ADD PRINTERS > function.
15. Scroll down the menu until the "IBM LaserPrinter 4019" line appears.
16. Select this line, and choose the INSTALL function.
17. At the next menu, select the CONFIGURE function.

    The menu that appears has a small window entitled "Ports."

    Scroll down this menu and select the "LPT1.PRN" line.

    Select OK.

    Before exiting this screen, make sure there is an "X" in the box marked "Print Manager."

    Select OK to exit this screen.
18. On the serial sharing device, there should be a set of 10 numbered switches on the outside of the box.

    Turn on switch numbers 8 and 10.

    (Switch number 8 sets the timeout delay to 15.0 seconds, and switch number 10 disables the append form feed.)

Microsoft Windows is now properly configured and ready for use with the serial sharing device.
– IBM Development Laboratory, Lexington, Ky.

HPFS Performance Fix

Corrective Service Diskette (CSD) WRO4064 introduced a problem into HPFS. This problem resulted in a severe degradation of performance, and as a result, HPFS became considerably slower than FAT.

The problem has been fixed and is available with CSD 4098. Meanwhile, HPFS performance can be restored by replacing the following files by their previous versions, which may be found on the original distribution diskettes:

```
HPFS.IFS
UHPFS.DLL
SWAPPER.EXE
```

– Larry Pollis, IBM, Dallas

HPFS Performance Tips

The High Performance File System (HPFS) has been available since the release of OS/2 1.2. Unfortunately, there seems to be some mystery and confusion surrounding its use. Although the following information is not a complete study of HPFS, it contains information that should help you understand how HPFS helps improve performance.

Use HPFS on hard disks (or logical partitions) greater than 60 MB. HPFS is much more efficient than the FAT system when used with large hard drives. This is due to several factors. The prime directive of HPFS is to reduce file fragmentation, which is inherent with the design of the FAT system. This causes performance to degrade very quickly. HPFS controls this problem.

Second, HPFS uses BTrees for most of its look-up functions. The FAT system does not use use any type of ordering of data; it uses a sequential unordered search of its tables, which makes access extremely slow.

A third improvement is gained by keeping directory information close to the information to which it relates. FAT keeps the same information at the beginning of the media, which results in more work for the drive to find new information.

Finally, HPFS has an efficient, built-in cache that boosts performance.

LAZYWRITE is an integral part of HPFS and its cache, and should be left on. If data integrity is a problem, change the cache parameters to give greater protection from lost data in the case of a power outage or other calamitous failure. For example, the default for MAXAGE is five seconds. *All* dirty data must be written after it ages more than the MAXAGE setting. By reducing MAXAGE to a smaller value, the chances of having a catastrophic loss of data are greatly reduced.

Blocks of data greater than 32 KB bypass LAZYWRITE and are written directly to disk as well as the cache. Only blocks 32 KB and less are stored in the cache and are under the control of LAZYWRITE. Data is also written to the hard disk before it is bumped from the cache (in the case of LRU) regardless of any timing parameters.

Get into the habit of using SHUTDOWN or CLOSE ALL from the Desktop Manager's Desktop menu. This ensures that the cache and buffers are flushed, and the file system is closed in an orderly manner. However, as long as there are no open files, Ctrl-Alt-Del shuts down the file system and flushes the data, too. Since the dual boot causes a warm boot when invoked, issuing a BOOT /DOS also provides an orderly shutdown of the file system.

HPFS gives the system a much greater chance of recovery from a system crash than FAT does. HPFS can rebuild all of its directories and files, and can also reconstruct freespace bitmaps. This effectively protects most of the data on the hard disk from catastrophic loss. With the FAT system, a power outage (with open files or during writes to the disk) causes widespread destruction. It is impossible to fully recover from such a catastrophic failure. *– Larry Pollis, IBM, Dallas*

Prompting In REXX

When using the REXX SAY statement with PULL for prompting for and entering data, the data is entered on the line following the prompt. The PULL statement issues a **?** indicating that the program is waiting for some data to be input. The lines

```
SAY "Please enter your name - "
PULL NAME
```

appear as

```
Please enter your name - ?
```

at which time you type your name to the right of the question mark. It would look better if you could make the cursor wait at the end of the line without the question mark (because this isn't a question!). To do this, try the following:

```
CALL CHAROUT "Please enter your
   name - "
name = LINEIN()
```

Because the question mark is not issued by LINEIN, you may want to specifically include it, if in fact you do ask a question. *– Larry Pollis, IBM, Dallas*

New Products

Hardware

IBM PS/2 Model 95 XP 486 (8595-0J9, -0JD, and 0KD)

The Personal System/2® (PS/2) Model 95 XP 486 systems are a series of high-performance, highly expandable, floor-standing systems, based on IBM's Micro Channel architecture and 32-bit 80486 processors. With disk storage capability of up to 1.6 billion bytes, these systems are especially suited as LAN servers and multi-user hosts.

Through a unique system design, the 80486 processor is contained on a removable processor complex. This expandable processor feature allows processor upgrades from the 25 MHz to the more powerful 33 MHz system. The ability to upgrade can extend the life of the system as requirements for enhanced processing performance grow. The new XGA (Extended Graphics Array) Display Adapter/A with its high-performance, 32-bit bus master video subsystem is an standard feature of the IBM PS/2 Model 95 XP 486 family of machines. XGA is an evolution of the Video Graphics Array (VGA) providing the user with enhanced resolution, color content, and hardware functionality.

The IBM PS/2 Micro Channel Small Computer System Interface (SCSI) Adapter with Cache (32-bit bus master) is a standard feature that can provide enhanced DASD performance, especially when multiple DASD devices are installed. The system has eight 32-bit slots, and a total of seven storage device "bays." Up to five high-speed SCSI fixed disk drives, a variety of other storage drives, and tape backup devices. One Direct Memory Access (DMA) serial port and parallel port are standard. The PS/2 Model 95 XP 486 also provides a unique ability to boot from any drive and an easy upgrade to Basic Input System (BIOS).

The PS/2 Model 95 XP 486 Model 0KD features the 33 MHz 80486 microprocessor with 320 million bytes (320 MB) SCSI fixed disk storage. The models 0J9 and 0JD include the 25 MHz 80486 microprocessor with 320 million bytes (320 MB) SCSI fixed disk storage. The processor includes an internal memory cache controller, an internal 8 KB memory cache, and an internal floating-point processor unit.

The PS/2 2 MB Memory Module Kit-70 ns allows 2 MB of 70-nanosecond system board memory expansion on the PS/2 Model 95 XP 486 and the PS/2 Model 90 XP 486.

The IBM PS/2 4 MB Memory Module Kit-70 ns allows 4 MB of 70-nanosecond system board memory expansion on these models of the PS/2 Model 95 XP 486 and the PS/2 Model 90 XP 486.

The PS/2 256 KB Cache Option provides additional memory cache capability beyond the 8 KB internal memory cache. The PS/2 256 KB Cache Option is supported on these models of PS/2 Model 95 XP 486 and the PS/2 XP 486.

The PS/2 Model 95 XP 486 is supported by OS/2 Standard Edition 1.2 and 1.3, OS/2 Extended Edition 1.2 and 1.3, and DOS Versions 3.3 and 4.0.

Highlights:

- New processor complex featuring the 80486 25 MHz or 33 MHz microprocessor
- 4 MB standard parity memory, expandable to 32 MB on the system board
- PS/2 SCSI 32-bit bus master adapter with cache
- Up to 1.6 GB of internal high-speed data storage
- Seven internal storage device "bays," supporting a combination of 3.5-inch half-high drives and 5.25-inch full-high drives
- Enhanced Performance XGA Display Adapter/A is standard, providing 1024 x 768 resolution
- One DMA Serial Port and one DMA Parallel Port
- Selectable boot and easy to upgrade licensed system programs

Letter #190-175, October 30, 1990

IBM PS/2 Model 90 XP 486 (8590-0J5, -0J9, and -0KD)

The PS/2 Model 90 XP 486 is a family of powerful and expandable desktop systems based on IBM's Micro Channel architecture and 32-bit 80486 processors. These new systems are designed to meet the needs of users in advanced computing environments requiring high performance with large amounts of memory, medium-to-large data storage capability, and high-resolution graphics.

Through a unique system design, the 80486 processor is contained on a removable processor complex. This expandable processor feature of the IBM PS/2 Model 90 XP 486 system allows processor upgrades from the 25 MHz to the more powerful 33 MHz system. The ability to upgrade can extend the life of the system as requirements for enhanced processor performance grow. The new Extended Graphics Array (XGA) high-performance, 32-bit bus master video subsystem is a standard feature of the PS/2 Model 90 XP 486 system. XGA is an evolution of the VGA providing the user with enhanced resolution, color content, and hardware functionality.

The IBM PS/2 Micro Channel SCSI Adapter with Cache (32-bit bus master) is a standard feature that can provide enhanced DASD performance and can attach up to seven Small Computer System Interface (SCSI) devices from a single adapter. Four DASD bays are provided, one of which can accommodate 5.25-inch half-high, and the other three supporting 3.5-inch DASD devices. Two bays are available for expansion for a fixed disk, CD-ROM, floppy disk drive, and tape backup. Three additional 32-bit Micro Channel expansion slots are available for a wide range of additional functions, such as connectivity capability, I/O device attachment, or co-processor adapters. Two Direct Memory Access (DMA) serial ports and one DMA parallel port are provided as stan-

dard. The PS/2 Model 90 XP 486 also provides the unique ability of booting from any drive and an easy upgradeable to Basic Input Output System (BIOS) capability.

The PS/2 Model 90 XP 486 Model 0KD features the 33 MHz 80486 microprocessor with 320 million bytes (320 MB) of fixed disk storage. The models 0J5 and 0J9 include the 25 MHz 80486 and 80-million bytes (80 MB) and 160-million bytes (160 MB) of fixed disk storage, respectively. The processor includes an internal memory cache controller, an internal 8 KB memory cache, and an internal floating-point processor unit.

IBM is enhancing its option product line with the IBM PS/2 5.25-inch Slim High Diskette Drive. This device is an internal 5.25-inch, 1.2 MB slim-high diskette drive with electrical button eject. It does not require an attachment card or expansion slot for installation, and is supported in PS/2 Model 90 XP 486 and PS/2 Model 95 XP 486.

The PS/2 Model 90 XP 486 is supported by OS/2 Standard Edition 1.2 and 1.3, OS/2 Extended Edition 1.2 and 1.3, and IBM DOS Versions 3.3 and 4.0.

Highlights:

- New processor complex featuring the 80486 25 MHz or 33 MHz microprocessor
- 4 MB standard parity memory, expandable to 32 MB on the system board
- Enhanced performance XGA graphics integrated on system board, providing 1024 x 768 resolution
- Up to 960 MB of internal high-speed data storage
- PS/2 SCSI 32-bit bus master adapter with cache
- Four, 32-bit Micro Channel expansion slots (one is used for the SCSI adapter)
- Selectable boot, and easy-to-upgrade licensed system programs

Letter #190-176, October 30, 1990.

IBM PS/2 Model 80 (8580-081, -161, and A16)

The IBM PS/2 Models 80-081, -161, and A16 are enhancements to the PS/2 Model 80 family of high-function, Micro Channel floor-standing systems. Featured are the IBM PS/2 80 MB SCSI Fixed Disk Drive and the IBM PS/2 160 MB SCSI Fixed Disk Drive. These new SCSI fixed disk drives provide enhanced price performance and storage expandability. The 80 MB version is standard in the Model 80-081. Models 80-161 and 80-A16 each contain the 160 MB version. The Intel 80386 microprocessor operating at 20 MHz and 2 MB of 80 ns memory is featured in the models 80-081 and -161. The Model 80-A16 contains the Intel 80386 microprocessor operating at 25 MHz with a 64 KB memory cache, and 4 MB of 80 ns memory.

In addition to having seven available adapter slots, the standard configuration supports up to five internal Direct Access Storage Devices (DASD) and removable media, such as diskette drives, fixed disk drives, tape backup, and CD-ROM. An optional feature supports up to six internal DASD devices.

The model 80 is supported by OS/2 Standard Edition 1.2 and 1.3, OS/2 Extended Edition 1.2 and 1.3, DOS Versions 3.3 and 4.0, IBM 4600 Operating System Version 2.0, and AIX™ PS/2 Version 1.2.

Highlights:

- SCSI fixed disk drive models
- 2 MB memory standard on system board (8580-081, -161) expandable to 4 MB
- PS/2 Micro Channel SCSI Adapters standard DASD controller
- 4 MB memory standard on system board (8580-A16) expandable to 8 MB
- Micro Channel Architecture with eight I/O slots
- Support for 5.25-inch internal DASD device

Letter #190-177, October 30, 1990.

IBM PS/2 Model 65 SX (8565-321)

The PS/2 Model 65 SX (8665-321) is a new Micro Channel architecture model based on the popular Intel 80386 SX processor. This floor-standing system features the technologically advanced 320 MB SCSI fixed disk drive and utilizes the 80386 SX processor running at a clock speed of 16 MHz.

Standard features include a 1.44 MB, 3.5-inch, half-high diskette drive VGA graphics, and 2 MB of memory on the system board. The fixed disk drive controller is the PS/2 Micro Channel SCSI Adapter. This bus master adapter provides additional expansion capability while providing an interface for the new 3.5-inch, half-high SCSI fixed disk drives. This model can be expanded to 8 MB of memory on the system board and supports up to 16 MB of system memory.

The 8565-321 offers the same configuration flexibility and expansion enhancements as the previously announced 65 SX Models. In addition to its seven available adapter slots, the standard configuration supports up to five internal Direct Access Storage Devices (DASD), such as diskette drives, fixed disk drives, tape backup, and CD-ROM. An optional feature supports of up to six internal DASD devices.

The Model 65 SX is supported by OS/2 Standard Edition 1.2 and 1.3, OS/2 Extended Edition 1.2 and 1.3, DOS Versions 3.3 and 4.0, IBM 4680 OS/2 Version 2.0, and AIX PS/2 Version 1.2.

Highlights:

- Intel 80386 SX 16 MHz Processor
- PS/2 Micro Channel SCSI Adapter as standard DASD controller
- SCSI 12.5 millisecond fixed disk drive models: 320 MB

- 2 MB memory standard on system board expandable to 8 MB
- Micro Channel Architecture with eight 16-bit I/O slots
- Support for 5.25-inch internal DASD device

Letter #190-180, October 30, 1990.

IBM PS/2 Model P75 486 (8573-161 and 8573-401)

The PS/2 Model P75 486 is a high-end addition to IBM's portable computer family. The Model P75 486 features the powerful i486™ processor, operating at 33 MHz and Micro Channel architecture. Also standard are:

- 8 MB memory expandable to 16 MB
- VGA 16 grayscale plasma display
- Choice of 160 MB or 400 MB disk drives
- Full-size PS/2 enhanced keyboard
- 3.5-inch, 1.44 MB diskette drive
- External Small Computer System Interface (SCSI) port, external storage device port, serial port, parallel port, and pointing device port
- High-resolution Extended Graphics Array (XGA) video port
- Four option card slots (two full-size 32-bit, two half-size 16-bit).

With its processing power, storage capacity, expandability, software support, and portability, the Model P75 486 is well-suited for mobile consultants, engineers, and other professionals. The Model P75 486 is ideal for those who want to improve their productivity and effectiveness by accessing their applications at home or in remote field locations. It can also be used as a network server or workstation for temporary offices at conventions, sporting events, and other temporary work locations.

The IBM 8573 Keyboard Extension Cable is an optional keyboard extension cable and travel case that is designed to enhance the usability of all 8573 models. This cable gives users flexibility for moving the keyboard and system unit to maximize comfort and convenience.

The IBM PS/2 Travel Case is constructed of molded plastic with easy rolling wheels and an integrated, telescopic handle for pulling. The interior is padded, and contains space for the P75 or P70, cables, and a mouse. It is designed to give users an easy, safe way to transport the system. It conforms to FAA luggage regulations, allowing it to be carried on board the aircraft and stored under the seat.

Highlights:

- Enables advanced engineering/ scientific and business solutions for applications that require compute-intensive processing, large data bases, connectivity, and expandability, and the mobility to transport these applications to the job site or temporary work location
- Protects a company's investment in applications and training by providing an AC portable (fully compatible with IBM desktop systems) with a familiar, full-size keyboard
- Allows application growth in complexity and data volumes

Letter #190-197, November 12, 1990.

IBM PS/2 Model 55 LS

The Model 55 LS enhances the PS/2 family of systems by offering the 32-bit 80386SX microprocessor in a local area network (LAN) station for token ring or Ethernet networks. The systems are adapted for connection to a LAN by the inclusion of either a 16/4 Token Ring Network Adapter/ A or an Adapter/A for Ethernet networks. Through the LAN, the system can be connected to selected PS/2 models, equipped with disk storage, which provide program and data storage and other services to the LAN. The Model 55 LS is provided without a diskette drive or fixed disk, but may be upgraded with the addition of 1.44 MB diskette drive or a 30 MB or 60 MB fixed disk, or both. The Model 55 LS is supported by DOS as a Client to either PC LAN Program or OS/2 LAN Server using Remote Program Load in a token ring network. Software support in an Ethernet network is provided by DOS as a Client to independent software vendor LAN operating systems using Remote Program Load.

Highlights:

- Compatible member of the PS/2 family with the 32-bit 16 MHz 80386 SX microprocessor
- 2 MB standard memory with up to 8 MB on the system board
- Ready for attachment to 4 Mbps or 16 Mbps token ring or Ethernet LAN
- Micro Channel architecture with two available 16-bit slots
- Optional 16 MHz 80387 SX co-processor for improved performance
- Optional 1.44 MB diskette drive and 30 MB or 60 MB fixed disk for growth

IBM PS/2 80 MB and 160 MB SCSI Fixed Disk Drive; IBM PS/2 SCSI Fixed Disk Drive Kit D

The PS/2 family of SCSI fixed disk drives is enhanced through introduction of two new high-performance, increased-capacity SCSI fixed disk drives. The PS/2 80 MB SCSI Fixed Disk Drive (80 million bytes) and 160 MB SCSI Fixed Disk Drive (160 million bytes) options feature faster average seek times and greater storage capacity than the 60 MB and 120 MB SCSI Fixed Disk Drive offerings. These new options, because of their SCSI interfaces and common mechanical design, are compatible with and across a wide range of IBM PS/2 systems, such as the PS/2 Models 60, 65, 80, 90, 95, and 3511-003.

The PS/2 SCSI Fixed Disk Drive Kit D supports the installations of either PS/2 60 MB Fixed Disk Drive or PS/2 120 MB SCSI Fixed Disk Drive in the PS/2 Models 90, 95, or 3511-003.

Highlights:

- By adhering to the industry-standard SCSI interface and a common mechanical design, an institution's investment is protected through a common set of PS/2 SCSI fixed disk drive options.
- User productivity is improved with high performance (17/16 ms average seek time) and high capacity (over 30 percent more storage capacity than the 60/120 MB drives) SCSI Fixed Disk Drive options

Letter #190-179, October 30, 1990.

IBM PS/2 1.44 MB 1-Inch Diskette Drive; IBM PS/2 320 MB SCSI Fixed Disk Drive; IBM PS/2 CD-ROM Drive

The IBM PS/2 1.44 MB 1-Inch High Diskette Drive, IBM PS/2 320 MB SCSI Fixed Disk Drive, and IBM PS/2 CD-ROM Drive have been enhanced to provide compatibility with and across a wide range of IBM PS/2 systems, including PS/2 Models 90 and 95.

The IBM PS/2 1.44 MB 1-inch High Diskette Drive supersedes the current PS/2 1.44 MB diskette drive. This option is supported in the same PS/2 models as its predecessor, as well as in the PS/2 Models 55 LS, 90, and 95.

The IBM PS/2 320 MB SCSI Fixed Disk Drive is functionally equivalent and supersedes the current PS/2 320 MB SCSI Fixed Disk Drive. This option is supported in the same PS/2 models as its predecessor, as well as in the PS/2 Models 90, 95, and 3511-003.

The IBM PS/2 CD-ROM Drive is functionally equivalent and supersedes the current PS/2 CD-ROM Drive. This option is supported in the same PS/2 models as its predecessor, as well as in the PS/2 Models 90, 95, and 3511-003.

These new options are compatible with a wide range of previously announced PS/2 systems, plus the 8590 and 8595, thus providing investment protection.

Letter #190-183, October 30, 1990.

IBM PS/2 400 MB SCSI Fixed Disk Drive

The IBM family of Small Computer System Interface (SCSI) fixed disk drives is enhanced with the addition of the technologically advanced PS/2 400 MB SCSI Fixed Disk Drive (400 million bytes). This high-capacity, 3.5-inch, half-high SCSI fixed disk drive, with its 11.5 ms average seek time and 128 KB look-ahead buffer, is the highest performance SCSI fixed disk drive available from IBM for PS/2 Micro Channel systems. This drive can be installed internally in PS/2 Models 60, 65 SX, 80, 90, 95, and 3511-003. When used in conjunction with the PS/2 External Storage Enclosure for SCSI Devices (3511-003), it can be attached to any PS/2 Micro Channel system.

Highlights:

- Recommended solution for LAN servers or technical workstations because it can quickly store and retrieve large amounts of data
- Adherence to the ANSI SCSI standard, plus the capability to install in PS/2 External Storage Enclosure for SCSI Devices (3511-003), which enhances growth enablement

Letter #190-196, November 12, 1990

IBM PS/2 2.3 GB Full-High SCSI Tape Drive

The family of PS/2 SCSI storage options is enhanced through the introduction of a high-performance, large-capacity SCSI tape drive. The PS/2 2.3 GB Full-High SCSI Tape Drive is an internal, 5.25-inch SCSI tape drive that features high performance. It accepts a large capacity data cartridge for those requiring a fast backup/restore option. This option is supported on PS/2 Models 95 and 3511-003, and includes a blank data cartridge and a cleaning cartridge. The drive supports 8 mm helical-scan camcorder data-type cartridges that adhere to the American National Standards Institute (ANSI) Helical-Scan Digital Computer 8 mm Tape Cartridge Format standard X3B5/89-136. The 2.3 GB SCSI Tape Drive, when combined with Sytos Plus/IBM OS/2 File Backup Utilities, provides a high-performance, large-capacity, easy-to-use tape backup/restore system.

Highlights:

- Protects data investment in the event of a catastrophic system failure when used in a backup/restore environment
- Provides growth enablement for PS/2 Models 95 and 3511 by adhering to the ANSI SCSI standard, which supports up to seven SCSI adapters

Letter #190-185, October 30, 1990.

IBM PS/2 External Storage Enclosure for SCSI Devices

The PS/2 External Storage Enclosure for Small Computer System Interface (SCSI) Devices Model 003 (3511-003) is a floor-standing expansion unit, similar in appearance to the PS/2 Model 95, and can attach to a PS/2 Micro Channel system with either a PS/2 Micro Channel SCSI Adapter or a PS/2 Micro Channel SCSI Adapter with Cache installed. The unit comes standard with a PS/2 320 MB SCSI Fixed Disk Drive (320 million bytes) and can be populated with a maximum of seven internal SCSI devices, such as fixed disk drives, CD-ROM drive, or a tape drive.

Highlights:

- Provides the capability to tailor system configurations to meet expanding application requirements
- Provides a high degree of hardware and software compatibility by adhering to the industry-standard SCSI interface

Letter #190-178, October 30, 1990.

IBM PS/2 1-8 MB 286 Memory Expansion Option

IBM enhances its family of PS/2 memory expansion products with the introduction of the IBM Personal System/2 1-8 MB 286 Memory Expansion Option. This adapter provides 1 MB of memory expansion with Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS) Version 4.0 support for PS/2 Models 50, 50Z, 55 SX, 60, and 65 SX.

The PS/2 1-8 MB 286 Memory Expansion Option is functionally equivalent to IBM's existing 2-8 MB 80286 Memory Expansion Option (8286, 6450609), but has 1 MB of standard 85 ns memory. It is expandable in increments of 1 MB or 2 MB up to a maximum of 8 MB by using existing Single In-Line Memory Modules (SIMMs), the 1 MB Memory Module Kit-85 ns, or the 2 MB Memory Module Kit-85 ns.

The PS/2 1-8 MB 286 Memory Expansion Option is functionally equivalent to IBM's existing 2-8 MB 80286 Memory Expansion Option, but with 1 MB of standard 85 ns memory. It is expandable with increments of 1 MB or 2 MB up to a maximum of 8 MB by using existing Single In-Line Memory Modules (SIMMs), the 1 MB Memory Module Kit-85 ns, or the 2 MB Memory Kit-85 ns.

The PS/2 1-8 MB 286 Memory Expansion Option can be used with OS/2 Standard Edition Versions 1.2 and 1.3, or OS/2 Extended Edition Versions 1.2 and 1.3 as extended memory, or it can be used as expanded memory. An expanded memory device driver compatible with LIM EMS Version 4.0 is included.

Highlights:

- EMS expanded memory support in a 1 MB adapter
- Growth by allowing expansion to 8 MB using existing 1 MB and 2 MB Memory Module Kits, or SIMMs
- Compatibility with the existing 2-8 MB 80286 Memory Expansion Option
- Easy-to-use interface provided for setup and driver installation

Letter #190-198, November 12, 1990.

IBM PS/2 XGA Display Adapter/A

The new PS/2 XGA Display Adapter/A is based on the Extended Graphics Array (XGA), which is a new video subsystem for the IBM PS/2 family of Micro Channel products.

The PS/2 XGA Display Adapter/A is a high-performance, 32-bit bus master. It was designed for compatibility with existing Video Graphics Array (VGA) software applications while offering many new advanced video features for the IBM PS/2 family of Micro Channel products. The video subsystem is an evolution of the VGA and provides the user with enhanced resolution, color content, and hardware function. The XGA design has been optimized for use by window managers and other Graphical User Interfaces.

The PS/2 XGA Display Adapter/A is standard on the PS/2 Model 95 XP 486 family of machines. It is also available as an optional display adapter for selected IBM PS/2 Micro Channel products.

XGA supports the 14-inch IBM PS/2 Color Display 8515 with its high-resolution 1024 x 768 capability as well as the 16-inch IBM PS/2 8514 Color Display. In addition, other PS/2 monochrome and color displays, including the 8503, 8507, 8512, 8513, and 8604, can be used depending on resolution and application requirements.

A PS/2 Video Memory Expansion Option can be used to upgrade the PS/2 XGA Display Adapter/A to 1 MB of memory for 1024 x 768 x 256 color support.

Highlights:

- Offers high-resolution 1024 x 768 for PS/2 Micro Channel 386 SX, 386, and 486 systems (excluding PS/2 Model P70 386)
- DOS XGA Adapter Interface provides 8514/A applications portability
- Integrates a 16-bit VGA and maintains compatibility with VGA applications
- Includes devices drivers for DOS, OS/2, Presentation Manager, and Microsoft Windows

Letter #198-182, October 30, 1990.

IBM PS/2 Image Adapter/A 1 MB, 3 MB, and 3 MB 6091

The PS/2 Image Adapter/A is a family of display adapters designed for image processing and general-purpose applications requiring high-resolution screen support. These adapters are supported on all PS/2 systems with Micro Channel architecture (except the PS/2 Model P70 386). Three new versions of the IBM PS/2 Image Adapter/A, which can be upgraded, supersede the existing IBM PS/2 Image Adapter/A and the PS/2 Extended Image Adapter/A, and provide an upgrade path for applications requiring additional memory for improved performance or an increased grayscale capability.

These adapters supports all existing PS/2 color and monochrome monitors including the 8506 monochrome portrait image monitor at resolutions up to 864 x 1200 with 256 levels of grayscale, the 8508 monochrome landscape image monitor at resolutions up to 1600 x 1200 with 16 levels of grayscale, and the IBM 6091-019 color monitor at resolutions of up to 1280 x 1024 with 256 colors. Additional resolutions are supported with varying levels of color and grayscale depending on the amount of memory installed on the adapter.

Memory upgrades are available for improved performance or increased grayscale/color capability. The IBM PS/2 Image Adapter/A 1 MB can be upgraded to 2 MB by adding the IBM PS/2 Image Adapter/A Memory Expansion Kit, and to 3 MB with the addition of two PS/2 Video Memory Expansion Kits.

A cable to attach the IBM 6091-019 monitor is provided with the IBM PS/2 Image Adapter/A 3 MB 6091. The cable can be ordered separately to upgrade the IBM PS/2 Image Adapter/A 3 MB to an IBM PS/2 Image Adapter/A 3 MB 6091.

Supplied with these adapters are drivers for OS/2 Presentation Manager, DOS Adapter Interface, Microsoft Windows/286 Version 2.1, AIX PS/2 Operating System Version 1.2, AIX PS/2 X-Windows Version 1.2, AIXwindows™ Environment for PS/2, and AutoCAD.

An enhanced performance OS/2 Presentation Manager driver and Microsoft Windows 3.0 driver will be available first quarter 1991.

Highlights:

- Supports high-speed compression and decompression algorithms that contribute to rapid response times for printing, scanning, or displaying image information
- Supports up to 256 colors from a palette of 16 million, and up to 256 grayscales, for a broad range of image and technical professional applications
- Provides compatibility with VGA and 8514/A modes
- Supports all IBM PS/2 displays and 8506, 8507, 8508, and 6091-019 displays
- Has high-performance processor and built-in hardware assist features
- Features 16-bit data transfer path to PS/2 Micro Channel bus

Letter # 190-108, August 14, 1990.

IBM PS/2 FaxConcentrator™ Adapter/A

The PS/2 FaxConcentrator Adapter/A is a high-function, microprocessor-based fax adapter that can provide voice-prompting, telephone tone detection, and optical-mark recognition functions when used with a suitable application program.

Features of the PS/2 FaxConcentrator Adapter/A allow realtime processing of tasks such as compression, decompression, and format conversions within the adapter, as opposed to doing these tasks with software in the PS/2 main processor.

The PS/2 FaxConcentrator Adapter/A, when used with a suitable application program, allows the PS/2 Micro Channel architecture family of computers to communicate with Group 3 facsimile devices over the public-switched telephone network for the purpose of routing fax data.

The PS/2 FaxConcentrator Adapter/A supports the IBM Facsimile Support/400 program offering for the PS/2, although it can be used with suitably developed programs for other PS/2 applications.

Highlights:

- New business-solution potential exists with applications in combination with the PS/2 FaxConcentrator Adapter/A to capitalize on the impressive growth of the facsimile machine market
- With the ability to install up to eight PS/2 FaxConcentrator Adapter/A's in one system, institutions can plan for growth
- Investments in Group 3 facsimile products will be protected because the PS/2 FaxConcentrator Adapter/A adheres to the CCITT Group 3 standards
- The Applications Programming Interface (API), as documented in the PS/2 FaxConcentrator Adapter/A Extended Device Driver Reference, has the documentation needed to develop applications that use the advanced functions of the PS/2 FaxConcentrator Adapter/A
- The PS/2 FaxConcentrator Adapter/A supports Error Correction Mode (ECM), a Group 3 extension feature, for more reliable communication with fax machines, which also support ECM.

Letter #190-118, August 21, 1990.

IBM PS/2 486/33 Processor Upgrade Option

The PS/2 486/33 Processor Upgrade Option from 486/25 enhances the PS/2 Models 90 XP 486 and 95 XP 486 family of systems. The processor upgrade option features the 33 MHz 80486 32-bit microprocessor and provides the capability of processor performance upgrades to the PS/2 Model 90 XP 486 and 95 XP 486 systems. The IBM PS/2 486/33 Processor Upgrade Option is an optional processor upgrade complex for the 25 MHz Model 90 XP 486 and Model 95 XP 486. This option significantly increases system processor performance in compute-intensive applications from the existing 25 MHz 80486 microprocessor to the 33 MHz 80486 microprocessor

The IBM PS/2 486/33 Processor Upgrade Option is supported by OS/2 Standard Edition 1.2 and 1.3, OS/2 Extended Editions 1.2 and 1.3, IBM DOS Versions 3.3 and 4.00.

Highlights:

- Internal memory cache controller and 8 KB internal memory cache
- Internal floating-point processor unit
- Allows processor upgrade in PS/2 90 XP 486 and 95 XP 486 systems
- Well-suited for compute- and numeric-intensive applications and for heavy multi-user, multitasking applications

Letter #190-186, October 30, 1990.

IBM 8230 Token-Ring Network Controlled Access Unit

The IBM 8230 Token-Ring Network Controlled Access Unit is an intelligent access concentrator that allows connection of from 0 to 80 workstations, via

pluggable lobe attachment modules (LAMs), to an IBM Token-Ring Network. The LAMs come in two versions; one accepts IBM Cabling System (ICS) connectors, and the other accepts RJ-45 connectors. The 8230 is switchable between 16 and 4 megabytes per second (Mbps) and has a media access control (MAC) appearance on both the main and backup rings. It is shipped with copper main ring modules and can be upgraded to fiber by plugging in the Optical Fiber Converter Module feature. The 8230 functions as a repeater in both directions. It is ideal for those desiring higher ring availability via automatic reconfiguration, and better access control and asset management. The 8230, in most cases, will also allow for lobe lengths longer than the 8228.

Highlights:
System management is improved by:

- Automatic wrap/reconfiguration of a failing ring segment, a failing lobe, a LAM of the 8230, or the entire 8230
- LAN Network Manager directed wrap reconfiguration of a failing ring segment
- Support of the LAN Network Manager's configuration table
- Support of the LAN Network Manager's access and asset control
- User interface light-emitting diodes (LEDs) and two-character hexadecimal error code display, all of which provide power status, wrap state, 8230 status, and insertion state of each LAM

User productivity is enhanced by:

- The network being available more often because of reconfiguration
- LAN Network Manager's configuration table allowing better use of resources
- The power supply meets worldwide requirements

Letter #190-144, September 5, 1990.

IBM 8209 Token-Ring Attachment Module

The IBM 8209 Token-Ring Attachment Module connects two local IBM Token-Ring networks into one logical ring. The connected rings can be any combination of 4 Mbps or 16 Mbps token ring networks. The 8209 Token-Ring Attachment Module utilizes the IBM 8209 Local Area Network (LAN) Bridge Base Unit, thus providing this function in a plug-and-play unit with no keyboard or display. The 8209 Token-Ring Attachment Module provides the same network management and similar filtering capabilities as the local IBM Token-Ring Network Bridge Program Version 2.2.

Highlights:

- Filtering on both source and destination addresses
- Capability to write your own filters

Letter #190-143, September 5, 1990.

IBM 8209 Enhanced Ethernet Attachment Module

The IBM 8209 Local Area Network (LAN) Bridge with the Enhanced Ethernet Attachment Module interconnects an IBM Token-Ring Network and an Ethernet Version 2 or IEEE 802.3/ISO 802.3 LAN. The Enhanced Ethernet Attachment Module provides the same function as the Ethernet Attachment Module, plus it offers enhanced network management on the IBM Token-Ring portion of the network. Specifically, the Ethernet Attachment Module offers Ring Error Monitor, a Configuration Report Server, and Ethernet LLC support for the IBM LAN Manager and IBM LAN Network Manager. The Enhanced Ethernet Attachment Module also provides filtering on both source and destination addresses.

Highlights:

- Bridges 4 Mbps or 16 Mbps token ring networks to Ethernet Version 2 or IEEE 802.3/ISO 802.3 networks
- Provides Ring Error Monitor Management Server for reporting token ring MAC frame error conditions

Letter #190-143, September 5, 1990.

IBM PS/2 Adapter/A Ethernet Networks

IBM enhances its local area network (LAN) family of products by announcing the PS/2 Adapter/A for Ethernet networks.

This LAN adapter attaches attaches PS/2 Micro Channel bus systems to Ethernet networks.

The adapter supports Ethernet Version 2 and IEEE 802.3 (CSMA/CD) interfaces, as well as data transfers at 10 Mbps via 10BASE5 (thick cable), 10BASE2 (thin cable), or 10BASET (twisted pair) through an Attachment Unit Interface (AUI) connection and user-provided external transceiver.

Remote Program Load (RPL) is included standard for systems requiring this capability. Only the Banyan Vines™ and Novell NetWare operating systems specified in the Programming Requirements section currently support Remote Program Load. The feature automatically issues a request to a loading server device any time the workstation is reset, has no system disk available, or if the disk program boot has been disabled.

OS/2 Extended Edition, IBM DOS, and PS/2-related programming are supported.

Highlights:
The PS/2 Adapter/A for Ethernet networks provides:

- Additional business solutions for offering both Ethernet and token ring products to create intelligent LANs
- Growth by allowing those with Ethernet/802.3 to expand their networks. Support of multiple interfaces and LAN software facilitates growth and applications exploitation

- Investment protection, by supporting Ethernet Version 2 and 802.3 (CSMA/CD interfaces
- Systems Management – a user-friendly interface is provided for installation and setup, configuration and diagnostic capabilities, and maintainability

Letter #190-172, October 9, 1990.

Software

IBM OS/2 Extended Edition Version 1.3

OS/2 Extended Edition Version 1.3 provides increased performance in memory-constrained environments while reducing the minimum memory requirement from 3.5 MB to 3.0 MB.

OS/2 EE Version 1.3 provides enhanced font support.

ACDI calls can be redirected across a LAN to the LAN Asynchronous Connection Server Version 2.0, and provides line and modem pooling facilities for asynchronous connections.

The Database Manager DOS Database Requester supports Ethernet DIX Version 2.0 and IEEE 802.3.

OS/2 EE Version 1.3 supports these systems as described in the Specified Operating Environment section.

Letter #290-690, October 30, 1990

OS/2 Standard Edition Version 1.3; OS/2 Programming Tools; Information Versions 1.2 and 1.3

OS/2 Standard Edition (SE) Version 1.3 fulfills IBM's stated intent to deliver in OS/2 SE a 2 MB OS/2 entry system and further expand the range of OS/2-capable systems. OS/2 Programming Tools Information Versions 1.2 and 1.3 will help application developers exploit this new function.

IBM OS/2 SE Version 1.3 has been enhanced to provide increased performance in memory-constrained environments and reduced requirements for system memory and fixed disk space. Also provided are Procedures Language 2/REXX, enhanced font support, improved printer device drivers, and usability enhancements.

Letter #290-691, October 30, 1990

OS/2 Standard Edition Version 1.3 Upgrade Offering

For a limited time only, those who have acquired OS/2 Standard Edition Versions 1.1 or 1.2 may receive a no-charge upgrade to Version 1.3.

To obtain this upgrade, the IBM No-Charge Upgrade Form must be completed. IBM will honor all forms that are properly completed and postmarked on or after November 30, 1990, but no later than March 29, 1991.

IBM reserves the right to withdraw this offer at any time.

Letter #390-185, October 30, 1990

IBM OS/2 EE OS/2 Programming Tools and Information Upgrade Offering

For a limited time only, those who have acquired the following OS/2 licensed programs may receive a no-charge upgrade to OS/2 Version 1.3:

- OS/2 EE Versions 1.1 or 1.2
- OS/2 Programming Tools and Information Version 1.2

To obtain these upgrades, the IBM No-Charge Upgrade Form must be completed. IBM will honor all forms that are properly completed and postmarked on or after December 31, 1990, but no later than April 30, 1991.

IBM reserves the right to withdraw this offering at any time.

Letter #390-186, October 30, 1990

IBM OS/2 Local Area Network Server Version 1.3

OS/2 Local Area Network Server Version 1.3 uses the improved program load facility of OS/2 EE Version 1.3 to provide performance enhancements when loading many OS/2 applications from servers to requesters. These enhancements include DOS LAN Requester support for Windows 3.0 and Ethernet support for DOS LAN Requester through the LAN Support Program Version 1.2, which is included with the product.

Highlights:

- Utilizes OS/2 EE 1.3 improved performance when loading application over the LAN
- Windows 3.0 support for DOS LAN Requester
- The user is authorized, without additional payment to IBM, to make and use up to 128 copies each of IBM LAN Support Program Version 1.2 and DOS LAN Requester.
- DOS LAN Requester NET STOP command

Letter 290-692, October 30, 1990

IBM LAN Network Manager Versions 1.0 and 1.1; IBM LAN Network Manager Entry

The IBM LAN Network Manager licensed program is designed for those desiring to manage multi-segment IBM Token-Ring Networks, broadband and baseband IBM PC Networks, and the IBM 8209 LAN Bridge that interconnects a token ring segment and an Ethernet segment. Using IBM LAN Network Manager, the LAN may be managed either centrally from an enterprise NetView®, or locally using the operator interface at the LAN workstation.

IBM LAN Network Manager provides facilities for managing the LAN media and LAN adapters in the token ring network or PC network. It uses IBM SAA

OS/2 EE Presentation Manager for the operator interface, and the IBM OS/2 EE Structured Query Language (SQL)-based Database Manager for building the LAN configuration database, which provides an excellent platform for developing customized LAN-management functions. IBM LAN Network Manager enhances the currently available IBM LAN Manager using IBM LAN Station Manager and the IBM 8230 Token-Ring Network Controlled Access Unit (CAU).

IBM LAN Network Manager extends the NetView-LAN commands to access an expanded set of LAN functions from NetView Version 2 Release 2, allowing for development of an integrated enterprise LAN/Wide-Area Network (WAN) management solution. IBM LAN Network Manager also provides an optional graphics display support at the LAN for enhancing user productivity, and a command-line interface at the IBM LAN Network Manager console that may be used for local automation.

IBM LAN Network Manager functions will be released in two versions.

IBM LAN Network Manager Entry, working with NetView Version 2 Release 2, allows management of remote, single-segment IBM Token-Ring or IBM PC Network (broadband or baseband) LANs. It runs as a task in the multitasking OS/2 EE and communicates with NetView at the host using the OS/2 EE Communications Manager via an existing LAN SNA gateway. IBM LAN Network Manager Entry provides expanded command/response facilities with NetView, allowing LAN/WAN management to be integrated into a central management facility with NetView. No local operator interface is provided at the LAN workstation.

Current users of IBM LAN Manager programs are offered an upgrade to IBM LAN Network Manager Versions 1.0 or 1.1 for a charge. Current users of IBM LAN Manager Entry and the IBM PC 3270 Emulation LAN Management Program are offered an upgrade to IBM LAN Network Manager Entry for a charge.

Letter #290-551, September 9, 1990

IBM LAN to LAN Wide Area Network Program

The IBM to LAN Wide Area Network Program is a licensed program that interconnects remote local area networks (LANs) across a wide area network (WAN). Applications written to the IEEE 802.2 interface and using the IBM NETBIOS frame protocol will be able to communicate with remote LANs using the wide area network as a transport medium. The IBM LAN to LAN Wide Area Network Program provides the interface between the LAN and the WAN. Sessions will be established between IBM LAN to LAN Wide Area Network Programs to provide the communication path. This communication will be via the LU 6.2 facilities provided by OS/2 EE Communications Manager 1.2.

Letter #290-553, September 9, 1990

IBM LAN Station Manager

The IBM Local Area Network (LAN) Station Manager provides LAN configuration and environment data to the IBM LAN Network Manager Version 1.1 from DOS and OS/2 LAN stations attached to the IBM Token-Ring Network and the IBM PC Network. The LAN Station Manager, in conjunction with the IBM 8230 Token-Ring Network Controlled Access Unit, also provides improved asset control for token ring stations.

National language support includes double-byte character set (DBCS) enabling, PC National Language code page support and translated documentation, and machine-readable information as required by the individual countries. Data input by the user, however, is limited to ASN.1 printable strings as defined by ISO. This 74-character set includes: A...Z, a...z, 0...9, space, single quote, open and closed parentheses, plus, comma, minus, period, slash (/), colon, equal sign, and question mark.

Highlights:

- Provides detailed LAN station information to the LAN Network Manager that allows it to manage LANs more effectively
- Centralizes information so that physical assets can be better monitored and managed
- Investment protection is ensured with support of existing LAN stations and communication protocols based on international standards that facilitate interoperability between different vendor products that support the same standards

Letter #290-552, September 9, 1990

IBM LAN Asynchronous Connection Server Version 2.0

IBM Local Area Network (LAN) Asynchronous Connection Server Version 2.0 continues to perform the prior version functions and is enhanced for new functions. It will now operate with Transmission Control Protocol/Internet Protocol (TCP/IP) protocols (both as a Client and Server), and perform ASCII-to-3270 protocol conversion, providing Telnet connections to IBM TCP/IP hosts to allow 3270 datastream access. Operation with IBM LAN Support Program Version 1.2 using NETBIOS protocols on Ethernet is also now possible. The two interfaces it supports have been adapted for operation under Windows 3.0 . Network management enhancements, allowing more efficient control of the operation, and better problem determination features have also been added. A graphical-based configuration file generation utility will be made available.

Letter #290-550, September 9, 1990

IBM Storyboard™ Live! – Version 1.0

Storyboard Live! – Version 1.0 is a multimedia application that allows users to mix video, audio voiceover and other sound with animation, still photography, drawing, painting, music, and text to produce high-quality, on-screen presentations.

Based on IBM Storyboard Plus, Storyboard Live! brings several new capabilities to the DOS multimedia environment, including:

- Video and audio playback without special hardware adapters
- Design and edit animation sequences
- Create and edit storylines via a mouse without programming
- Create full-motion video sequences with any supported video motion adapters

Highlights:

- A powerful multimedia application that combines drawing, painting, text, video, music, and voice, to produce high-quality, animated, on-screen audiovisual presentations
- Supports PS/1™ computers that meet minimum storage and hard-disk requirements. Many additional printers, audio, visual, and other input/output devices are supported, allowing users to select the proper combinations necessary to enhance growth, yet remain within a company's budget
- Used to make sales/marketing presentations, develop product and service demonstrations, produce imaginative trade show exhibits. and create interactive tutorials
- Allows users to import data in file formats used by in many popular programs that may currently be included in their business software library
- Provides increased flexibility by offering greater integration with other software and hardware

Letter #290-715, November 6, 1990

Triumph!™ Workstation Manager Version 2 and Triumph/DES!™ Workstation Manager Version 2

The protection of enterprise information continues to be essential in maintaining an organization's competitive edge. Programmable workstations make up the fastest-growing element of the enterprise data processing environment. Triumph! Workstation Manager Version 2 and Triumph/DES! Workstation Manager Version 2 are designed to respond to users' security demands for a secured DOS workstation.

Both products provide the equivalent of C2 security functions for user identification and authentication, file discretionary access control, object reuse, and auditing. Also included are automatic file encryption, Common Signon, Central Site Administration, an audit reduction program, security API and user identification via tokens/token reader or linkage to the IBM 4754 Transaction Security System, IBM Personal Security Card, PIN, and signature-verification support.

Highlights:

- Secures user data on client workstation
- Central site administration, audit record management
- Workstation resource management and resource security
- Application Program Interface (API)
- Transparent security, common signon

Letter #290-715, November 6, 1990

Index to Past Issues of *IBM Personal Systems Technical Solutions*

Issue 4, 1990 (G325-5009)
First Look at the New IBM PS/1 Computer
Using the IBM 4019 LaserPrinter Effectively
Micro Channel Interface Chip Sets
Token Ring Bus Master LAN Adapters
Extension of Wiring Rules for 4-Mbit/s Token Ring Using UTP Lobes
SCSI and DISK.386.SYS
Operating System Platforms: A Business Perspective
Minimum OS/2 1.2 DASD Requirements
User Profile Management
Understanding OS/2 1.2 LAN Server Performance
PM: An Object-Oriented Approach
DOS 4.00 SHARE
A "C" Programming Model for DOS Device Drivers
An Electronic Bulletin Board for PC Users

Issue 3, 1990 (G325-5007)
DOS – A Look under the Hood to See How It Spins
Memory Management in a DOS Environment
FASTOPEN – The DOS Performance Enhancer
DOS 4.00 Compatibility Issues
'Out of Environment Space' Errors
A New LAN Requester for DOS Systems
Creating a Dialog Box Dynamically Using WinCreateDlg
An Alternative for the OS/2 START Command
CUA: A Consistent Interface

Issue 2, 1990 (G325-5006)
OS/2 End User Advantages
What's New in OS/2 Standard Edition Version 1.2?
An Application Developer's View of OS/2
Object-Oriented Programming with C and OS/2 PM – Is It Possible?
Design Goals and Implementation of the New High Performance File System
OS/2 EE 1.2 Database Manager – Remote Data Services
OS/2 EE Database Manager Precompiler API
UNION, INTERSECT, EXCEPT
Writing a Database Manager COBOL/2 Program
Database Manager Programming with Procedures Language 2/REXX
APPC Performance Tips for OS/2 EE
EASEL OS/2 EE PROFS: Host Code Interface
PS/2 RPG II Application Platform and Toolkit
The IBM Independence Series Products

Issue 1, 1990 (G325-5005) (Out of Print)

Issue 4, 1989 (G325-5004)
Requirements for Advanced Bus Architecture
Micro Channel System Configuration Considerations
Features and Benefits
Bus Masters and Applications
Overview of Extended Micro Channel Functions
New Micro Channel Features
SCB – An Architecture for Micro Channel Bus Masters
Design Alternatives with Micro Channel Systems
The IBM PS/2 Micro Channel SCSI Adapters
PS/2 Wizard Adapter
Experience in Bus Master Design
Bus Master Adapters from Independent Option Vendors
Bus Masters and OS/2
Micro Channel Issues in AIX PS/2
Information for Developers
Book Report: "The Winn Rosch Hardware Bible"

"**Acting as a 32-bit bus master, XGA can access both system memory and the video display buffer memory.** (page 3)

"**Most of the attention on UTP has been focused on the wiring between the telecommunications closet and the work station outlet.** (page 7)

"**Using external alias definitions is a common way to make external connections.** (page 15)

"**You are now logged on to both servers concurrently!** (page 20)

"**The DOS Extender allows applications to directly access the entire physical memory up to 16 MB without any special interface.** (page 27)

"**The source of the problem will probably be found in the LIBPATH statement in your CONFIG.SYS file.** (page 32)

"**Another facility provided by Database Manager that may improve performance in this environment is the Database Application Remote Interface.** (page 36)

"**This program demonstrates some of EASEL's new features and can be modified to meet the user's additional needs.** (page 53)

"**In order to enable a product, code should be structured to make modification easy.** (page 63)

"**The OS2EE12 gateway is similar in many ways to the PCOM3270 gateway.** (page 75)

G325-5010-00