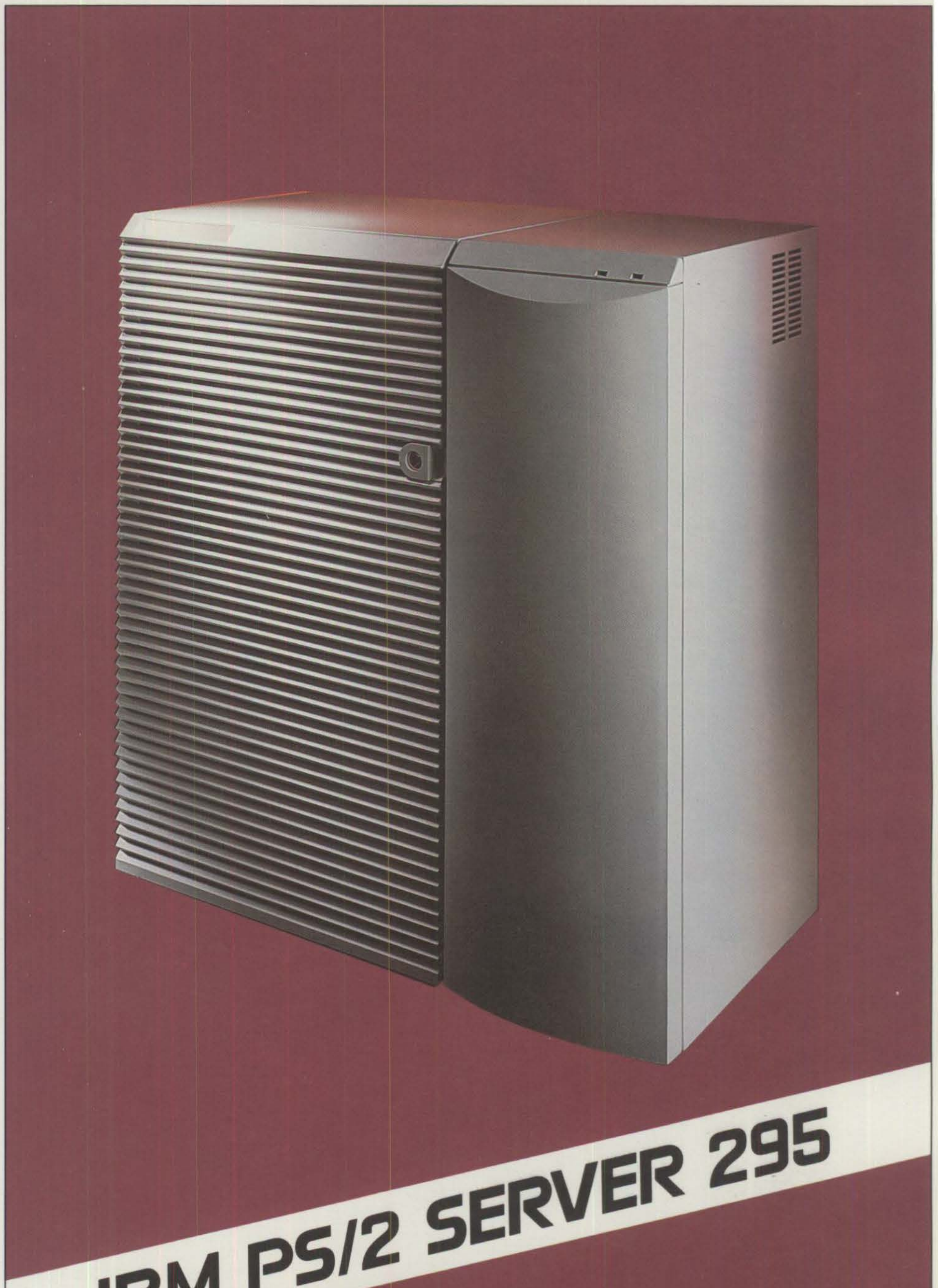
Personal Systems

July 1992



IBM PS/2 SERVER 295

IBM Personal Systems Technical Solutions

IBM®

$12.00

IBM Personal Systems Technical Solutions is published quarterly by IBM United States technical support center, International Business Machines Corporation, Roanoke, Texas, U.S.A.

Editor	Libby Boyd
Communications	Elisa Davis
Cover Design	Corporate Graphics
Illustrator	Bill Carr
Publication Services	The TDA Group
Typesetting & Diagrams	Genesis Publications
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, call 1-800-551-2832. IBM employees should order through SLSS using order form number GBOF-1229.

Copying or reprinting material from this magazine is strictly prohibited without the written permission of the editor. Titles and abstracts, but no other portions, of information in this publication may be copied and distributed by computer-based and other information-service systems.

IBM believes the statements contained herein are accurate as of the date of publication of this document. However, IBM hereby disclaims all warranties as to materials and workmanship, either expressed or implied, including without limitation any implied warranty of merchantability or fitness for a particular purpose. In no event will IBM be liable to you for any damages, including any lost profits, lost savings or other incidental or consequential damage arising out of the use or inability to use any information provided through this service even if IBM has been advised of the possibility of such damages, or for any claim by any other party.

Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so the above limitation or exclusion may not apply to you.

This publication could contain technical inaccuracies or typographical errors. Also, illustrations contained herein may show prototype equipment. Your system configuration may differ slightly.

IBM has tested the programs contained in this publication. However, IBM does not guarantee that the programs contain no errors.

This information is not intended to be a statement of direction or an assertion of future action. IBM expressly reserves the right to change or withdraw current products that may or may not have the same characteristics or codes listed in this publication. Should IBM modify its products in a way that may affect the information contained in this publication, IBM assumes no obligation whatever to inform any user of the modifications.

Some of the information in this magazine concerns future products, or future releases of products currently commercially available. The description and discussion of IBM's future products, performance, functions, and availability are based upon IBM's current intent and are subject to change.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not imply giving license to these patents.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming or services in your country.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever.

All specifications are subject to change without notice.

Printed on recycled paper

Contents

Hardware

Software

Random Data

Trademarks

IBM, Micro Channel, Operating System/2, PS/2, OS/2, Personal System/2, AT, Personal Computer AT, AIX, Systems Application Architecture, Presentation Manager, DB2, BookMaster, OS/400, NetView, and AS/400 are registered trademarks of International Business Machines Corporation.

PC XT, SAA, RISC System/6000, ES/9000, Personal Computer XT, BookManager, and Series/1 are trademarks of International Business Machines Corporation.

ANSI is a registered trademark of the American National Standards Institute.
Apple, Macintosh, LocalTalk, and AppleTalk are registered trademarks of Apple Computer, Inc.
Ellipse is a trademark of Cooperative Solutions, Inc.
ARCNET is a registered trademark of Datapoint Corporation.
Dun & Bradstreet is a registered trademark of Dun & Bradstreet, Inc.
GEnie is a registered trademark of General Electric Company.
PenPoint is a trademark of Go Corporation, Inc.
Hayes is a registered trademark of Hayes Microcomputer Products, Inc.
Hewlett-Packard, HP, and LaserJet are registered trademarks of Hewlett-Packard Company.
IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers.
OASAS is a trademark of Integra Technologies, Inc.
Intel is a registered trademark and i486 is a trademark of Intel Corporation.
COMDEX is a registered trademark of the Interface Group, Inc.
Lotus, Lotus Notes, and 1-2-3 are registered trademarks of Lotus Development Corporation.
Micro Focus is registered trademark of Micro Focus Limited.
Microsoft and Windows are registered trademarks of Microsoft Corporation.
Motorola is a trademark of Motorola, Inc.
Novell and NetWare are registered trademarks of Novell, Inc.
Parallan is a trademark of Parallan Computer, Inc.
PeopleSoft is a registered trademark of PeopleSoft, Inc.
OmniDesk is a trademark of Sigma Imaging Systems, Inc.
Softerm is a registered trademark of Softronics, Inc.
Sony is a registered trademark of Sony Corporation.
Radio Shack is a registered trademark of Tandy Corporation.
Toshiba is a registered trademark of Toshiba Corporation.
Rumba is a registered trademark of Wall Data Incorporated.
WordPerfect is a registered trademark of WordPerfect Corporation.
Ethernet is a trademark of Xerox Corporation.

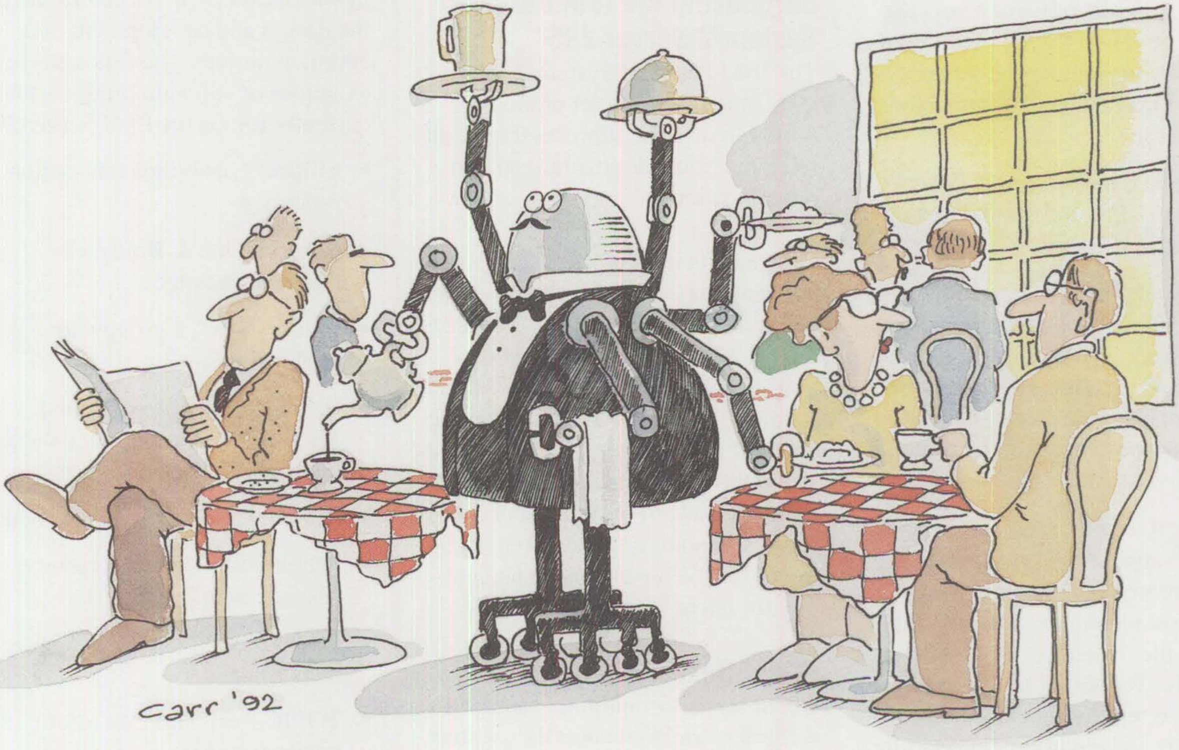
IBM PS/2 Server 295: New Thresholds for Client/Server Networking

Mike Engelberg
The TDA Group
Los Altos, California

IBM's new PS/2® Server 295 answers LAN users' demands for high reliability, data integrity, high performance, and fault tolerance in an application database server. Sophisticated technologies implemented in the PS/2 Server 295 make it extremely reliable and suitable for use in mission-critical environments such as banking, public utilities, securities trading, and airline reservations. Software in the PS/2 Server 295 enables administrators to manage system resources from central locations, eliminating the need to have technical experts at each location on a network.

The new PS/2 Server 295 has much to offer users. Consider these impressive facts:

- Mass storage devices whose Mean Time Between Failure (MTBF) is expected to be at least 18 years and up to more than 40 years
- A server that keeps functioning even if a hardware component fails, and whose storage devices can be replaced while the computer is running
- A logical array of storage devices, across multiple Small Computer Systems Interface (SCSI) buses, which can protect data against failure of any component – even power supplies
- Fault tolerance that logically disconnects a failing component and



switches to other working components and to spare components if installed – all without human intervention

- A hierarchical bus architecture that features dual Micro Channel® expansion buses and a 64-bit, 200 MB per second main bus where up to eight specialized processors are attached – for application processing, data file and network transactions, data movement within memory, and system control and maintenance
- Up to four SCSI bus channels, each with its own disk processor, that control up to 28 GB of total storage
- Up to two 32-bit, 20 MB per second Micro Channel computing subsystems that feature Intel® 80486 cached processors, configured as tightly coupled microprocessors, running at either 33 MHz or 50 MHz
- Up to twelve slots for Micro Channel adapters and busmaster cards, including multiple Local Area Network (LAN) attachments
- Up to 128 MB of Error Checking and Correction (ECC) high-speed memory in an independent subsystem controlled by a 64-bit memory processor
- Optional Uninterruptible Power Supply (UPS) and optional redundant power supply
- A separate maintenance processor, with battery backup, that monitors system activity and usage; detects, logs, and highlights system errors; and enables a remote system administrator to tune and reconfigure the server system
- Interactive software, running as an OS/2® application, which provides the remote system administrator with instantaneous status reports about the overall server system, and also the ability to ask for detailed reports about system components and environmental settings
- Full compatibility with major industry standards, including IBM's Systems Application Architecture® (SAA™); networking architectures such as IBM's Systems Network Architecture (SNA); operating systems such as OS/2; graphical user interfaces such as the OS/2 Workplace Shell; and LAN managers such as OS/2 LAN Server
- An application database server system that protects users' investments when they add to it at a later date

How many computer systems does it take to provide all this? Now, just one – and what a computer it is!

The IBM PS/2 Server 295 is the first product of a long-term alliance between IBM and Parallan Computer, Inc.

Introducing the IBM Personal System/2 Server 295

The IBM Personal System/2® Server 295 is the first member of the PS/2 family to offer multiprocessing, fault tolerance, and administration from other locations.

Designed for mission-critical database applications that run in client/server network environments, the IBM PS/2 Server 295 is the first product of a long-term alliance between IBM and Parallan™ Computer, Inc. Under the terms of this alliance, IBM has exclusive rights to manufacture and market Parallan's award-winning application server technology, while Parallan continues to develop advanced technology for client/server computing.

Client/Server Computing

Client/server computing is an architectural model that takes the greatest advantage of each component on a computer network – the central server and the remote (client) computers. In client/server computing, an application is divided between an application database server and client computers. Clients request data that is stored on the server, receive that data, and further process it; the server performs functions that require large amounts of storage, memory, file processing, transmission, and data security.

Client/server networks require powerful, reliable, secure server systems. The degree to which the PS/2 Server 295 responds to these requirements makes it suitable as the server in a network that handles mission-critical applications, such as public utility service and timely transaction processing.

Typical Applications

Enterprises use client/server computing in applications such as relational database management; communications and electronic mail; financial spreadsheets; word processing and desktop publishing; and groupware, in which diverse users contribute to the design and development of a common project. The following are examples of software products that currently run on the PS/2 Server 295:

- Ellipse™, an online transaction processor
- Selected Dun & Bradstreet® financial packages
- Lotus Notes®, a groupware application
- IBM Database Manager and Microsoft® SQL Server, database management systems
- IBM OS/2 Communications Manager
- PeopleSoft® Human Resource Management System
- OmniDesk™, an image management system
- TOPIC Real-Time, an information retrieval system

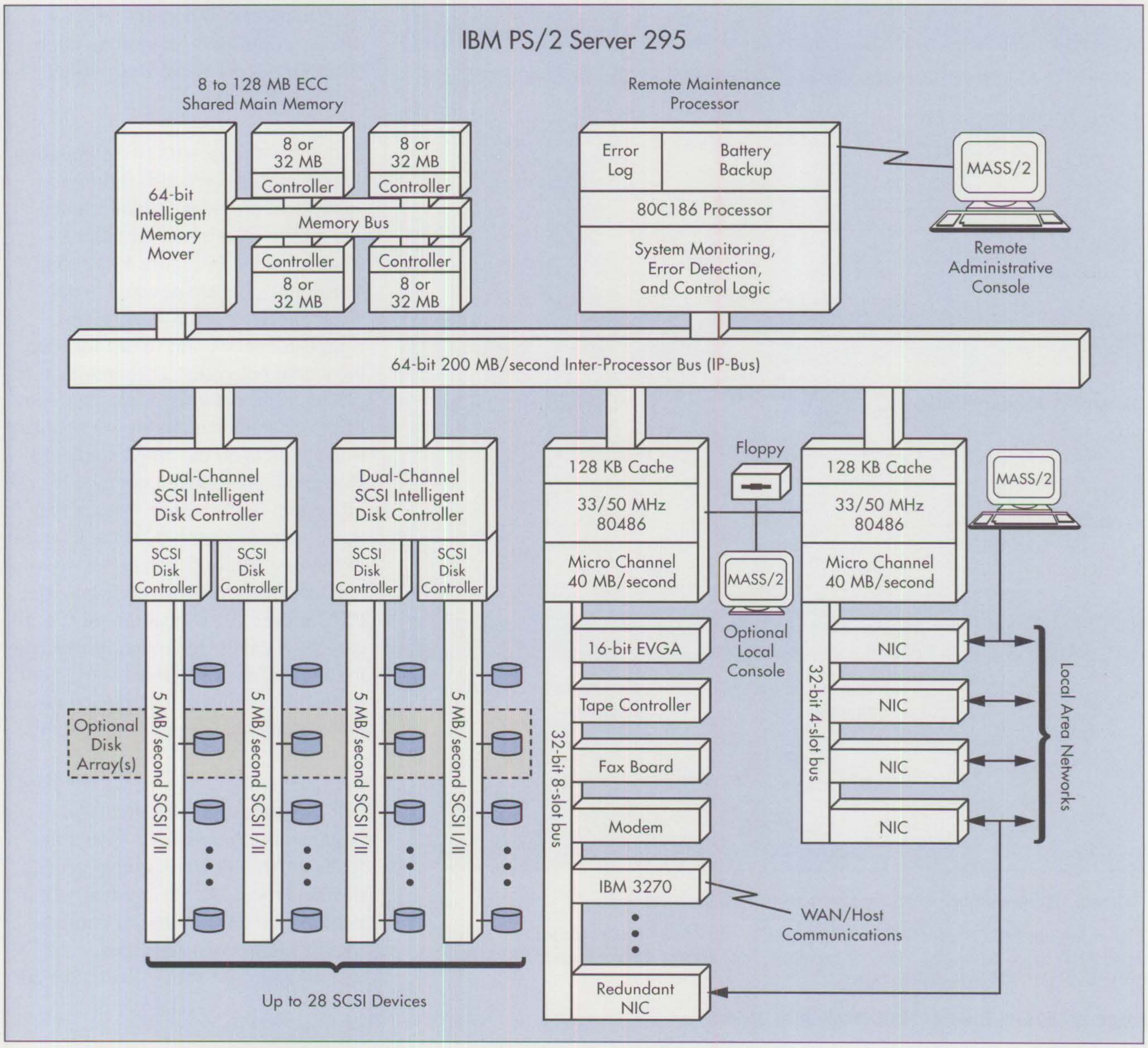


Figure 1. PS/2 Server 295 Architecture

The PS/2 Server 295 is an open system; that is, its operating system is not proprietary. All applications developed to run under OS/2 will run on the PS/2 Server 295.

Architectural Overview

Figure 1 gives an overview of the architecture of a fully configured PS/2 Server 295. The basis of the entire architecture is the 64-bit Inter-Processor (IP) Bus with its incredible traffic capacity – 200 MB per second. With this capacity, the IP Bus is the major traffic artery within the entire system, linking all the specialized subsystems: computing, memory, mass storage, and maintenance.

Although the IP Bus is standard in all PS/2 Server 295 systems, all the subsystems except maintenance have configuration options. Enterprises can order their PS/2 Server 295 systems in configurations designed to meet their specific needs. The PS/2 Server 295's modular design enables enterprises to add resources as their needs grow. Figure 2 lists the available configuration options within each subsystem.

PS/2 Server 295 systems are built to order. Enterprises can order PS/2 Server 295 systems in any configuration that is within the minimum and maximum limits shown in Figure 2. Other components are also available, such as network adapters, high-speed

Computing Subsystem

Number of 33 MHz 80486s	Number of 50 MHz 80486s	Total Number of 80486s
1	0	1
0	1	1
1	1	2
2	0	2
0	2	2

Memory Subsystem

Number of Memory Banks	Amount of Memory Per Bank	Total Memory Capacity
1	32 MB	32 MB
2	32 MB	64 MB
3	32 MB	96 MB
4	32 MB	128 MB

Storage Subsystem

Number of SCSI Channels[1]	Minimum Storage Capacity	Maximum Storage Capacity
2	400 MB (1 hard disk @ 400 MB)	14 GB (14 hard disks @ 1 GB)
4	800 MB (2 hard disks @ 400 MB)	28 GB (28 hard disks @ 1 GB)

[1] The maximum number of storage devices per SCSI channel is 7.

The minimum PS/2 Server 295 configuration is one 80486 processor (either 33 MHz or 50 MHz), one bank of 32 MB of memory, one storage device (either 400 MB or 1 GB), and one 1.44 MB diskette drive.

Figure 2. IBM PS/2 Server 295 Configuration Options

tape backup devices, modems, and fax adapters. IBM will install and test all components ordered. IBM software – OS/2 and the Maximum Availability and Support System/2 (MASS/2) – is standard with all configurations, and is preloaded before delivery.

Computing Subsystems

In the lower right quadrant of Figure 1 are two complete computing subsystems. In a two-processor computing environment, the processors are configured as multiprocessors, capable of performing simultaneous operations. The 33 MHz and 50 MHz processors come with an 8 KB cache on the processor chip itself. The 33 MHz processor has a 128 KB, zero-wait-state secondary cache; the 50 MHz processor has a 256 KB, zero-wait secondary cache.

In PS/2 Server 295 configurations with one computing subsystem, the "computer" is the one on the left in Figure 1 – the one whose Micro Channel bus has eight slots (for seven 32-bit adapters and one 16-bit adapter). This bus can accommodate adapters for computer components such as display controllers, communication controllers, and Network Interface Controllers (NICs).

In configurations with two computing subsystems, the second (rightmost) subsystem provides four additional slots, for a total of twelve Micro Channel slots. In Figure 1, the four additional slots are devoted to network interface controllers (token ring, Ethernet™, and so on) for four local area networks. This architecture places faster network adapters on one Micro Channel bus and slower application adapters on the other, so that the slower adapters do not degrade overall system performance. Notice that there is a redundant NIC in the last slot of the first Micro Channel bus. System administrators can configure the PS/2 Server 295 so that the most critical LANs can be redundantly attached to the first Micro Channel subsystem to ensure continued up-time in case the primary subsystem has problems.

In Figure 1, the eight-slot Micro Channel subsystem is configured as the Application Processor (AP) and the four-slot Micro Channel subsystem as the File Processor (FP). System administrators have the option of configuring the eight-slot subsystem as the FP and the four-slot subsystem as the AP.

Memory Subsystem

The upper left quadrant of Figure 1 shows the PS/2 Server 295's memory subsystem. Memory is accessed through a 64-bit high-speed interface to improve performance. The memory itself is 80 ns ECC page-mode memory. It has up to four memory banks, each with 32 MB of memory, for a total of 128 MB. Each memory bank has its own independent memory controller that performs concurrent memory bank accessing.

Disk Controller Subsystems

The lower left quadrant of Figure 1 has two RISC-based SCSI disk processors called Intelligent Disk Controllers (IDCs). Each IDC has two SCSI channels, for a total of four SCSI channels. (The minimum configuration has a single IDC with two SCSI channels.) Each channel is capable of transferring data at the rate of 5 MB per second; all four channels can handle a total of 20 MB of data per second. Each channel has its own SCSI disk controller and up to seven attached devices. The devices can be either 400 MB or 1 GB SCSI hard disk drives. When all four SCSI channels are fully populated with 1 GB disks, the IBM PS/2 Server 295 has 28 GB of mass storage.

Figure 1 shows a shaded area called Disk Arrays that crosses all four SCSI disk buses. This is one primary fault-tolerance feature of the PS/2 Server 295, and will be discussed later.

Remote Maintenance Subsystem

The final processor component of the PS/2 Server 295 is the Remote Maintenance Processor (RMP) subsystem, shown in the upper right quadrant of Figure 1. This processor is an 80186 with 128 KB of Static RAM (SRAM) and a 256 KB Erasable Programmable Read-Only Memory (EPROM). The major purposes of this processor are to enable a system administrator – who may be at a remote location – to monitor, tune, and control the PS/2 Server 295 without affecting its throughput, and to track and log problems that may occur in the system. To make the problem log available when the system's power is disabled, the RMP subsystem includes a nickel-cadmium rechargeable backup battery with a life of up to 15 hours without recharging.

Storage Devices and Enclosures

Figure 3 shows a cutaway view of the PS/2 Server 295. Its physical enclosure is divided into two major parts: the system enclosure for buses and circuit boards, and the storage enclosure. The configuration shown has room for 9 half-height disk drives, diskette drives, or Digital Audio Tape (DAT) drives. To attain the maximum of 28 drives, three more storage enclosures can be attached. Storage enclosures have a solid front door that keeps dust out of the drives. Within the main enclosure, air coming in is filtered before being circulated. Some physical security devices available for cabinetry and for the cables inside are shown in Figure 3.

The storage enclosure that comes with the PS/2 Server 295 accommodates 9 devices. If a PS/2 Server 295 configuration has 10 to 19 storage devices, a second storage enclosure is required; if a configuration has 20 to 28 storage devices, a third storage enclosure is required.

Reserved Memory Manager

The PS/2 Server 295's Reserved Memory Manager (RMM) enables application programs to access all the available memory. An application, therefore, can access and use as much as 128 MB of memory. This greatly enhances the performance of applications that heavily use memory, such as database programs. Those applications can take advantage of large amounts of main memory directly or through large virtual RAM drives.

Fault Tolerance Features

The PS/2 Server 295 sets new heights of fault tolerance in client/server environments. Advances in hardware technology and software capability give the PS/2 Server 295 a level of fault tolerance not previously available in a client/server network environment. The list of fault tolerance features includes the following:

- MASS/2 software
- Remote Maintenance Processor
- Online spare storage devices
- Orthogonal RAID-5 Disk Array/2
- Dual 80486 processing subsystems
- Dual Micro Channel buses
- Up to two dual SCSI buses
- Parity checking on all buses in the system
- ECC memory
- An optional UPS and redundant power supply

Two of these features – MASS/2 and Orthogonal RAID-5 Disk Array – are significant advances.

Maximum Availability and Support Subsystem/2

The MASS/2 software is a set of tools for monitoring, tuning, and controlling PS/2 Server 295 systems from local or remote locations. Together with the PS/2 Server 295's fault-tolerant hardware technology, MASS/2 enables PS/2 Server 295 systems to recover from failures and to continue to run – all without the intervention of the system administrator. MASS/2 is easy to use, and it runs without affecting the performance of the server system.

Briefly, MASS/2 provides these functions:

- Monitoring and control of resource utilization
- Configuration management from remote locations
- Establishment of thresholds for continued operation of the system and for alarms
- System problem notification when attention is necessary
- Battery access to system trace logs in case of power outage

MASS/2 runs on the Remote Maintenance Processor discussed earlier. MASS/2 tracks hardware and network problems, notifies the administrator whenever a problem occurs, and enters the problem information into a log kept in the RMP subsystem. Because the RMP has battery backup, the administrator can access

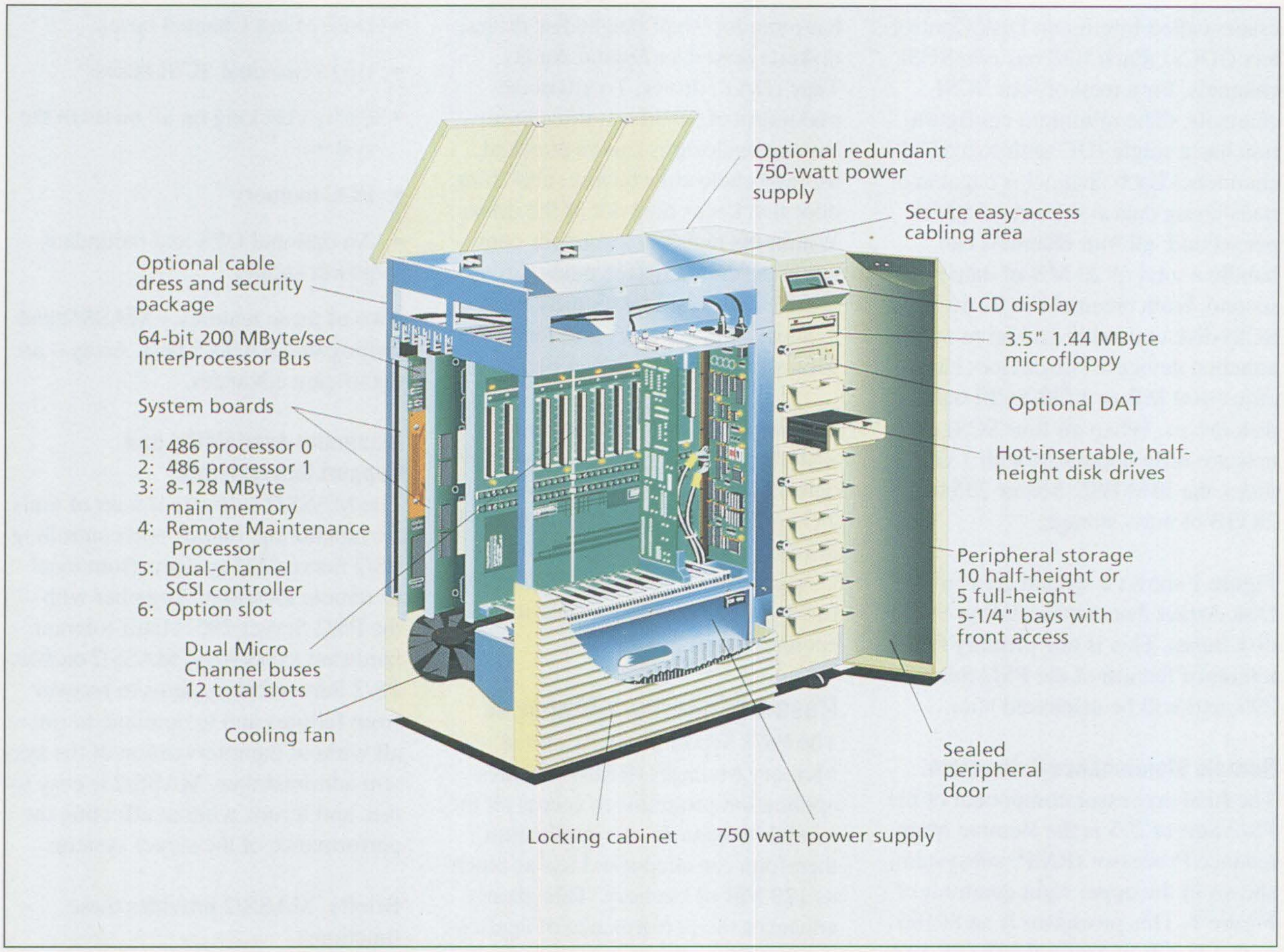

Optional redundant 750-watt power supply
Secure easy-access cabling area
LCD display
3.5" 1.44 MByte microfloppy
Optional DAT
Hot-insertable, half-height disk drives
Peripheral storage 10 half-height or 5 full-height 5-1/4" bays with front access
Sealed peripheral door
750 watt power supply
Locking cabinet
Cooling fan
Dual Micro Channel buses 12 total slots
Optional cable dress and security package
64-bit 200 MByte/sec InterProcessor Bus
System boards
1: 486 processor 0
2: 486 processor 1
3: 8-128 MByte main memory
4: Remote Maintenance Processor
5: Dual-channel SCSI controller
6: Option slot

Figure 3. Cutaway View of PS/2 Server 295

the problem log even when power to the system is disabled.

As shown in Figure 1, MASS/2 can be run in several locations. In the upper right corner of Figure 1, a remote terminal running MASS/2 is connected to the RMP over a communications line. Figure 1 also shows that MASS/2 can be run from a monitor attached to a client on a LAN, or from a monitor attached locally to the 80486 computing subsystems. This flexibility enables the system administrator to be located at a remote site and to control several PS/2 Server 295 systems from that site.

MASS/2 Monitor Screen: Figure 4 shows the MASS/2 Monitor screen. This screen can be displayed on monitors attached to the PS/2 Server 295, to a client on a LAN, or to a remote computer that uses a modem to connect to the server. From the MASS/2 Monitor screen, a single system administrator can manage many PS/2 Server 295 computers located in several different places.

Figure 4 displays, in graphic form, the overall status of all subsystems in a PS/2 Server 295. At a glance, the system administrator can see what is happening in a server system. The administrator can bring up additional information about any single component.

The left side of the MASS/2 Monitor screen shows the processors that are attached to the IP Bus. IP Bus slot 0 has the first of two 80486 computing subsystems, labeled FP; slot 1 has the second 80486, labeled AP. Slot 2 contains the memory processor; slot 3 has the Remote Maintenance Processor; and slot 4 has a SCSI channel processor. Slot 5, which is empty, is reserved for the second dual-channel SCSI processor.

In the lower left corner are twelve buttons indicating the twelve slots on the 80486 Micro Channel subsystem buses. The FP is the one with eight slots; the AP has four. Notice that button 7 in the FP subsystem and button 2 in the AP subsystem are highlighted. This means the administrator is requesting detailed information about the adapters in those slots. The screen that displays the detailed information is shown in Figure 5.

If an optional UPS is installed in the PS/2 Server 295, its status is shown in the lower left corner of Figure 4.

The upper right quadrant of Figure 4 displays real-time statistics and bar charts of the utilization of all major subsystems. In the lower right quadrant, the administrator can select the subsystem and display the history of that subsystem's utilization during the past hour. (The subsystem shown in Figure 4 is the dual 80486 computing subsystem.) The hourly statistics are saved on disk for subsequent analysis. These statistics are useful for tuning the system configuration. The administrator can tell from the utilization statistics whether a subsystem's usage is heavy or light, and can therefore determine which subsystems have excess capacity and which need more resources.

Warning Thresholds: MASS/2 maintains system warning thresholds that the administrator can set. MASS/2 will then notify the administrator if the limits are exceeded. Figure 6 shows the Autodial Error screen, which contains several categories of potential system errors. MASS/2 sends a warning to the system administrator whenever a threshold for a selected item is exceeded. The administrator can then take appropriate action to notify users, to shut down the system if necessary, or to request maintenance. Usually the administrator need not take action, because MASS/2 has already interceded to keep the system running.

Soft Shutdown and Reboot: If a hardware component or an application program fails, the PS/2 Server 295 system and the MASS software work in tandem to ensure minimal impact on continued operation. The server system has built-in diagnostics that can identify a faulty hardware

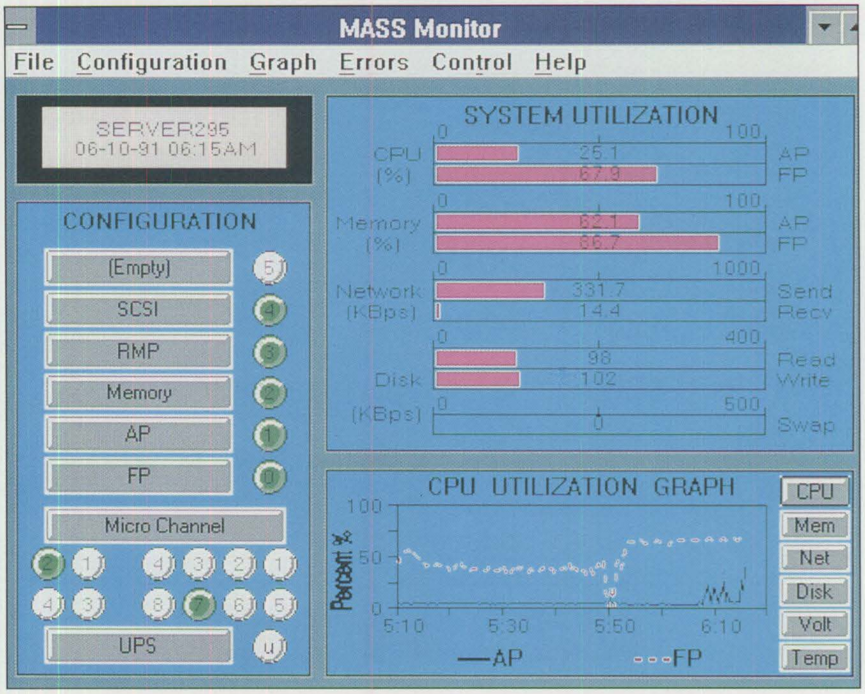
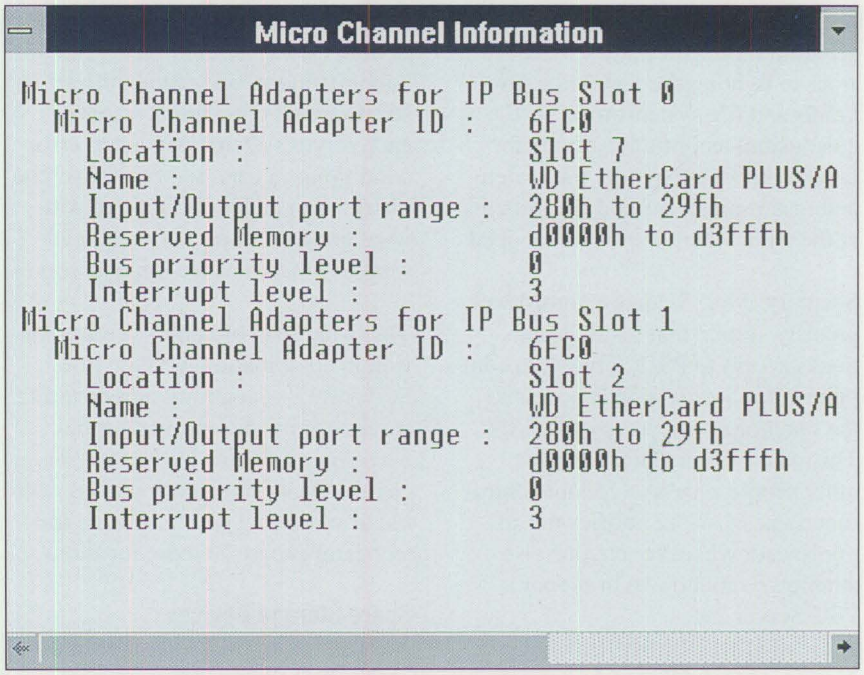
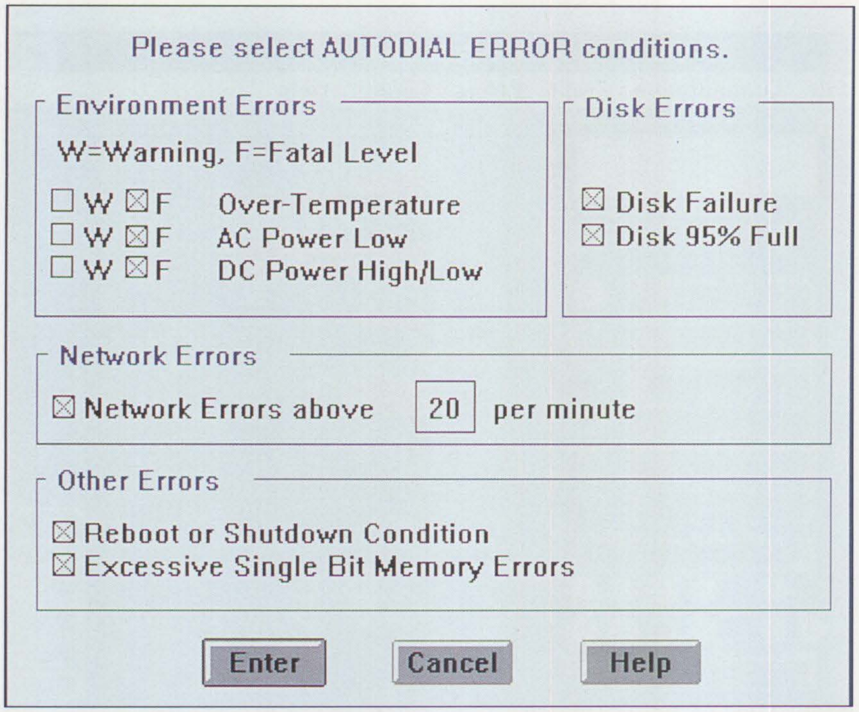


Figure 4. MASS/2 Monitor Screen



```
Micro Channel Adapters for IP Bus Slot 0
  Micro Channel Adapter ID :    6FC0
    Location :                  Slot 7
    Name :                      WD EtherCard PLUS/A
    Input/Output port range :   280h to 29fh
    Reserved Memory :           d0000h to d3fffh
    Bus priority level :        0
    Interrupt level :           3
Micro Channel Adapters for IP Bus Slot 1
  Micro Channel Adapter ID :    6FC0
    Location :                  Slot 2
    Name :                      WD EtherCard PLUS/A
    Input/Output port range :   280h to 29fh
    Reserved Memory :           d0000h to d3fffh
    Bus priority level :        0
    Interrupt level :           3
```

Figure 5. MASS/2 Micro Channel Adapter Information Screen



Please select AUTODIAL ERROR conditions.

Environment Errors

W=Warning, F=Fatal Level

□ W ☒ F Over-Temperature
□ W ☒ F AC Power Low
□ W ☒ F DC Power High/Low

Disk Errors

☒ Disk Failure
☒ Disk 95% Full

Network Errors

☒ Network Errors above 20 per minute

Other Errors

☒ Reboot or Shutdown Condition
☒ Excessive Single Bit Memory Errors

Figure 6. MASS/2 Autodial Error Screen

component and logically disconnect it from the system. The server can also sense, through timers, when a program freezes or aborts. MASS/2 steps in to bring the remaining programs and file systems to a "soft" conclusion, reconfigure around the failed component, reboot the system, and restart programs and file systems at the point where they were stopped.

Security: MASS/2 has a multi-level security system that restricts and tracks access to PS/2 Server 295 computers. Passwords protect access to the functions provided by MASS/2. The main administrator can give other people access to certain control functions. MASS/2 notifies the administrator whenever anyone else attempts to shut down or reboot a PS/2 Server 295.

More Control, Higher Levels of Support, Fewer Resources: With MASS/2, system administrators enjoy a level of control not previously available. MASS/2 has a remote console feature that enables administrators to control and manage, from a central location, multiple PS/2 Server 295 computers that are installed at several remote locations. Centrally located administrators can see the utilization statistics and operational status of each server system, respond to error conditions on each server, control the distribution and installation of software updates for each system, and schedule maintenance for each server.

This wide span of administrative control, in combination with the PS/2 Server 295's capability of recovering from error conditions, means that enterprises can now have client/server environments at remote locations without also needing administrative and technical experts at those locations.

Spare Storage Devices

Another major fault tolerance feature of the PS/2 Server 295 is its implementation of spare storage. Any disk drive in a storage array can be configured as a spare. A spare storage device is used only to substitute for another device that has problems. Whenever the PS/2 Server 295 senses that a storage device is failing, it activates a spare device, reconfigures the array, and activates the spare in place of the failing device. All this is done with minimum impact on the overall performance of the server system, and with no intervention by the system administrator.

When a storage device goes down, MASS/2 notifies the system administrator. The administrator can then request replacement of that device. Notice that all this time the LAN is up and running. In Figure 3, the storage device in the fourth slot down is labeled "hot-insertable." This means that the service technician can remove the faulty device and insert a new one while the PS/2 Server 295 is up and running. The new storage device becomes the spare.

Spare storage devices are included in the maximum number of 7 devices per channel or 28 devices per PS/2 Server 295 system.

Orthogonal RAID-5 Disk Array/2 Software

One element in Figure 1 is left to discuss: the shaded area in the lower left corner, labeled "Disk Arrays." The Orthogonal RAID-5 (Redundant Array of Inexpensive Disks) Disk Array/2 software is a major fault tolerance feature that enables the PS/2 Server 295 to protect against failure of any kind of component, even the power supply.

Typically, logical arrays are set up on a single SCSI bus, and consist of several storage devices and a single dedicated parity device to aid in the reconstruction of data if necessary. Having only one parity device constrains the overall performance of the devices in the array, which in turn slows the entire server. RAID-5 distributes the parity information among all the devices in the logical array. Parity writes occur simultaneously

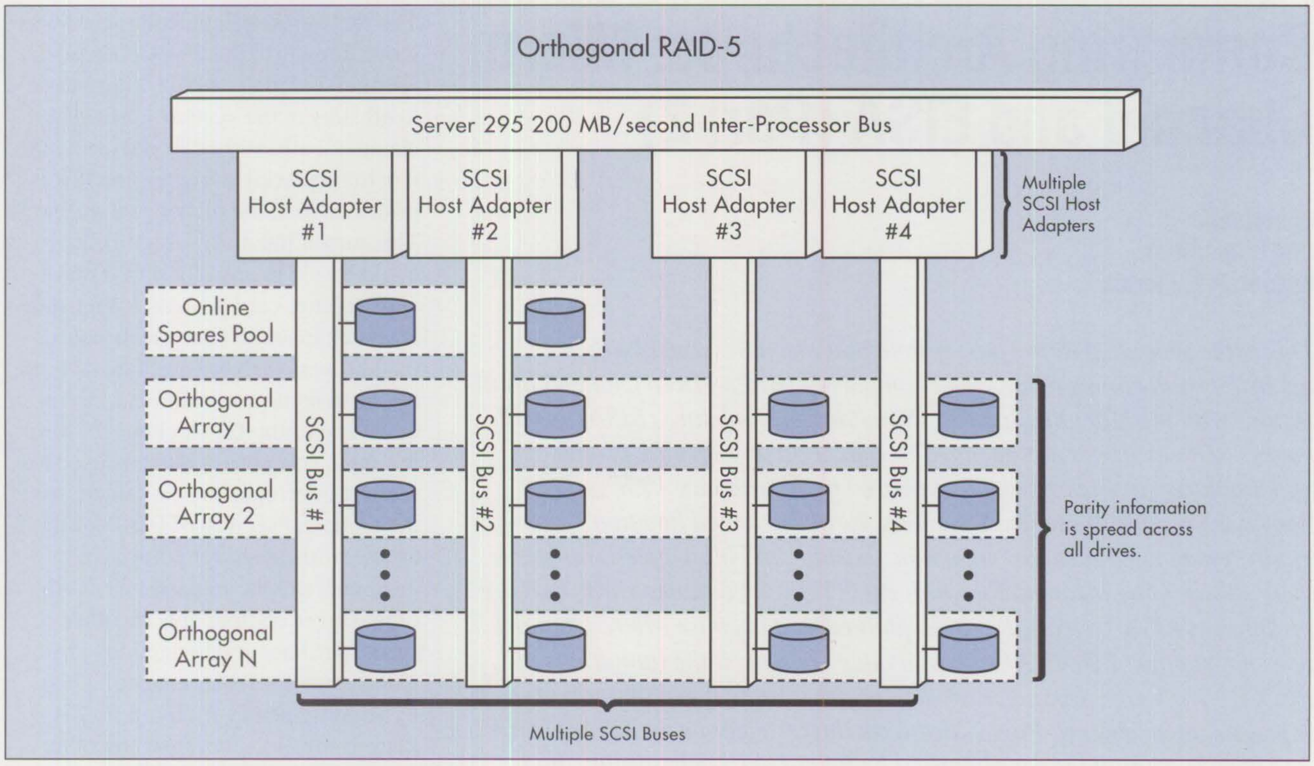


Figure 7. Orthogonal RAID-5 Disk Array Concept

with data access. Spreading the parity information among all devices in an array ensures performance and data reliability.

Orthogonality takes the RAID array concept a step farther. Logical arrays are set up to span multiple SCSI buses, as shown in Figure 7. Each SCSI bus has its own processor that accesses data. By multiplexing data-access activity across four SCSI processors, the Orthogonal RAID-5 concept significantly improves data-access performance.

Each disk array can have from 2 to 16 storage devices, and there can be multiple arrays, adding to the overall performance of the storage subsystem.

An important benefit of the orthogonality concept is reliable system recovery. Suppose either a storage device or a SCSI bus fails. Making calculations based on the parity data that is spread across all the SCSI buses, Disk Array/2 recreates the data that is on the faulty device, writes that data onto a spare device, and makes that device active within the array. All this takes place while the PS/2 Server 295 keeps running.

Because it encompasses multiple SCSI channels and storage devices, the Disk Array/2 software ensures that server performance will continue if any of these components fails. Software and hardware logic reroutes data-access requests to alternate storage devices, as mentioned above, and to alternate SCSI channels. If an active storage device is on a channel whose SCSI controller fails, that device is removed from its disk array. The Disk Array/2 software then reconstructs the contents of that "missing" device as follows:

If a spare device is available, the following occurs:

- Disk Array/2 reconfigures the disk array to include the spare device.
- Using parity information that is spread among the remaining devices in the array, Disk Array/2 reconstructs the "missing" data on the spare device.

If no spare device exists, the Disk Array/2 software continues to dynamically reconstruct data requested from the "missing" drive. Note that data requested from all other drives continues to be available without any performance penalty.

__Mike Engelberg__ is a 26-year veteran of IBM who is now an editorial consultant. While with IBM, he was an applications programmer and the editor of several technical newsletters and magazines about the IBM Personal Computer and Personal System/2 families and their software. He has a BS in mathematical statistics from the University of Chicago and an MBA from the University of Illinois.

Comparing Architectures: Micro Channel and EISA (Part 2)

Chet Heath
IBM Corporation
Boca Raton, Florida

This is the second of a two-part article that discusses and clarifies assertions and claims made when comparing IBM's Micro Channel architecture with the Extended Industry Standard Architecture (EISA) specifications. The article is not intended to reflect on any specific product or manufacturer. In Part 1, which appeared in the January 1992 issue of Personal Systems Technical Solutions, *EISA advanced function – automatic system configuration, interrupt sharing, and high-speed transfer – were shown to be incompatible with PC/XT™/AT® adapter cards that are installed in EISA systems. Part 1 explained that because Micro Channel does not support PC/XT/AT cards, it does not inherit the limitations of PC/XT/AT systems. Finally, Part 1 gave the technical explanation of interrupt-sharing problems. Part 2 discusses the restrictions on EISA's DMA and bus arbitration that are imposed by PC/XT/AT compatibility, and shows the pitfalls of EISA's design and strategy compared to the design and strategy of Micro Channel architecture.*

When the IBM Personal Computer was introduced in 1981, it had a major advantage over the current market leader, the Apple® II computer: the PC's Direct Memory Access (DMA) enabled some devices to operate at higher speeds than were possible with 8-bit processors alone. A "DMA channel" acted like a second processor with only one instruction: block move. The PC hardware used bus cycles efficiently, moving one byte at a time across the 8-bit PC bus between memory and I/O adapters. The protocols for DMA data movement were embedded in the design of the silicon, and did not involve the processor except to start and end movements of blocks of data. The IBM PC had four DMA channels: one used by the system and three available to cards.

Conflict Between PC/XT/AT Cards: Unusable DMA Channels

It was not long until the three DMA channels available to adapter cards were permanently reassigned to specific system functions. After that, newly developed adapter cards were either precluded from using DMA, or if they attempted to use DMA, they caused conflicts and possible permanent damage.

DMA Channel 2: Diskette Controller

For I/O adapters that move large blocks of data, DMA was more efficient and faster than the processor. For example, the diskette controller, initially an option in the original IBM PC, needed to move 512-byte blocks of data at 50,000 bytes per second. DMA supported this rate as well as the much higher speeds of larger capacity diskette and hard disk drives that were to come.

The PC's diskette controller card was wired directly to the DMA controller's channel 2 signals, which were bused to all slots in the system. Choosing channel 2 allowed other device adapters to be placed at higher and lower priorities than the diskette controller. This meant the diskette controller could reside in any slot, but it prevented other card designs from using DMA channel 2. That was because the hardware driver components used by the diskette controller card to request use of the DMA channels were the same component designs used to request interrupts. As explained in the technical section of Part 1 of this article, if these driver component designs share the request lines with other driver components on other cards, permanent damage may ensue. Consequently, DMA channel 2 had to be permanently reserved for diskette controllers, because any other card that used the same DMA channel 2 would potentially damage itself or the diskette controller card. It may have been acceptable to reserve one channel for a function that soon became standard, but it set a precedent for the use of other channels that came later.

DMA Channel 0: DRAM Refresh

Dynamic Random Access Memory (DRAM) is an extremely compact design for large-scale memories in computers. Using Very Large Scale Integration (VLSI), DRAM, with capacity sufficient to support the original IBM PC, can now be packaged as a single unit about the size and shape of a stick of gum. Despite the technical advances, DRAM memory components have one intrinsic characteristic: they "forget." Their memory is volatile if not refreshed every few thousandths of a second. In DRAM, minute amounts of electrical charge remain in one place inside the silicon, to indicate the presence of a logical one or zero. The electrical charge dissipates within a few thousandths of a second, so memory contents will disappear quickly unless they are continuously and rapidly refreshed. Fortunately, the memory system can refresh DRAM by periodically reading its contents. The refresh and the restoration of the charge inside the silicon occur as a result of the read operation.

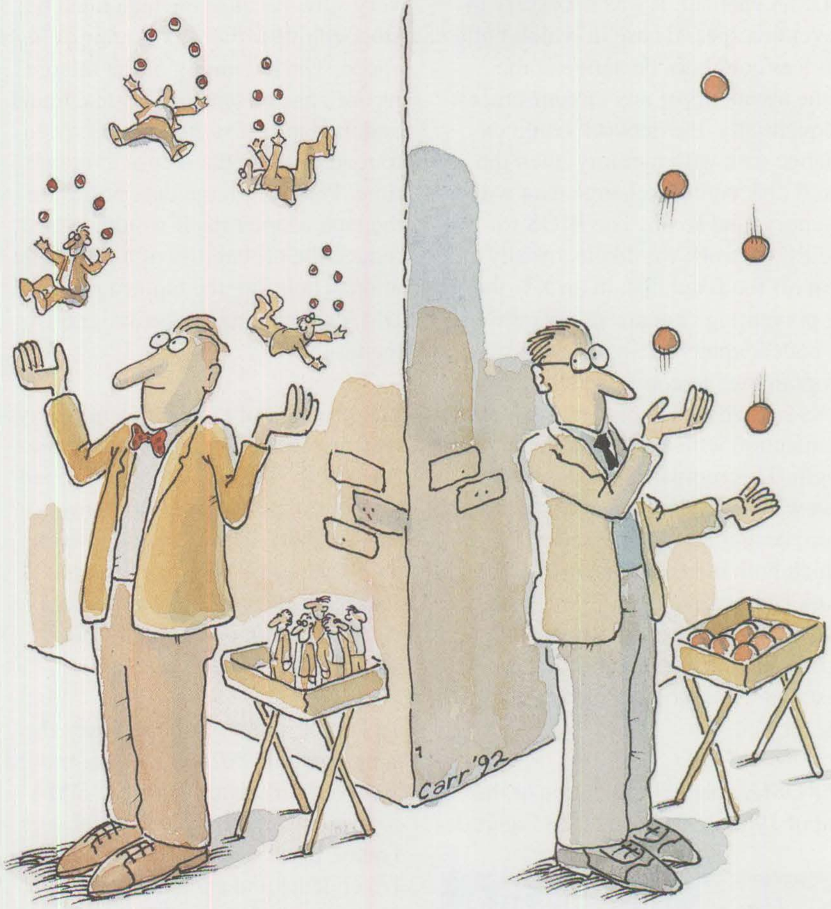
Because data integrity is among the highest priorities in system design, the highest priority channel of the DMA controller – DMA channel 0 – was wired directly to a timer on the system board. The timer requested a dummy read from memory every few thousandths of a second, and that operation refreshed the memory. Thus, the second of four channels on the DMA controller in the PC and PC/XT became permanently dedicated to memory refreshment.

DMA Channel 1: SDLC

Synchronous Data Link Control (SDLC) adapters also characteristically move data as blocks. While they could not justify the high priority of DRAM refresh, SDLC adapters required that a block be retransmitted when the system could not keep up with SDLC's needs. SDLC adapters needed a priority lower than that of DRAM refresh, but higher than that of the diskette controller; therefore, they were hardwired to a fixed assignment at DMA channel 1.

DMA Channel 3: Hard Disk Controller

A diskette drive can wait one more rotation to access data if the system cannot keep up with its needs. This appears as a performance degradation rather than a data integrity problem. Most users never notice that the diskette drive goes back to its master index, finds the data's starting point again, and goes to the appropriate track to catch the data as it passes the read head a second time. This is called a retry operation.



The IBM Personal Computer XT® added a 10 MB hard disk that vastly increased the speed and capacity of the system beyond the relatively slower 360 KB diskette drives of the day. Compared to the XT's diskette drive, a retry operation on the hard disk was far less noticeable. The XT's fixed-disk adapter card was hardwired to the DMA channel 3 request line on the PC/XT bus interface.

At this point in 1983, each DMA controller channel was permanently reserved for one and only one I/O adapter design.

Sharing Channel 3 with a PC Network

The 10 MB hard disk of the IBM PC/XT could support 30 user files, each equivalent to a diskette in capacity. It would be more cost effective if many PC users could somehow share the more costly XT hard disk. Networking many PCs to a common XT server was the answer. But with only a few thousand instructions per second from the 8088 processor, and with all DMA channels assigned, it was difficult to find resources in the system to support a network.

The "temporary" solution was to break a rule and to share DMA channel 3 between the fixed disk and a PC network. It was the only feasible choice, because all the other DMA assignees would fail miserably if forced to share with a slow network. (A more permanent solution would have to await the IBM Personal Computer AT®, which had more DMA channels and a much faster processor that could also move blocks of data.)

To enable the XT's fixed disk adapter and the PC network adapter to share a DMA channel, it was necessary to develop a special case in which both devices could not be active at the same instant under any circumstances. Sequentially, the network could exchange data with memory, then the fixed disk could exchange data with memory, and so on. The BIOS for the PC network would electrically turn off the fixed disk in an XT, thereby preventing concurrent operation of both adapters. Also, DOS was programmed to complete all transactions with the fixed disk before communication with a network could begin. This mutually exclusive situation was inefficient compared to the more desirable concurrent design, in which both the fixed disk and a network can operate at the same time. Indeed, the XT was totally unsuited to be a network server, but it was the best that technology could offer a decade ago.

The DMA channel utilization at the end of 1983 is summarized in Figure 1.

DMA Channel	PC or PC XT Use	Width
0	DRAM Refresh	Byte
1	SDLC	Byte
2	Diskette	Byte
3	Other Fixed Disk or PC Network used sequentially	Byte

Figure 1. DMA Channel Utilization in IBM PC and XT

Concurrent Operation in the PC AT

The IBM PC AT expanded the simple DMA structure in the PC and XT. The AT implemented dedicated hardware to refresh DRAM, thereby freeing DMA channel 0 for use by adapters. This made the refresh operation more efficient. The free DMA channel – the one with the highest priority – was assigned to tape backup systems. Retry operations in tape systems are very slow because the tape must be stopped rather than continuing at high speed. The beginning of the missed record must be searched for and found, and the tape must then regain speed to read or write the record a second time. Placing a large data buffer on the tape adapter itself would, at best, reduce the probability of a retry operation. Therefore the highest priority DMA channel had to be assigned to the task.

Tape backup of larger files in AT systems became advisable because the AT was the first system that defined concurrent operations, albeit for just two adapters. Those adapters were the PC network and the hard disk adapters – exactly the two devices that should run concurrently in a minimal file server.

To handle concurrent operation, the more powerful 80286 processor moved the fixed disk data, while the DMA controller supported the PC network. Today, faster network cards such as Token Ring and Ethernet use the processor rather than DMA. This has increased the demands on the processor in PC/XT/AT systems to the point where very fast processors and associated fast memory systems are used, at great expense, largely to support high-speed data transfer through the processor.

DMA channel 3 is still permanently reserved for PC network cards, because they may still be installed in such systems. No other adapter card design should use channel 3 because, as explained previously, it might damage itself or the drivers on another card. This is one of the major penalties a system architecture must pay for AT card compatibility: the number of PC/XT/AT DMA adapter card designs that can exist without causing a system integrity problem or requiring much higher skills for installers is limited to the number of DMA channels in the system.

As a consequence, compared to the number of programmed I/O adapters for PC/XT/AT systems, very few DMA adapters exist beyond the few listed here. Some DMA adapters have been defined to use the "allocated" DMA channels, but they are typically installed in custom system configurations with knowledge of the potential conflicts and system integrity problems. Users of any new DMA card designs must be aware of the DMA resources already used in any system before installing such cards, or risk permanent damage to the system unit or cards.

Thus, technical constraints limit the market for many more DMA adapter cards for PC/XT/AT systems. These constraints, however, have not deterred the marketing of new adapters that use DMA.

DMA Expansion in the PC AT

When the IBM PC AT was announced in 1984, it defined three more DMA channels that could double data-transfer throughput by moving 16-bit words, rather than just one 8-bit byte, at a time. Plans were made to enable DOS to support addressing blocks of words set on boundaries that were multiples of two bytes. The two-byte boundary restriction occurred because the AT incremented (by two bytes at a time) the address that the DMA controller used to select data in memory. The DMA controller had to start transfers on a memory address that was a multiple of 2, end transfers at an address that was a multiple of 2, and move only even numbers of bytes in each block transfer.

The plans for DOS – and for using the three new DMA channels – never materialized. DOS was unable to ensure that data buffers in memory would always reside on word boundaries. In fact, DOS would most likely organize buffers so that data would begin or end on odd boundaries. Any read from memory by a 16-bit DMA channel may miss the first or last byte of a buffer, or get an extra byte from an adjacent buffer. Even worse, write operations by a 16-bit channel could destroy a byte of data in an adjacent buffer by overwriting it. For this reason, DMA channels 5, 6, and 7 have seldom been used to move data in PC AT systems.

The DMA channel utilization at the end of 1984 is summarized in Figure 2.

DMA Channel	PC or PC XT Use	PC AT Use	Width
0	DRAM Refresh	Tape Backup	Byte
1	SDLC	SDLC	Byte
2	Diskette	Diskette	Byte
3	Fixed Disk or PC Network	PC Network	Byte
4		Cascade[1]	Word
5, 6, 7		Not compatible with DOS and unused except for busmaster upgrade	Word

[1] When channels 5, 6, and 7 were added for AT systems, DMA channel 4 was consumed as the result of cascading one DMA controller into the input of another.

Figure 2. DMA Channel Utilization in IBM PC, XT, and AT

The restrictions in Figure 2 apply to all PC/XT/AT compatible systems and in EISA systems that accept PC/XT/AT cards. Plugging an adapter card into an EISA system that already has a card using the same DMA channel can damage the card or other cards in the system. The purported advantage of EISA – its ability to accept PC/XT/AT adapter cards – represents an exposure to costly problems.

Micro Channel: A Fresh Start

In 1987, IBM announced a fresh start called Micro Channel architecture. A fresh start was necessary to discard the legacy of limitations developed by PC/XT/AT cards and because the demands of even the near future could not be met by modifications of the past.

What kinds of demands? Multitasking, multi-user systems typically have more I/O devices than PC/XT/AT systems can configure. The I/O demands of adapters and devices are advancing faster than a single processor can support. Multimedia, in particular, requires much higher throughput than Industry Standard Architecture (ISA) systems can handle. The demands of the "multi" environment are also much greater than EISA systems with ISA cards installed may be able to support (see Part 1 of this article).

Although the need for an advanced bus was initially satisfied by the Micro Channel architecture, a consortium that formed 18 months later endorsed that need with a competing proposal. This consortium developed the EISA design specification, which cloned some of the functionality already available in Micro Channel architecture. Apple's NuBus architecture, Micro Channel, and EISA defined auto-configuration, interrupt sharing, higher throughput, and busmaster capabilities – all requirements of multitasking, multi-user, or multimedia systems.

Figure 3 shows that EISA systems, which accept PC/XT/AT cards, have only DMA channels 5, 6, and 7 available for expansion. A single DMA serial/parallel adapter card can then consume all the DMA resource in the EISA system. EISA is a 32-bit-only architecture, so its busmaster adapter designs are 32-bit only, and a maximum of eight can be installed.

In contrast, Micro Channel PS/2 systems can accommodate 15 DMA adapters, 15 busmaster adapters, or any combination totaling 15 (16 in some multiple bus systems). A Micro Channel system could theoretically have 4, perhaps 5, DMA serial/parallel adapters (each such adapter uses three DMA channels), where **EISA systems are limited to one** because there are only three spare DMA channels, as shown in Figure 3. This gives Micro Channel greater expandability to create balanced processor and I/O expansion capabilities over the life of the system.

Arbitration Schemes

EISA systems centrally arbitrate (prioritize) requests by busmasters and DMA adapters. Each slot in an EISA system has a request signal to notify the central arbitration bus that a busmaster or adapter card in that slot is requesting control of the system. All requests are given the same priority level with a scheme called *strict fairness* or *rotating priority*, because each adapter or busmaster has the same chance to control the system. This arbitration scheme defines a minimum of seven DMA request signals, seven DMA acknowledge signals, eight busmaster request signals, eight busmaster acknowledge signals, and one signal for PC/XT/AT-compatible DMA called Address Enable (AEN). This requires 31 signals on the bus.

Micro Channel systems, on the other hand, distribute the arbitration function. They use an encoded request-and-acknowledge structure where

Priority (Arbitration) Level	EISA			Micro Channel Architecture with PC Software Compatibility		
	Use	Reservation	Width	Use	Reservation	Width
0	DMA	Tape Backup	Byte [1]	DMA/BM	Available	8/16/32-bit
1	DMA	SDLC Comm	Byte [1]	DMA/BM	Available	8/16/32-bit
2	DMA	Diskette	Byte [1]	DMA/BM	Diskette	8/16/32-bit
3	DMA	DMA Network	Byte [1]	DMA/BM	Available	8/16/32-bit
4	DMA	Optionally implemented		DMA/BM	Available	8/16/32-bit
5	DMA	Available	8/16/32 bit	DMA/BM	Available	8/16/32-bit
6	DMA	Available	8/16/32 bit	DMA/BM	Available	8/16/32-bit
7	DMA	Available	8/16/32 bit	DMA/BM	Available	8/16/32-bit
8	BM	Available	32-bit only	DMA/BM	Available	8/16/32-bit
9	BM	Available	32-bit only	DMA/BM	Available	8/16/32-bit
A	BM	Available	32-bit only	DMA/BM	Available	8/16/32-bit
B	BM	Available	32-bit only	DMA/BM	Available	8/16/32-bit
C	BM	Available	32-bit only	DMA/BM	Available	8/16/32-bit
D	BM	Available	32-bit only	DMA/BM	Available	8/16/32-bit
E	BM	Available	32-bit only	DMA/BM	Available	8/16/32-bit
F	BM	Available	32-bit only	DMA/BM	Available	8/16/32-bit

[1] Shown with PC/XT/AT portability fixed assignments on DMA channels 0 to 3

BM = Busmaster

Note: In some single-bus Micro Channel designs, the processor is a busmaster at priority (arbitration) level F.

Figure 3. Configurability of Arbitrating Devices in Micro Channel and EISA Systems

arbitration is done between the adapters, but coordinated centrally. In a Micro Channel system, an adapter card or busmaster issues a request using a signal called "Preempt" and places a 4-bit prioritized value on an arbitration bus. If a requesting card detects that another arbitrating adapter has a higher priority value on the arbitration bus than its own bid, the requesting card removes its bid but retains the preempt signal. In a few billionths of a second the auction is over, and the winner is the only card whose bid remains active. A line called "Arbitrate/Grant" signals the end of the arbitration bidding auction. This requires a total of only six signals on the bus.

The number of signals used to communicate the arbitration mechanism is an important consideration in Large Scale Integrated (LSI) circuit designs used in Micro Channel and EISA adapters. Micro Channel architecture requires fewer lines, making it less expensive to implement in LSI – both on the adapter and the system board. The superior LSI affinity of Micro Channel now helps a modern generation of card development produce less costly LSI designs.

Card Size: EISA advocates have claimed an advantage due to the greater space on EISA cards making it possible to implement more function on a card. EISA needs more space than Micro Channel on some cards to implement a function. In addition, many Micro Channel systems now support cards of the same size.

Multiple Priorities per Card: EISA systems provide only one busmaster request-and-acknowledge signal to each card slot. This makes it impossible to place multiple busmasters on a card at different system-priority levels. Micro Channel can package multiple busmasters per card, with each at a separate priority level.

This is important in applications such as serial communications. In full-duplex mode, data is transmitted and received at the same time; for example, a keyboard transmits data into a system while a display receives characters from that system. Many terminals doing very high-speed duplex serial operations make efficient use of dedicated lines to remote mainframe computers, or between departments or regional centers. A busmaster or DMA interface that handles both sending and receiving would be ideal in the server that drives these lines. Whereas outbound data can wait until the transmitting end is ready to operate, data being received must be dealt with immediately; otherwise data may overflow available buffers. This means that the inbound channel must have much higher priority than the outbound channel. Micro Channel systems can accommodate these different priorities on a single card or busmaster; **EISA systems cannot**.

Programmable Fairness: Micro Channel supports *arbitration fairness* as a default mode for all adapters in the system, but also allows *programmable fairness* when adapters, such as full-duplex serial communication adapters, have one function that requires much greater priority than the other. Multiple priorities are needed on tape controller cards, where the length of an outbound record is known and can be written slowly to a buffer, but the inbound records are of unknown length and may exceed a buffer. This imbalance requires higher priority inbound and lower priority outbound. Network adapters, whose performance is higher than that of serial communication adapters, likewise have different priorities for inbound and outbound transfer. Micro Channel has been designed to accommodate situations of disproportionate priorities on the same card; **EISA's design cannot**. This means that Micro Channel cards can apportion system priorities where they are needed rather than forcing all cards to a lowest common priority.

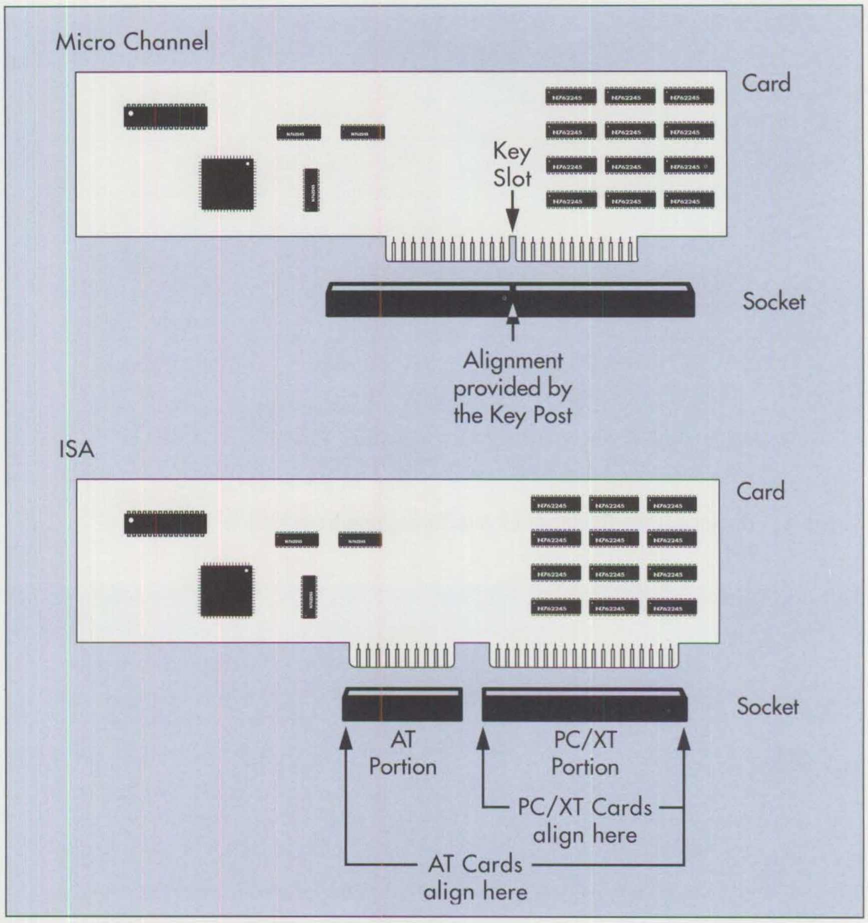


Figure 4. Mechanical Alignment of Adapters in Micro Channel and ISA Systems

Bus Throughput

The mechanical design of the connector used in Micro Channel systems may not seem significant when comparing the architectures of Micro Channel and EISA, but it is. As shown in Figure 4, a PC or XT card plugs into an ISA slot and aligns against the end bulkheads of the PC/XT portion of the ISA system socket. Likewise, an AT or EISA card aligns against the end bulkheads of the full EISA socket. As a consequence, the EISA socket can never be wider than the card, because alignment is from the ends of the card. Because all signal tabs are dedicated, it is very difficult, perhaps impossible, to add signals.

In contrast, a Micro Channel-based system can expand to a wider data bus by expanding its socket length to add more signal tabs (accommodating a bus width of 256 bits). In fact, IBM is already using such a connector – it is the connector that the processor upgrade cards plug into PS/2 Model 90 and 95 systems. Using no more than the 160 MB per second protocols (already defined and shown publicly at BUSCON West 92), expanded to a width of 256 bits, a peak throughput of 640 MB per second can be attained. This is not the only option available to extend Micro Channel throughput; it is simply one of the more easily explained options. It shows that there is already at least one way that will support existing 64-bit cards.

Bus throughput is important because it determines the ability of a system

Component	1981	1991	Growth Ratio
Processor Execution Rate	0.3 MIPS	15 MIPS	50:1
Display	64 KB/second for still image	1 MB/second for still image	16:1
		30 MB/second for motion	480:1 16:1 with 30x "lossless data compression"
Fixed Disk Speed	0.625 MB/second (ST-506)	5.0 MB/second (SCSI)	8:1 single file 64:1 file array
Communication	300 bits/second (asynchronous)	100 Mbits/second (FDDI LAN)	333:1

Figure 5. The Increasing Demands on Bus Throughput from 1981 to 1991

Parity	Data	Description
1	0 1 0 1 0 1 0 1	This is the 9-bit value (data bits plus parity bit) transmitted. If it is received like this, the receiver is assured that no single error occurred in transmission.

0
1
1
	0 1 0 1 0 1 0 1
	0 0 0 1 0 1 0 1
	0 1 1 1 0 1 0 1
		If any of these 9-bit values are received, then an error has been detected. The error may have happened because the parity bit itself is invalid.

1	0 0 0 0 0 1 0 1	If any even number of errors occurs at once, they cannot be seen from the context. Fortunately, multiple errors in one byte are very rare compared to single errors. Odd numbers of errors are detected like single errors.
1	1 1 1 1 1 1 1 1	The total number of 1's in the data and parity together is always an odd number for odd parity.

Figure 6. Examples of Data Parity

to keep up with the growing demands of I/O. Historically, the performance of I/O devices has grown faster than the capacity of processors to process information. Growth comparisons between 1981 and 1991 are shown in Figure 5.

I/O requirements are growing so fast that even 640 MB per second may be inadequate within a decade. IBM's demonstrated capability to support 160 MB per second now, and four times that without redesign, gives confidence that Micro Channel can meet that challenge. On the other hand, EISA's maximum is 33 MB per second (see Part 1 of this article), with no way to widen the data path. Thus, Micro Channel can be easily expanded for more throughput or advanced functions, where apparently **EISA cannot**.

Ensuring Data Integrity at High Speeds

There is little value to have data traveling at high speed if there are no brakes to avoid catastrophe – the loss of data integrity due to error. Fortunately, Micro Channel design includes error-detection and response protocols called *bus parity* and *synchronous channel check* that go beyond the simple mechanism defined for the PC and inherited by EISA.

Parity is a concept inherited from mainframe and minicomputer systems. Parity is the context that is associated with the data to indicate its validity. A parity bit is transmitted with the data so that the receiving device can check both parity and data for validity. With odd parity, if the total count of 1's in a byte is an even number, the parity bit is a 1; if the total count is an odd number, the parity bit is a 0. Examples of parity are shown in Figure 6.

Bus parity is not foolproof; multiple errors, while unlikely, can get through. Although total elimination of error is impossible, the probability of undetected errors can be greatly reduced. The objective is to increase the integrity of the system to the highest possible degree. Parity simply requires an extra pin per byte of data on the bus. Parity is one data integrity tool available in Micro Channel systems, because extra pins are already defined on the Micro Channel interface. Extra pins, and therefore parity, do not exist on the EISA bus. Thus, Micro Channel defines error-detection capabilities that **EISA cannot**.

Most of today's PCs and PS/2s do not process parity, but newer Micro Channel-based systems in the IBM family of products do process parity information along with the data. The workstation products in the IBM RISC System/6000™ family and the IBM mainframe 7437 and ES/9000™ systems use parity to check data. As personal computer products provide more memory and hard disks, processors that perform parity checking on data transfers will likely become commonplace.

Asynchronous and Synchronous Channel Checks

Using parity, a protocol in PC/XT/AT systems that was inherited by Micro Channel and EISA, called *asynchronous channel check*, can indicate that an error has occurred somewhere within a block of data. Recovery involves retransmitting or rereading the entire block.

The need for synchronous channel check arises when the error must be identified rapidly and with great precision. For example, what if the data error occurs when instructions are fetched from various parts of memory, or as part of a long train of data moving at high speed? Does the processor have to be halted or must the entire operation be repeated? Efficient recovery becomes possible if the individual erroneous byte can be identified. Then perhaps the process can be recovered by restarting just before the error, or by retransmitting only the error in a stream of data. To do this, detection of an error must be synchronized with the receipt of data. As part of its legacy from mainframes, Micro Channel inherited synchronous channel check, which precisely identifies the failing byte within a transfer. PC/XT/AT and EISA systems, having only asynchronous channel checking, can only isolate errors to the block. They require retransmission of entire records, making error correction less efficient. EISA adapter cards do not have the pins required to implement the signals for synchronous channel checking.

Summary

In Parts 1 and 2 of this article, the abilities of the Micro Channel and EISA architectures to meet meaningful user objectives have been compared. Part 1 shows Micro Channel to be an open and consistent standard, whereas the foundation of EISA – Industry Standard Architecture – is inconsistent between the documentation and real-world implementations. The ability of the architectures to be applied to a broad spectrum of applications and system designs has been contrasted. In addition, the ability of the architectures to grow to accommodate advancing system needs for I/O performance and integrity has been outlined. The objective of Micro Channel architecture to migrate mainframe architecture to advanced microcomputer systems has been shown to be superior to EISA's short-term objective of "Extending personal computer architecture to advanced microsystems." Part 1 shows that EISA's affinity for a single family of processors and the operating systems that support these processors effectively make EISA a "proprietary" solution.

More than 1,200 cards are available from various Micro Channel manufacturers. There are approximately 100 members of the Micro Channel Developers Association, including many system vendors in addition to IBM.

Part 1 shows Micro Channel to be technically superior in its ability to offer advanced automatic setup of the system as a whole, regardless of feature card configuration. Micro Channel architecture's interrupt sharing capability is uncompromised by system integrity exceptions, unlike EISA. High-speed data transfer in Micro Channel systems is accomplished by existing products, implemented at very high throughput, whereas EISA systems' high-speed modes are subject to interference from installation of PC/XT/AT cards. Part 2 shows that direct memory access, and busmaster configurability and flexibility, are far superior in Micro Channel architecture compared to EISA. Finally, the data integrity features needed in advanced systems are defined in Micro Channel, yet are nearly absent in EISA. Not one function in EISA is absent in Micro Channel, but examples of the converse abound.

EISA is left with only one advantage over Micro Channel: the ability to install existing PC/XT/AT cards. Yet the advanced functions (that mimic Micro Channel) expected from the architecture are lost when PC/XT/AT cards are plugged into an EISA system. EISA can only come close to Micro Channel when all installed cards are EISA cards. Even then, the legacy that EISA inherits from compatibility with the PC, XT, and AT makes the comparison heavily favor the Micro Channel architecture.

References

- *EISA Technical Reference Guide*, 2nd Edition. Compaq Computer Corporation, 1990. (130584-002)
- *IBM Personal System/2 Hardware Interface Technical Reference* (S84F-9808)

***Chet Heath** is a Senior Technical Staff Member and engineer at IBM's Entry Systems Technology Laboratory, in his twenty-second year with IBM. He holds a BS in electrical engineering from the New Jersey Institute of Technology (Rutgers) and an MS in electrical engineering from the LSI Institute at the University of Vermont. Chet was the primary innovator of the PS/2's Micro Channel architecture and has led its definition since inception. He has received numerous awards for his work on Micro Channel architecture including IBM's eighth-level invention award, Outstanding Technical Achievement, Outstanding Innovation, and Corporate Technical Achievement awards. He was named one of the ten most influential people on PCs in the '80s by* PC User *magazine in the U. K.*

Synergy by Design

Chet Heath
IBM Corporation
Boca Raton, Florida

This article shows how the synergy between Micro Channel architecture and Operating System/2® 2.0 produces a demonstrable benefit for users of PS/2 Micro Channel systems. In the everyday function of printing, there is dramatic improvement in the time needed to transfer print files, and dramatic improvement in process efficiency, when OS/2 2.0 and Micro Channel architecture are used together.

The article discusses an experiment and gives instructions for conducting it yourself. There is no magic involved other than that which is designed into OS/2 2.0, IBM Advanced BIOS (ABIOS), and Micro Channel architecture. Everything is either right off the shelf – in announced IBM hardware and software – or printed in this article.

The experiment discussed in this article shows the benefits and value of the synergism of IBM's PS/2 Micro Channel architecture, Direct Memory Access (DMA), IBM's Advanced BIOS (ABIOS), and OS/2 2.0 operating in concert. This experiment undertakes a common function: printing. In the experiment, a batch program copies a large file from diskette to fixed disk, prints the file from the fixed disk to a high-speed laser printer attached to the parallel port LPT1, and displays the start/stop times for printing the file. The experiment is conducted in two identical parts, with each part using a different operating environment. The first part runs under DOS; the second runs under OS/2 2.0 on a PS/2 Micro Channel system with ABIOS and an enabled DMA parallel printer port. The object is to demonstrate the significant reduction in elapsed printing time under the latter environment. The significant difference in the elapsed times is especially noticeable when both parts of the experiment are started simultaneously on two side-by-side computers.

The Components

The computer used in the experiment is an IBM Personal System/2 Model 57 SX, which contains an Intel 80386 SX processor running at 20 MHz and a hard disk large enough to hold the data to be printed.

The data to be printed is in a file of about 200,000 bytes – roughly 40 pages of graphics-mode or spreadsheet text. The data consists of null characters (00 hex), chosen because null characters do not actually print, so they use no paper.

The time needed to transfer data to a printer depends on the performance of the printer's computer interface. The printer used in the experiment is an IBM 4029 LaserPrinter Model 5E running in Fast Bytes transfer mode. This mode, which is selectable from the front panel of the 4029, is designed for high-speed data transfer. In Fast Bytes mode, the 4029's internal microprocessor gives priority to the printer's computer interface while its print buffer is being loaded. The result is that the printer accepts the transferred data faster, and the experiment concludes sooner. Fast Bytes mode does not skew the experiment; it has the same effect on both parts of the experiment.

Disabling Print Spoolers

To ensure that the experiment measures only the performance of the printing function, any print spoolers installed in DOS and OS/2 must be disabled. Print spoolers simply send data elsewhere in memory; they do not actually perform the I/O, but they signal that the I/O is complete before it is done. Print spoolers can cause inaccurate results in the concurrent I/O test, so they must be disabled. Terminating print spoolers in a DOS environment requires removing them from the `AUTOEXEC.BAT` file, then rebooting the system. Alternately, the system may boot DOS from diskette. The instructions for disabling the OS/2 2.0 print spooler are given in Figure 1.

Part 1: DOS and Programmed I/O

DOS uses the system's microprocessor to move data between memory and I/O ports according to programmed instructions. This is called Programmed Input/Output (PIO). In Part 1 of this experiment, the print file transfer takes place in PIO mode under DOS.

Programmed I/O printing is typical in PC/XT/AT systems. Print data transfer in PC/XT/AT systems takes place in two sequential steps. The steps are sequential because both use a common element: the processor. In step 1, the data moves from the hard disk, through registers in the processor, to the system memory. In step 2, the data moves from system memory, back through the processor, to the printer port. The function of DOS is to sequentially schedule (that is, program) these steps.

The PC's Basic Input/Output System (BIOS), resident in Read-Only Mem-

How to Disable the OS/2 2.0 Print Spooler

1. Boot OS/2 2.0 from diskette.
2. Double-click the left mouse button on the OS/2 System icon.
3. Double-click the right mouse button on the System Setup icon.
4. Click the right mouse button on the Spooler icon.
5. If the displayed context menu shows the Disable Spooler option, the spooler is currently enabled. Select that option, then click the left mouse button. A pop-up message will remind you to reboot the system to disable the spooler.
6. Click the left mouse button on the OK icon.
7. Press Ctrl-Alt-Del to reboot.
8. Repeat steps 2 through 5. The context menu in step 5 should now display the Enable Spooler option.

Figure 1. Disabling the OS/2 2.0 Print Spooler

ory (ROM), operates the printer by reading and writing the three registers, shown in Figure 2, that comprise the printer port. The first register sends commands, such as "accept the data," to the printer; the second register provides a path for the data; the third register accepts status reports such as "printer busy" or "need paper."

PIO is an interrupt mechanism for servicing print requests. In PIO mode, data is transferred to the printer port – one byte at a time. When the processor finishes delivering a byte to the printer interface, the printer signals an interrupt to request another byte. Therefore, the processor receives an interrupt for each byte it transfers to the parallel printer port. Each interrupt takes about 200 to 300 instructions, or the software may poll the parallel port interface using as many (or more) instructions, when DOS uses the PC's BIOS[1]. Because there are so many instructions associated with each byte being transferred, and because there are so many bytes to print, the processor can be heavily loaded just servicing printer interrupts. In most PC/XT/AT systems, the processor is additionally burdened with operating the hard disk that is the source of the data being printed. Therefore, the performance of the processor limits the speed of the data-transfer operation. The total

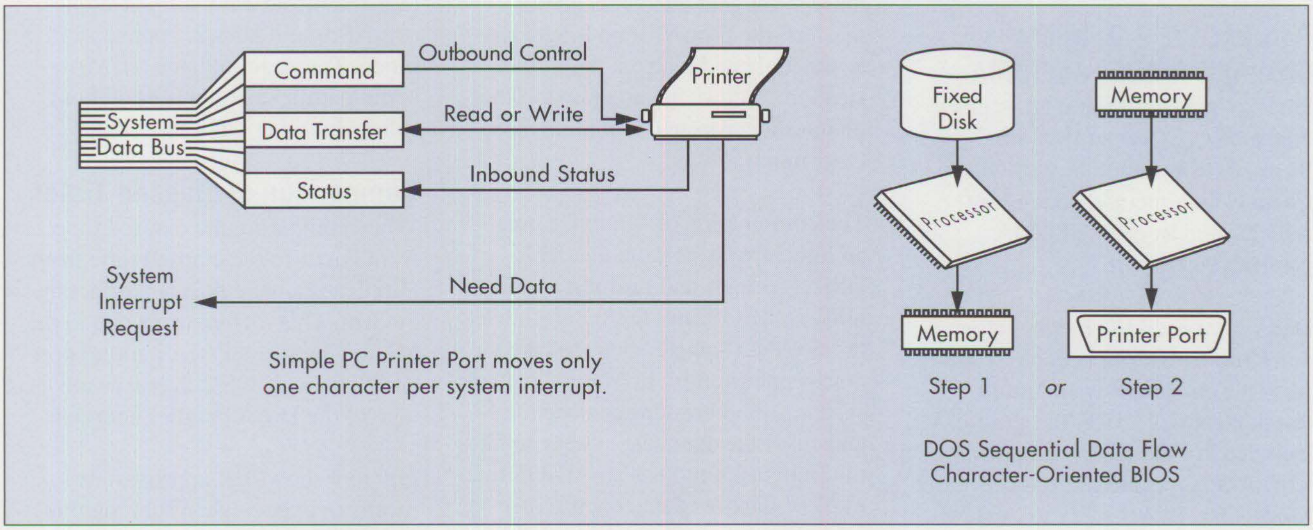


Figure 2. Simple PC Printer Port Operation

[1] Even when OEM BIOS poll the printer interface, the character-oriented BIOS of ISA machines still involves a software interrupt per character at the operating system interface.
Running the print operations in the background in a multitasking operating system will demonstrate that ISA systems consume cycles from the foreground process while PS/2 Micro Channel DMA does not. An impressive demo comparing fully installed OS/2 2.0 operating on ISA and PS/2 Micro Channel systems can be made where little residual performance is provided to applications in an ISA system concurrently operating with high-speed background printing, as in a print server. A similarly configured PS/2 system will handle almost any level of background printing with little noticeable degradation of concurrent application performance.

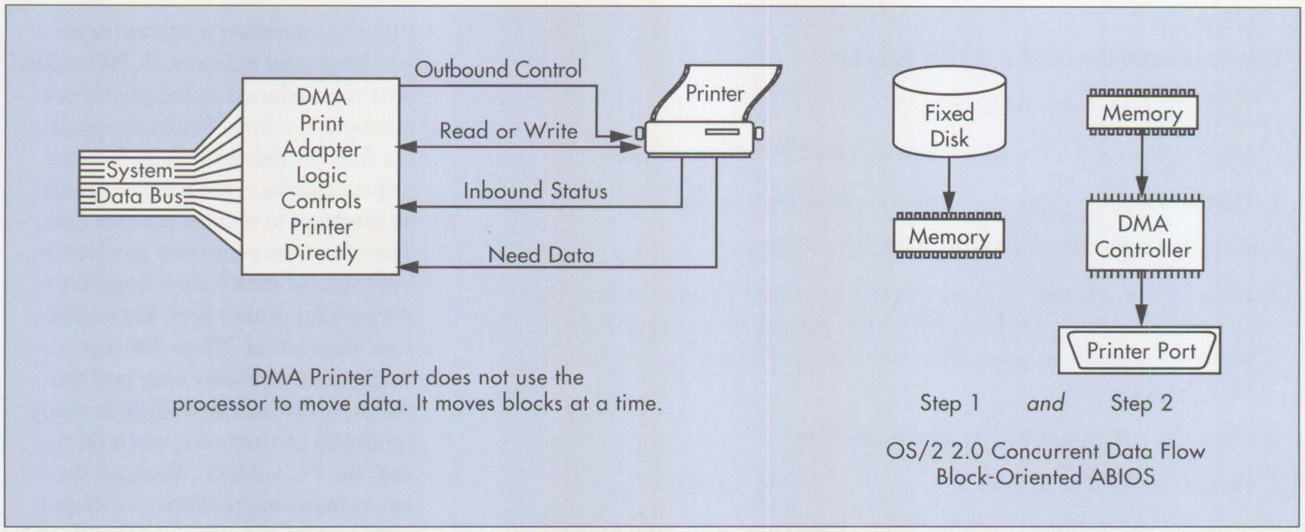

DMA Print Adapter Logic Controls Printer Directly
System Data Bus
Outbound Control
Read or Write
Inbound Status
Need Data
Printer
DMA Printer Port does not use the processor to move data. It moves blocks at a time.
Fixed Disk
Memory
Memory
DMA Controller
Printer Port
Step 1 and Step 2
OS/2 2.0 Concurrent Data Flow Block-Oriented ABIOS

Figure 3. DMA Printer Port Operation

time to complete the operation is the sum of all file and print transfer delays, because they are done sequentially.

IBM PS/2 Micro Channel systems provide PIO mode for compatibility with DOS and Microsoft Windows® applications.

Part 2: OS/2 2.0, Micro Channel, ABIOS, and DMA

Part 2 of the experiment takes place under OS/2 2.0 on an IBM PS/2 Model 57 SX computer with Micro Channel architecture, concurrent ABIOS, and an enabled DMA parallel printer port.

DMA print is an I/O mode that is much more efficient because it supports the concurrency and multitasking of OS/2 2.0. DMA print is included in PS/2 Micro Channel system models recently introduced: PS/2 Models 56, 57, 80-A21, 80-A31, 90, and 95.

DMA uses "intelligent" I/O adapters that off-load much of the processor's burden by using the DMA controller, instead of the processor, to move data (as shown in Figure 3). The DMA parallel adapter moves data from a previously loaded memory buffer, through the DMA controller, to the printer port, bypassing the processor. The SCSI file adapter is a Micro Channel busmaster that controls the fixed disk adapter and moves data from the hard disk directly to a memory buffer in one operation, also bypassing the processor. Because of these intelligent adapters and busmasters, the processor no longer has to schedule and manage the I/O; instead, it can turn its attention to other computing tasks and complete those tasks much sooner.

The combination of Micro Channel architecture (with more available DMA channels and busmaster capability), ABIOS, and OS/2 2.0 enables the two data-transfer steps to take place concurrently. In Micro Channel systems, the two steps can be simultaneous if the operating system defines multiple tasks and has an ABIOS that moves blocks of data concurrently using busmasters and DMA peripheral adapters. OS/2 2.0 has been designed to fulfill these requirements.

If the DMA parallel interface is enabled by Programmable Option Select (POS) during configuration of a Micro Channel system, the DMA interface is then selected by ABIOS and OS/2. However, if the DMA parallel interface is disabled or not present in a Micro Channel system, OS/2 defaults to the DOS-compatible mode, PIO, which has been implemented in Micro Channel systems for compatibility with DOS applications. An IBM Personal Computer AT system and most (if not all) AT clones, even if running OS/2, operate in PIO mode. They cannot have a DMA serial/parallel adapter port without ruling out other devices.

Comparison of Elapsed Times

When comparing the elapsed time from Part 1 to the elapsed time from Part 2, a dramatic difference becomes evident. That difference is due to the Micro Channel's DMA parallel port and to the way OS/2 2.0 takes advantage of the DMA parallel interface.

Figure 4 shows the approximate results to expect when running this experiment on various Micro Channel computers. As seen in Figure 3, even the most powerful processor available today – a 486 DX running at 50 MHz – takes almost twice as long under DOS and PIO as it takes on any system listed in the figure that is running OS/2 and DMA printing. Clearly, using Micro Channel systems with DMA, ABIOS, and OS/2 2.0 means *performance*.

Add Real Life...

When everyday tasks are added to the system's workload, the synergy among Micro Channel architecture, ABIOS, and OS/2 2.0 shows a more dramatic improvement.

In a computer that has other tasks to perform, Terminate-and-Stay-Resident (TSR) utility programs are usually present under DOS. With common TSR utilities loaded for disk cache, LAN requester, and 3270 emulation, the interrupt service times for PIO increase significantly. This is because DOS's primitive ability to support background tasks in TSR programs depends on having each TSR execute for a brief period following every timer tick interrupt. The timer tick has higher priority than both print and serial communications, so it gets first access to the processor's attention. This adds to the delay in servicing the timer tick interrupt and all lower priority interrupts, and places an interrupt latency burden on any system in which I/O performance depends on the processor. In real-life situations, DOS TSRs take more of the processor's capability and give the processor less time to attend to applications such as printing. Further, TSRs reserve part – sometimes as much as half – of the 640 KB of system memory available to DOS.

In contrast, in a PS/2 Micro Channel system running OS/2 2.0, the DMA controller and the DMA print function do not depend on the processor to move data, so they are essentially unaffected by processor activity.

In one test on an unmodified PS/2 Model 57 SX with the common TSRs loaded, Part 1 under DOS and PIO took about four minutes. Part 2, under OS/2 and DMA (print to an IBM 4029 LaserPrinter Model 5E) remained at about 12 seconds. In this real-life situation, the result is a 20:1 improvement.

Processor	DOS with PIO[1]	Micro Channel with DMA, ABIOS, OS/2 2.0[2]
386 SX at 20 MHz	61 seconds	< 12 seconds
386 SLC at 20 MHz	39	< 12
386 DX at 20 MHz	58	< 12
386 DX at 25 MHz	49	< 12
486 DX at 25 MHz	38	< 12
486 DX at 33 MHz	29	< 12
486 DX at 50 MHz	22	< 12

[1] This print experiment should not be construed as a benchmark for the comparison of programmed I/O performance between systems, but as a comparison of the efficiency of Direct Memory Access and PIO transfers.

[2] This time includes file seek and any inaccuracies due to coarse resolution of the real-time clock for short intervals.

Figure 4. Results on Various Micro Channel Systems with a 4029 LaserPrinter

Analysis

When both parts of this experiment were performed on a 20 MHz 386 SX-based Micro Channel system, the printing task took 55 seconds longer under DOS and PIO than under OS/2 and DMA. A 20 MHz 386 SX computer executes approximately three million instructions per second (3 MIPS). Therefore, the savings is 3 million instructions/second x 55 seconds = 165 million instructions. By off-loading the print I/O to the DMA controller under OS/2, the processor can do other useful work simultaneously, resulting in higher productivity and throughput.

The 16-bit DMA parallel printer port operates at the same speed on all Micro Channel systems, because its speed is not significantly affected by the efficiency of the processor. The speed of the PIO parallel port, however, depends on the processor's efficiency in executing instructions. Therefore, the performance improvement with DMA print is most significant in systems whose processors have comparatively lower performance and therefore take proportionately longer to transfer the print data. This demonstrates that Micro Channel has value across the whole spectrum of systems from high-end servers to entry-level desktop computers.

PS/2 Micro Channel Value to the User

It is not unusual to print a spreadsheet or document and then move on to the next task. If the processor moves the data using PIO, the delay before starting the next task can be long enough to impact the user's productivity. Operating systems that use the 32-bit and Virtual 86 modes of Intel 80386 and higher processors (for example, Windows 3.0, OS/2 2.0, and AIX® 386) may use 1,000 or more instructions to respond to an interrupt. Why so many? Recall that PC/XT/AT print adapters can move only one character per interrupt. For each interrupt, the processor must save the state of its registers and other system statuses. The advanced operating systems use many more registers and have far more complex system statuses; these account for the increase from 200 to 1,000 instructions per interrupt.

Let us look at how this affects the printing of very large files. Typically, printed text may contain 2,000 characters per page, including much blank space. A page produced in graphics mode by desktop publishing or spreadsheet software may print in every position on the page; in this case, 5,000 characters per page is not unreasonable. (If we assume that pages have 5,000 characters, the 200,000-byte file transferred to the printer port in this experiment can be considered as 40 pages of graphic or spreadsheet output.)

Therefore, the transfer of the print data to the printer port using PIO and advanced 386 modes consumes 5,000 bytes/page x 1 interrupt/byte x 1,000 instructions/interrupt = 5 million instructions/page. If a printer could print one page per second – a reasonable expectation for laser printers in the foreseeable future – the calculation becomes 5 million instructions/ page x 1 page/second = 5 million instructions/second, or 5 MIPS of computer power sapped from the system just to keep up with the printer. This is more computing power than is found in most 386 SX-based systems. As a result, the print speed could be limited by the capabilities of the system processor. Even if the printer were fully buffered, the same number of processor instructions would be necessary to move the same size block of print data. *Net*: The print performance would degrade, or concurrent applications using the processor would be impacted, or both.

As noted above, print performance degrades further when the computer is also running TSR utilities that perform background operations, such as 3270 operation or LAN attachment. In this case, the number of processor cycles available to print operations is further reduced. On a DOS/PIO system burdened by TSRs, the 200 KB file that was printed in this experiment took several minutes to print. (This leads to a strong case for preemptive multitasking, as in OS/2 2.0 – but that is a topic for another article.) The performance does not degrade, however, in PS/2 Micro Channel systems that can delegate processor tasks to autonomous subsystems such as DMA print. For example, the DMA print function on Micro Channel systems absorbs 96% of the printer management work formerly done by the processor. The SCSI busmaster adapter further relieves the processor. Very little system overhead remains to be done by the processor, so that it can concentrate on executing operating system and application instructions. Therefore, much more of the processor's power is available for program execution.

The DMA print function on Micro Channel systems absorbs 96% of the printer management work formerly done by the processor.

In desktop personal systems, to equal the reserve processing power of a 20 MHz, 386 SX-based Micro Channel system with ABIOS and DMA print, a computer that does not have ABIOS and DMA print would perhaps need an additional 5 MIPS of processing capacity with a very high-speed printer. That level of performance is typical of 486 systems operating at 25 MHz or higher.

As a server, a file server system cannot afford to be impacted by adding the print serving function (which includes moving the data among hard disk, memory, and printer port) on top of the file server overhead in the typical PC/XT/AT system. Yet, for economic reasons, high-speed printers are often attached to systems that are shared among several network clients. File server performance degrades noticeably when a fast, fully buffered laser printer is attached to the network's file server system. Consequently, serving data to high-speed printers is usually delegated to a separate print server system.

The DMA print function, on the other hand, operates without significant impact on the processor, so the same server system can be used for both printing and network serving. This eliminates the costs for a second server, its server license, lifetime hardware and software maintenance costs, and several megawatt hours of energy. A cost saving of this magnitude easily justifies investment in an IBM PS/2 Micro Channel system with DMA print capability.

Other DMA Channels Required

The seven PS/2 Micro Channel systems shown in Figure 4 also contain a DMA serial interface that can operate up to 345K baud – 10 to 20 times the capabilities of today's fastest systems. The DMA serial port uses two more DMA channels, one for read and one for write operations. Thus, three DMA channels are needed to support the DMA serial/parallel adapter. However, PC/XT/AT bus systems do not typically have three spare DMA channels. Only four DMA channels in PC/XT/AT systems are DOS-compatible. Those channels are already assigned to tape backup, Synchronous Data Link Control (SDLC), diskette, and network DMA interfaces by hard-wired connections on typical PC/XT/AT card designs. Consequently, it would be very difficult to retrofit DMA serial/parallel capability into cards designed for most PC/XT/AT systems. An EISA system may be able to configure one DMA serial/parallel combination, but the EISA system does not have enough available DMA channels for additional DMA serial/parallel ports. (For details, see the article "Comparing Architectures: Micro Channel and EISA," parts 1 and 2, in the April

1992 issue and in this issue of *Personal Systems Technical Solutions*.)

The SCSI file adapter is also on the system board and uses another priority level. A total of 15 DMA adapters or busmasters can be added to PS/2 Micro Channel systems, giving the Micro Channel the flexibility to accommodate future growth.

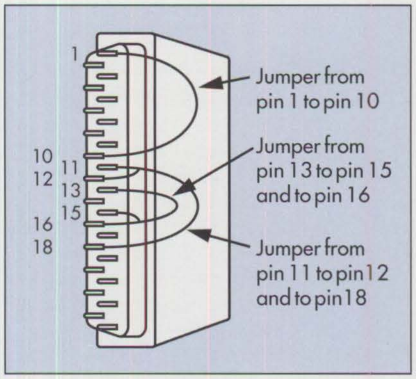
Making This Experiment Portable

The experiment discussed in this article can be put into a package small enough to carry in your pocket, purse, or briefcase. There are two good reasons to do this: the experiment is then portable and can be performed at other locations, and the difference in the elapsed times will only reflect the system's DMA print performance, with no degradation due to the buffered printer itself, as shown in Figure 4.

The secret is to use a small piece of hardware called a *wrap connector* instead of a real printer. When properly configured, the wrap connector simulates a fully buffered printer. It attaches to the parallel port on the system board. It receives the data intended for the printer, but it discards that data right away and immediately asks for more. The wrap connector therefore removes delays caused by the printer interface, enabling the parallel port to operate at maximum speed. Because there is no overhead from a real printer, the results of the experiment will be a true reflection of the performance improvement.

Figure 5 shows the schematic for wiring the wrap connector to simulate a fully buffered printer.

Results Using the Wrap Connector


1
10
12
11
13
15
16
18
Jumper from pin 1 to pin 10
Jumper from pin 13 to pin 15 and to pin 16
Jumper from pin 11 to pin12 and to pin18

The wrap connector can be made by connecting pin 1 to 10, pin 13 to 15 and 16, and pin 11 to 12 and 18 on a standard 25-pin male D-shell (DB-25) connector that mates with the parallel port. Standard D-shell connectors are available at electronics stores. An example is Radio Shack® part number 2761547B (Connector) and part number 2761549B (Cover), or their equivalents.

Figure 5. Wire Wrap Connector

Both parts of the experiment in this article were also performed on Micro Channel systems with the wire wrap connector attached to the parallel port in place of the 4029 printer. Because of the absence of printer interface delays, the performance in both parts of the experiment improved by as much as ten seconds. Figure 6 shows the results using the wrap connector.

In this "pure" environment, the performance advantage of Micro Channel, DMA, ABIOS, and OS/2 2.0 becomes even more striking. Now, comparing the performances of PIO versus DMA on the 386 SX running at 20 MHz, we see that 57 seconds were reduced to less than 2 seconds – a 96.5% improvement. That same 386 SX-20 running OS/2 and DMA was 83% faster (10 seconds saved divided by 12 seconds) than even the fastest 486 DX-50 running DOS and PIO.

Creating this Experiment from OS/2 2.0 Diskettes

This experiment requires some preliminary creation of diskettes, data files, and a batch program.

Preliminary Steps

1. Copy the OS/2 2.0 Installation Diskette and label the copy "Installation Utility."
2. Copy OS/2 2.0 diskette #1 and label the copy Diskette 2.
3. On the newly created Diskette 2, modify `CONFIG.SYS` by changing the line `basedev=print01.sys` to `basedev=print02.sys`. This enables the DMA print driver for the listing of Micro Channel systems.

Processor	DOS with PIO[1]	Micro Channel with DMA, ABIOS, OS/2 2.0[2]
386 SX at 20 MHz	57 seconds	< 2 seconds
386 SLC at 20 MHz	29	< 2
386 DX at 20 MHz	48	< 2
386 DX at 25 MHz	39	< 2
486 DX at 25 MHz	28	< 2
486 DX at 33 MHz	19	< 2
486 DX at 50 MHz	12	< 2

[1] This print experiment should not be construed as a benchmark for the comparison of programmed I/O performance between systems, but as a comparison of the efficiency of Direct Memory Access and PIO transfers.

[2] This time includes file seek and any inaccuracies due to coarse resolution of the real-time clock for short intervals. Variations from 1.5 to 2.5 seconds are possible.

Figure 6. Results on Various Micro Channel Systems with a Wrap Connector

```
echo off
echo *************************************************************************
echo *                 DMA Parallel Port Performance Test                    *
echo *This test demonstrates the length of time to transmit a file to the    *
echo *parallel port. Running under the DOS operating system, data is         *
echo *transmitted using PIO, resulting in single-byte transfers.  Using the  *
echo *OS/2 2.0 operating system, printing results in large buffer transfers, *
echo *achieving higher performance. OS/2 2.0 takes advantage of a new DMA    *
echo *hardware feature available on PS/2 Micro Channel models 56, 57, 80-A21,*
echo *80-A31, 90, and 95. The improved performance results in higher printer *
echo *utilization by running printers at rated speed while leaving more time *
echo *for the processor to execute applications.                             *
echo *This program is the property of IBM Corporation.                       *
echo *Chet Heath, Barry Noto, and Frank Schroeder, IBM Boca Raton, Florida.  *
echo *************************************************************************
echo on
pause
echo off
echo *************************************************************************
echo *The following instructions set up the test to run from the hard disk.  *
echo *Running from the hard disk more closely shows Micro Channel performance*
echo *************************************************************************
echo on
c:
md c:\prtdemo
cd c:\prtdemo
copy a:\test c:\prtdemo
echo off
echo *************************************************************************
echo *The following instructions perform the test.                           *
echo *************************************************************************
echo on
time <CR
copy testprt1 lpt1
time <CR
echo off
echo *************************************************************************
echo *Calculate the elapsed time, then press any key to continue.            *
echo *************************************************************************
echo on
pause
echo off
echo *************************************************************************
echo * The following instructions remove the test from the hard disk.        *
echo *************************************************************************
echo on
cd c:\
del c:\prtdemo <a:\test\YES
rd c:\prtdemo
a:
******************************************************************************
```

Figure 7. Batch Program to Run Experiment

Note: For ISA machines, another diskette without this change must be made with the proper driver to enable ISA programmed I/O ports. Otherwise, the version of Diskette 2 would be identical.

Warning: If some ISA is operated with the Micro Channel driver, the operation will terminate prematurely with no data transfer and/or other unpredictable results.

4. Create a subdirectory on Diskette 2 named `TEST`.

5. Create the file `TEST.CMD`, as shown in Figure 7, in the `TEST` subdirectory on Diskette 2.

6. Copy the `TEST.CMD` file and name the copy `TEST.BAT` in the subdirectory. OS/2 2.0 will execute the

`TEST.CMD` file in the same manner that DOS executes `TEST.BAT`.

7. On the hard disk, create a file called `TESTPRT1` by copying text files together until the `TESTPRT1` file has approximately 200,000 bytes. This is done by listing text file names all on one line, with no spaces around the plus signs, as shown below:

```
COPY
TEXTFILE.one+TEXTFILE.two+...
```

   A file that contains only null characters (00 hex) will not print if you use a real printer instead of a wrap connector for this experiment. After creating `TESTPRT1` on the hard disk, copy the completed file to the `TEST` subdirectory on Diskette 2.

8. Make a three-byte file called `CR` and fill it with a single carriage-return character. Add this file to the `TEST` subdirectory. The carriage-return character in the `CR` file responds to the time prompt, enabling the batch file to continue processing. Most file editors put three characters into this file – the carriage-return character, a line-feed character, and the end-of-file character.

9. Make a four-byte file called `YES` and fill it with a single letter Y and the carriage-return character. (The file editor will add the line-feed and end-of-file characters.) Add this file to the `TEST` subdirectory. This file enables the batch file to respond to the “Are you sure?” prompt when deleting the contents of the subdirectory on the hard disk.

10. (optional) The `README` file is placed on Diskette 2 for convenience, in case these instructions are misplaced.

11. For this experiment to be portable, make the wrap connector shown in Figure 5.

When steps 3 through 10 have been completed, the `TEST` subdirectory on Diskette 2 will resemble this:

```
.               <DIR>
..              <DIR>
TEST      CMD       3115
TEST      BAT       3115
TESTPRT1          196606
CR                     3
YES                    4
README              4076
```

You are now ready to begin the experiment. Parts 1 and 2 of the experiment are described below.

Part 1

1. Be sure to purge print spoolers and any other TSR programs before proceeding.
2. Boot DOS from the hard disk or from diskette.
3. Insert Diskette 2 into drive A.
4. If you booted from the hard disk, type `a:`.
5. Type `cd \test`, then press Enter.
6. Type `test`, then press Enter. This starts the batch file `TEST` from Diskette 2.
7. Follow the prompts in the batch file. Calculate elapsed time from the start and end times that are displayed, and write it down.

Part 2

1. Insert Diskette 1 into drive A.
2. Press Ctrl-Alt-Del to start OS/2.
3. At the prompt, replace Diskette 1 with Diskette 2, then press Enter.
4. Look for the first screen that allows you to press Esc to cancel.
5. Press Esc. OS/2 will display an [A:\] prompt.
6. Type `cd a:\test`, then press Enter.
7. Type `test`, then press Enter. This starts the batch file `TEST` from Diskette 2.
8. Follow the prompts in the batch file. Calculate elapsed time from the start and end times that are displayed, and write it down.

After completing Part 2, compare the elapsed times generated in Parts 1 and 2.

Acknowledgments

I thank Frank Schroeder and Barry Noto of the OS/2 2.0 development team for realizing that if the OS/2 2.0 installation process is canceled, the system is placed in a state in which the DMA print function remains enabled. Without that critical perception, a diskette-based, portable experiment would not have been possible. Frank is also the author of the batch program used in the experiment.

***Chet Heath** is a Senior Technical Staff Member and engineer at IBM's Entry Systems Technology Laboratory, in his twenty-second year with IBM. He holds a BS in electrical engineering from the New Jersey Institute of Technology (Rutgers) and an MS in electrical engineering from the LSI Institute at the University of Vermont. Chet was the primary innovator of the PS/2's Micro Channel architecture and has led its definition since inception. He has received numerous awards for his work on Micro Channel architecture including IBM's eighth-level invention award, Outstanding Technical Achievement, Outstanding Innovation, and Corporate Technical Achievement awards. He was named one of the ten most influential people on PCs in the '80s by* PC User *magazine in the U. K.*

Pen-Based Computers

Leo L. Suarez
IBM Corporation
Boca Raton, Florida

Pen-based computers, commonly called tablets, are a new technology in portable personal computing. Tablets make us look at computers in an entirely different way because these computers enable us to process data and communicate by using a very familiar tool – the pen. This article discusses the classifications of pen-based systems, their hardware components, and their operating systems.

Most industry consultants forecast a tremendous growth for tablet computers during the next five years; they consistently predict compound growth rates of over 100%. These same consultants predict that within the next ten years, tablets will be as pervasive and ubiquitous as the telephone. This article explains the composition of a "typical" tablet computer and its unique hardware and software requirements.

Classifications of Pen-Based Systems

Like keyboard-based portable computers, which are categorized into different sizes and function (transportable, laptop, and notebook), tablet computers are divided into several categories. The size dictates the functionality of the machine and the applications for which the system will be used. This section describes the most common classes of tablets. Unlike keyboard-based machines, where terminology such as "laptop" or "notebook" has become widely accepted, the tablet industry is still in its infancy; therefore, the names used below may not be the ones ultimately chosen by the industry.

Palm-Top

The palm-top is the smallest form factor for a tablet system. A palm-top typically measures 9.8 x 4.7 x 1.3 inches (245 x 117 x 32 mm) and weighs about 19.6 ounces (560 grams). The processor is either a proprietary design or a design based on a 16-bit microprocessor such as the Intel 80C88. The display resolution is small (640 x 200 pixels) and the screen size is typically 7.25 inches diagonal. Storage is accomplished via an integrated-circuit storage card. The pen is tethered and a limited number of I/O ports is provided. These machines are used primarily for specific vertical applications such as personal calendars, appointments, telephone numbers, and so on. They are the least expensive tablet systems, and currently the most widely seen and used systems.

B5 (Small Tablet)

B5 tablets (the term "B5" refers to a paper size) measure 10.2 x 7.2 x 1.3 inches (254 x 180 x 32 mm) and weigh approximately 2.4 pounds (1.1 kg). Until recently, these machines have been designed around microprocessors such as the Intel 80286. More powerful versions designed around the IBM 80386 SLC processor will be introduced in time for the 1992 Fall COMDEX®. The display resolution used can be as large as 640 x 480 pixels, although the screen size is still small, about 8 inches diagonally. B5 tablets incorporate either solid-state files or 1.8-inch rotating media for storage. They usually have one or more Personal Computer Memory Card International Association (PCMCIA) slots for memory or other options designed to communicate via PCMCIA interface standards.

Machines can have tethered or untethered pens. Data collection – such as inventory control, maintenance logs, or signature capture – is a typical usage for these machines. They are used primarily by people who work in the field. For example, a distributor checking the inventory at a local supermarket uses the pen to check off each item in a roster of products and to enter the quantities to order. As wireless connectivity becomes a reality, B5 tablet systems will become very popular for executives and managers.

A4 (Large Tablet)

A4 tablets (A4 is another paper size) measure 12.2 x 9.2 x 1.4 inches (305 x 229 x 35 mm) and weigh up to 5.7 pounds (2.6 kg). They are based on powerful processors such as the Intel 80386SX. This class of tablet system will probably be the first to have more powerful processors, such as the i486™ or even RISC-based processors. These computers use either solid-state files or rotating media. The larger form factor allows the use of either 1.8-inch or 2.5-inch hard disk drives, so large storage capacities of 120 MB and higher are possible. The display is typically 640 x 480 pixels; it uses a larger pixel size of .29 mm and measures 9.5 inches diagonally. This class of system usually has a full complement of I/O ports: serial, parallel, keyboard, modem, SCSI, and others. Graphic-intensive applications are best suited for this class of machine. For example, an insurance adjuster could bring up the image of an automobile and mark on the screen to show the area that was damaged. A parts list would then appear, enabling the adjuster to easily calculate cost of repair.

Convertible

The convertible is a computer whose mechanical design allows the system to be transformed from a familiar "clamshell" portable to a keyboardless, pen-based system. The form factor and weight of a convertible computer are similar to those of a notebook. The larger form factor allows the machine to have the same functional capabilities of a notebook system with the additional input capability of a stylus. This class of system runs applications that require much text entry, and occasional applications that use the pen.

Desktop Tablet

A desktop tablet system provides an alternate way to enter data into a desktop personal computer. The desktop tablet computer is connected to one of the PC's Input/Output (I/O) ports. The tablet is not a stand-alone, pen-based system; it depends on the PC's processor, I/O, power, and storage capability to function. Therefore, it is not portable. The desktop tablet is primarily used to develop tools or application programs for pen-based software. Another use includes substituting for a mouse or other pointing device. Many users want the capability of a desktop personal system that accommodates occasional use of pen-based applications.

The Hardware

One's first observation about a tablet computer is that it does not look like a typical computer. It looks more like a book or a binder, hence the term "tablet." Also, the absence of a keyboard is very evident. A pen-based computer has several key components, shown in Figure 1. Inside the tablet are all the familiar subsystems normally found in a portable computer: the processor complex and logic, memory, storage, power supply, battery, display, and I/O ports. However, as one would expect, these subsystems have been uniquely designed to meet the requirements of pen computers. This section describes each key component shown in Figure 1.

Processor Complex

The processor complex is the one component that resembles a laptop or notebook computer's motherboard. Today's tablet systems are typically designed around a logic chip set that has been optimized with power management features such as suspend/resume. Suspend/resume is the ability to power down the computer temporarily without losing the application and data being processed. In this state, battery life is significantly prolonged. When the user is ready to resume operation, the computer powers up and returns control to the exact place where the application was before the computer was suspended. (For details, see the article "PS/2 Model L40SX Laptop Portable Computer" in Issue 3, 1991 of *Personal Systems Technical Solutions*.) Integrated on the motherboard are a DC/DC power supply and all the I/O connectors and control hardware.

Storage

Storage on tablets is accomplished by one of two methods. The first is to use Electrically Erasable Read-Only Memory (EEROM), also called Solid-State Files (SSF). The second way is to use a rotating hard disk. While these two technologies may look familiar to the reader – EEROM has existed for many years and rotating media are nothing new to computers – the way these technologies have been packaged for tablet systems is unique.

SSFs use EEROM with cell sizes of 7.8 microns or smaller, and can pack 4 MB of data on a 75 mm^2 chip. They are typically made on semiconductor manufacturing lines with design tolerances of 1.0 microns or less. The chips are then assembled on a printed circuit board along with an Integrated Drive Electronics- (IDE-) compatible memory controller, chip driver, and buffer circuitry. The entire assembly for a 20 MB SSF is about 3 x 2 x .28 inches (76.9 x 50.8 x 7.1 mm), and it weighs about 1.2 ounces (35 grams). To the computer's hard disk controller, the SSF looks exactly like a rotating hard disk, so it is completely transparent to the operating system software.

The major advantages of SSF design are ruggedness and low power. A typical SSF can withstand operating shock of 500G (where one G is the acceleration due to gravity) when the system is operating, and 1000G when it is not. Power consumption is about 0.5 watts for reading and 1 watt for writing or erasing. In a standby or sleep state, the SSF consumes only 0.02 watts per hour.

The rotating hard-disk technology in best-of-breed tablets uses a disk that is 1.8 inches in diameter. These incredibly small hard disks – smaller than the size of a typical business card – can pack large densities, up to 60 MB. The 1.8-inch hard disks have the same features, such as an integrated controller and power management capability, that are found in the more common 2.5-inch hard disks in laptop and notebook portables. The size and weight of 1.8-inch hard disks make them ideal for tablets.

Display

Tablets, like most portables, use Liquid Crystal Displays (LCDs). However, unlike most LCDs found on laptops and notebook computers, tablet LCDs are *transflective*. A transflective LCD is a display that can use either the surrounding ambient light or an internal Cold Cathode Fluorescent Lamp (CCFL) to illuminate the display. The transflectivity is accomplished by designing a backlight diffuser that reflects ambient light back to the display when the CCFL is not turned on.

The advantage of a transflective LCD is that it can be used outdoors and in other bright lighting conditions. A

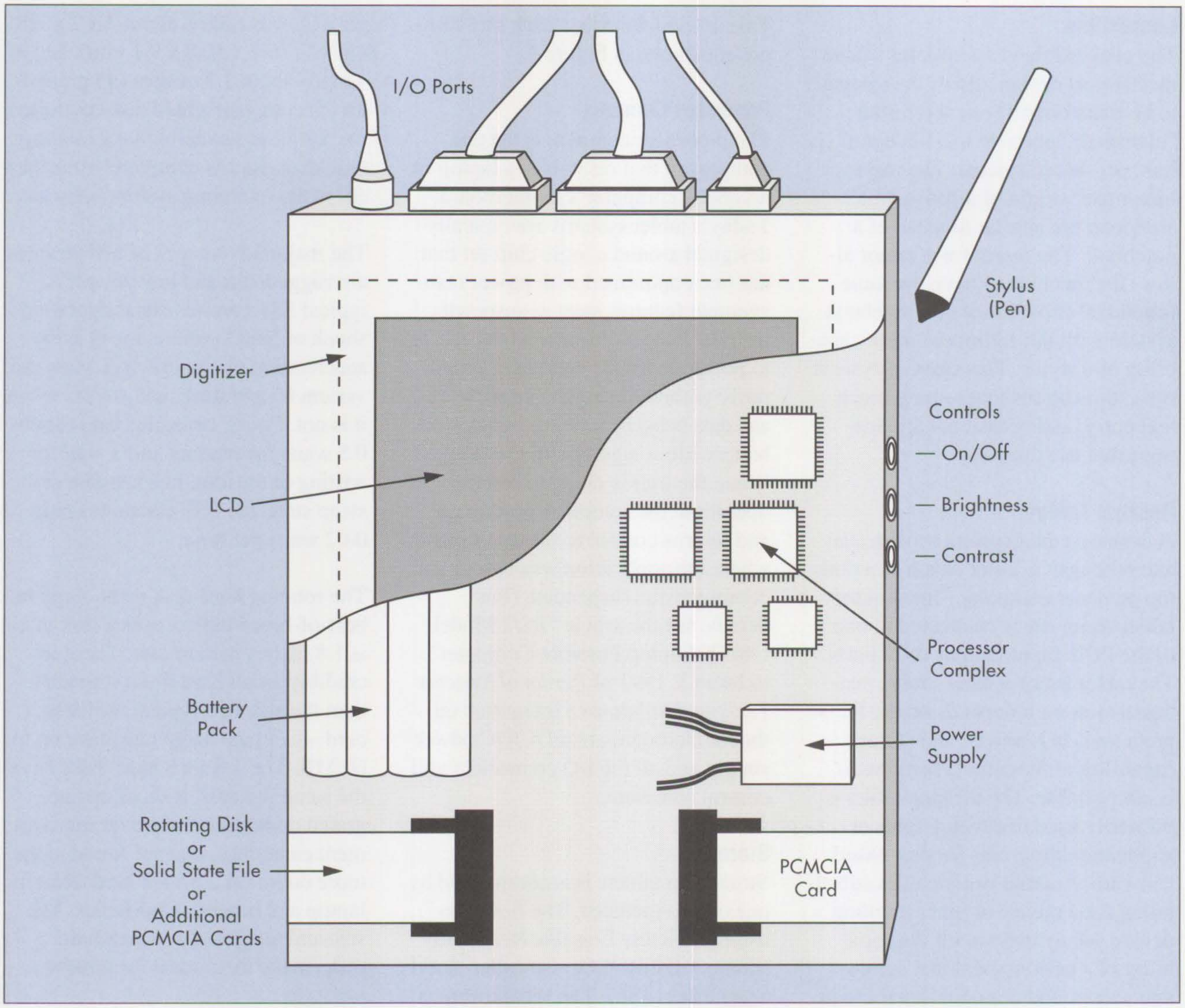


Figure 1. Components of a Pen-Based Computer

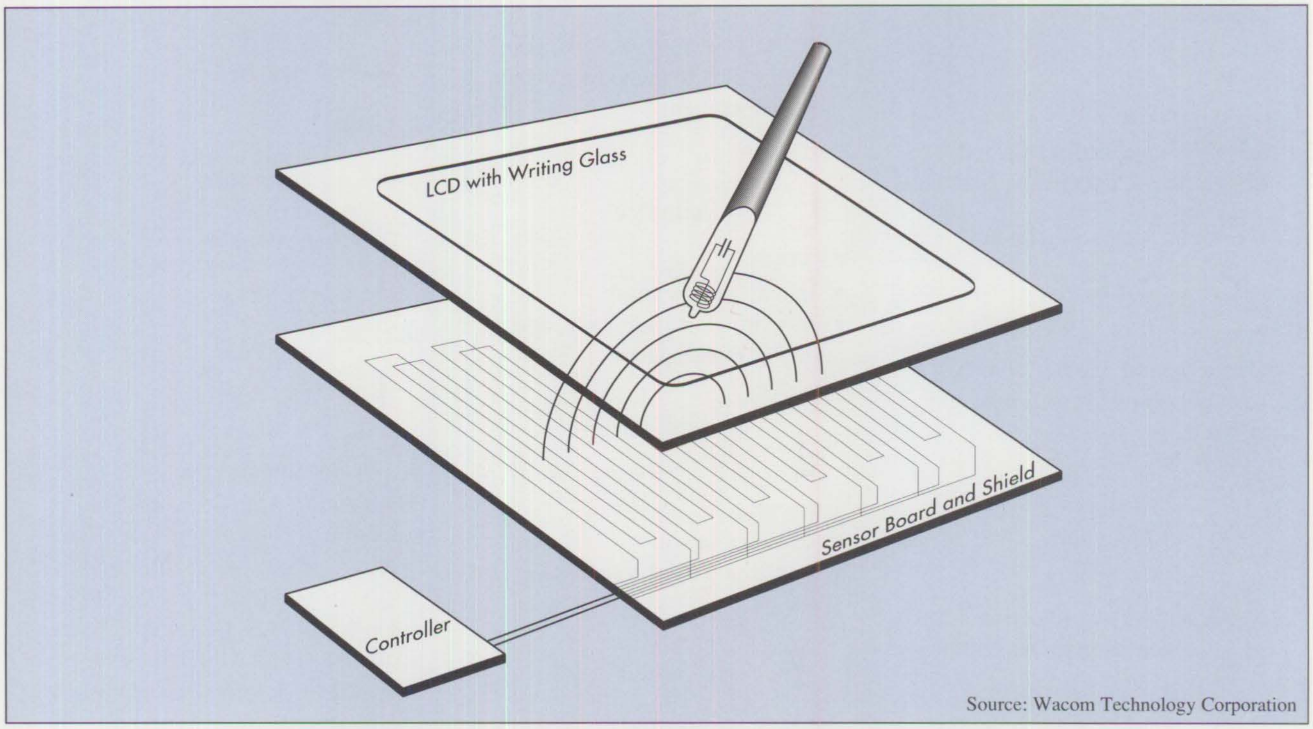
typical transmissive LCD, found in laptops and notebooks, cannot be used in bright light conditions because the image tends to wash out. A second advantage is that when the transflective LCD is used in a bright light environment, the back light is not needed and can be turned off. The ability to turn the back light off achieves a significant power reduction. This translates into a substantial increase in battery life, because the back light accounts for more than 25% of the total power consumption in a typical portable computer or tablet.

Digitizer and Pen

There are two important requirements for entering data into tablet computers. The first is that the pen must not be physically tied to the computer. Cordless operation is important because the objective is to simulate the natural feeling and freedom of a pen or pencil. The second requirement is to simulate the tactile feel or pressure of pen to paper.

To accomplish the cordless design, an electromagnetic resonance technology, shown in Figure 2, is used. The tactile feel of the pen is accomplished by using a pressure sensing tip, in which the desired pressure necessary to input data to the system can be adjusted by the user. Other techniques use slightly textured LCD glass to simulate the resistance of paper.

Pressure is also important because when we write with a pen or pencil, more or less pressure on the pen normally yields a wider or narrower line. Pressure sensing can be designed in, so that the tablet behaves as closely as possible to paper and pencil.



Source: Wacom Technology Corporation

How a Cordless, Batteryless Pen Works. Cordless input for pen computers is made possible with an untethered stylus that looks and feels like a pen, yet contains no batteries or magnets. Instead, it takes advantage of electromagnetic resonance technology developed by Wacom Co. Ltd. in which radio waves are sent to the stylus and returned for position analysis. In operation, a grid of wires below the screen alternates about every 20 microseconds between transmit and receive modes. The transmitted energy produces electromagnetic resonance in the stylus, where the electromagnetic energy is stored in a coil-and-capacitor resonant circuit. In receive mode, the stylus re-emits a signal at a different frequency back to the grid of wires. This is analyzed to determine positional and other information including pressure, if required. The electromagnetic system does not require the pen to be connected to the computer to provide the high-quality data essential for handwriting recognition.

Figure 2. Electromagnetic Resonance Technology

Operating Systems

Tablet computers are designed to substitute for pen and paper. Therefore, the operating systems used in tablet systems are very different from those in traditional keyboard-based computers. There are two types of operating systems used in tablets. The first is known as pen-enabled, and the second is called pen-centric.

Pen-Enabled Operating Systems

Pen-enabled operating systems are designed to work primarily with applications in which the pen substitutes for a pointing device. Therefore, pen-enabled operating systems are used in familiar applications that run on desktop or portable computers and normally use a mouse as the pointing device. Pen-enabled operating systems are best suited for horizontal applications, used by a large number of users across many market segments. Examples are spreadsheets, databases, communications (such as faxing), calendars, and word processors.

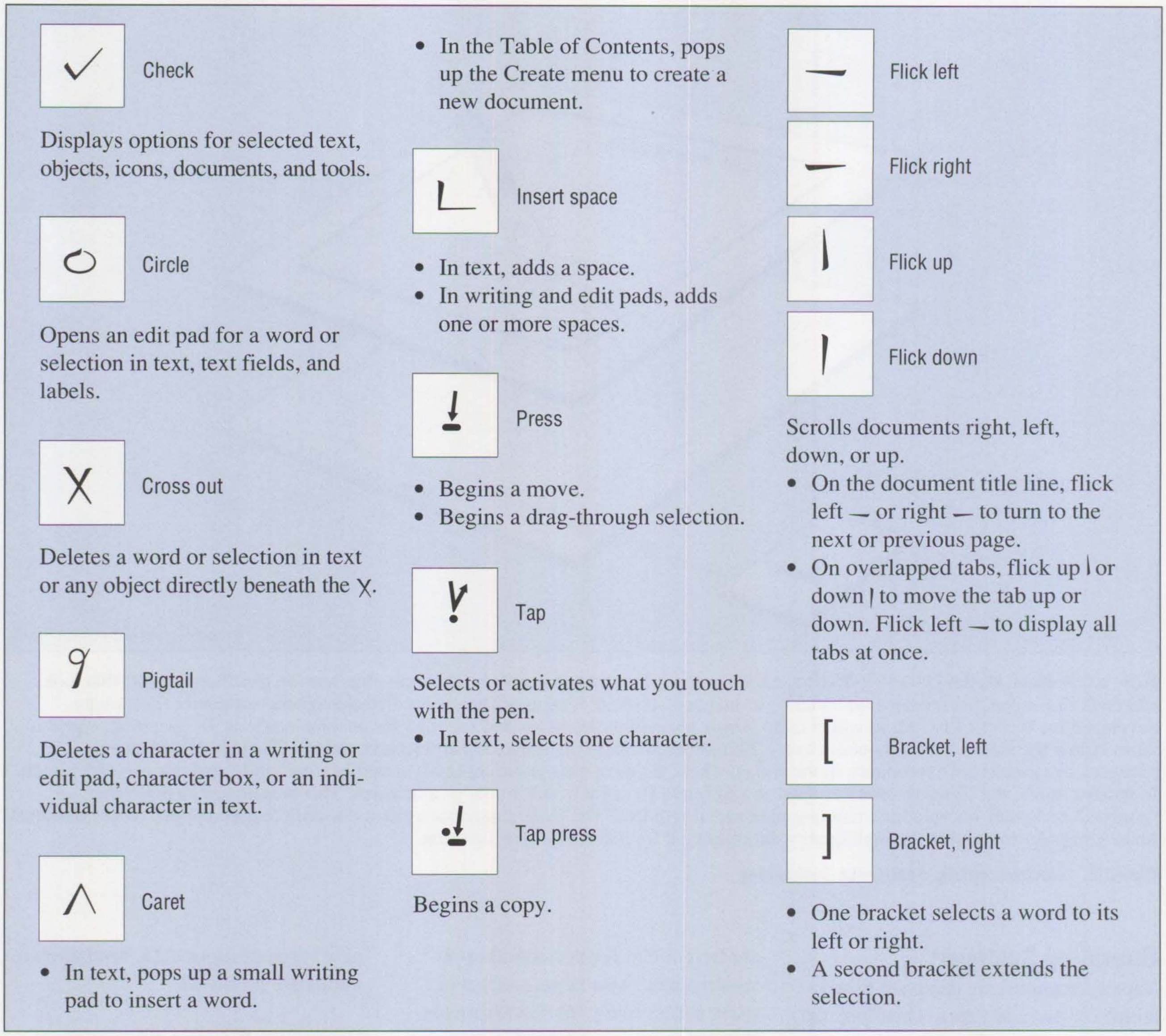
Pen-Centric Operating Systems

The pen-centric operating system is designed specifically for pen-based computers. Pen-centric applications tend to be classified as vertical. Vertical applications are usually custom-designed to meet the requirements of a particular market niche, such as filling in forms or inventory checklists. The pen-centric operating system is unique in that it uses gesturing, electronic ink, and graphics as input information, in addition to handwritten or typed characters. Because these are new types of input data, brief explanations are provided.

Gesturing: Specific movements of the pen have different meanings to the operating system. Figure 3 gives examples of a few popular gestures and their corresponding meanings, used by Go Corporation's PenPoint™ operating system. The number of gestures is limitless and depends only on the operating system used.

Electronic Ink: This is input from the stylus that is not translated into ASCII characters, as typed on a keyboard. Examples include signature capture and sketches.

Graphics: Images displayed on the tablet can be altered, stored, or ap-



Check

Displays options for selected text, objects, icons, documents, and tools.

Circle

Opens an edit pad for a word or selection in text, text fields, and labels.

Cross out

Deletes a word or selection in text or any object directly beneath the X.

Pigtail

Deletes a character in a writing or edit pad, character box, or an individual character in text.

Caret

- In text, pops up a small writing pad to insert a word.
- In the Table of Contents, pops up the Create menu to create a new document.

Insert space

- In text, adds a space.
- In writing and edit pads, adds one or more spaces.

Press

- Begins a move.
- Begins a drag-through selection.

Tap

Selects or activates what you touch with the pen.

- In text, selects one character.

Tap press

Begins a copy.

Flick left

Flick right

Flick up

Flick down

Scrolls documents right, left, down, or up.

- On the document title line, flick left — or right — to turn to the next or previous page.
- On overlapped tabs, flick up | or down | to move the tab up or down. Flick left — to display all tabs at once.

Bracket, left

Bracket, right

- One bracket selects a word to its left or right.
- A second bracket extends the selection.

Figure 3. Basic PenPoint Gestures

Source: Go Corporation

pended the same way words and sentences would be on a keyboard-based machine. Examples include pictures taken by a digital camera or images picked up by a scanner, which are then read into the computer.

Summary

Pen-based systems are a new personal computer paradigm. They will be used widely and commonly in the 21st century. Over the next several years, their size and weight will decrease while their capabilities will increase dramatically. Individual users and businesses who take advantage of this emerging technology today will be uniquely positioned to compete more effectively against their present and future competitors.

***Leo L. Suarez** is a product manager for portable system strategy and requirements in IBM's Personal Systems Line of Business. He joined IBM in 1978 as a component test engineer on the Series/1™ and workstations. He has since held positions ranging from technology staff at division headquarters to numerous management assignments in both development and manufacturing. Prior to his current assignment, Leo was involved in the design and development of the IBM PS/2 Models P70, P75, and L40 SX laptop computers. Leo holds a BS and an MS in electrical engineering from the University of Miami.*

Why Doesn't My Portable's Battery Last Longer?

Leo L. Suarez
IBM Corporation
Boca Raton, Florida

Many users of portable computers are amazed at all the function and power that is packed into these small, lightweight machines. Yet, they are equally amazed at how quickly their batteries run out. This article describes the problems and challenges facing designers of portable systems as they strive to provide users with computers that can last for an eight-hour day on a single battery charge.

Even though advances in miniaturization have allowed designers to produce very small, lightweight subsystems in portable computers, many of these same subsystems still require considerable amounts of power to operate.

The best way to highlight this problem is to look at how power is consumed in a typical portable computer. Figure 1 shows the major subsystems in a portable computer and their corresponding percentages of power usage. This analysis assumes a "typical" portable has a 386 SX processor running at 20 MHz, with 2 MB of main memory, a 60 MB hard disk, a 640 x 480 monochrome backlit Liquid Crystal Display (LCD), a 3.5-inch diskette, and serial, parallel, mouse, and external video ports.

The two biggest power consumers are the motherboard logic and the display. These two subsystems represent the biggest area of opportunity for reducing overall system power consumption. The other subsystems, such as the hard disk and diskette, consume significant power too. But since they are not typically used continuously, their power consumption can be controlled. The "typical" portable system specified above consumes a maximum of 20 watts of power when all subsystems are operating.

The Weight Problem

Because a portable computer is not truly portable unless it can be carried easily, weight is the single biggest obstacle in providing longer battery life. Portable system designers could easily increase the battery life by making the battery larger, but a larger battery has several drawbacks. The additional cells or denser cells will increase its weight. An increase in the battery size also means that the physical size of the computer must increase to accommodate the larger battery footprint. Larger frames and covers, in turn, increase the weight of the machine. So, if the size of a portable computer cannot be increased, but additional battery life is required, then system function must be removed to make room for the larger battery. Deletion of functions could make the machine less appealing.

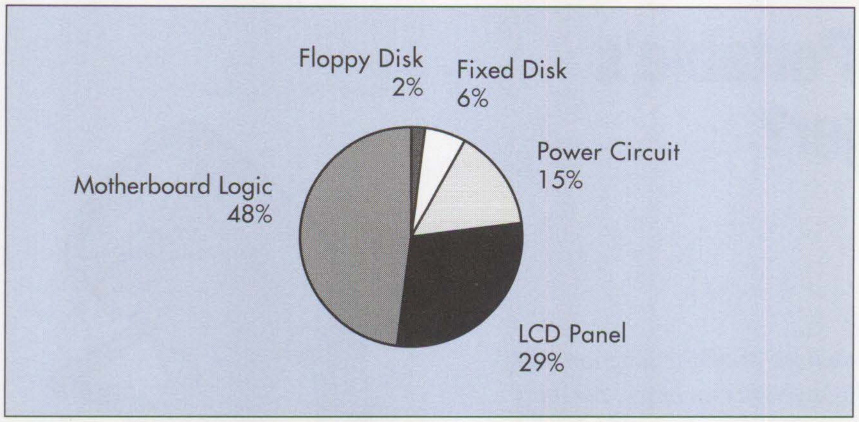
For example, a state-of-the-art Nickel-Cadmium (NiCd) battery's weight will increase by 18 grams for each additional watt-hour of performance. To double the life of the "typical" portable computer's battery by increasing its rated life from 20 to 40 watt-hours, the battery's weight would increase from approximately 360 grams to 720 grams.

Alternatives

Given these problems, what is the best solution? As usual, the answer depends on the application. Typically, the compromise solution considers the alternatives described below.

Reduced Motherboard Power Consumption

Figure 2 shows how power consumption on the motherboard is distributed. The logic chip set is an area of great opportunity for motherboard power reduction. This can be accomplished most efficiently through functional integration, or "logic sweeps," which reduces the overall chip count. Significant power is saved by producing components that provide multiple functions. Also, fewer external support devices, such as off-chip drivers, are needed to tie the independent components together. Eliminating these external support devices can save a significant amount of power.



Note: Percentages represent power usage.

Figure 1. Major Subsystems in a Portable Computer

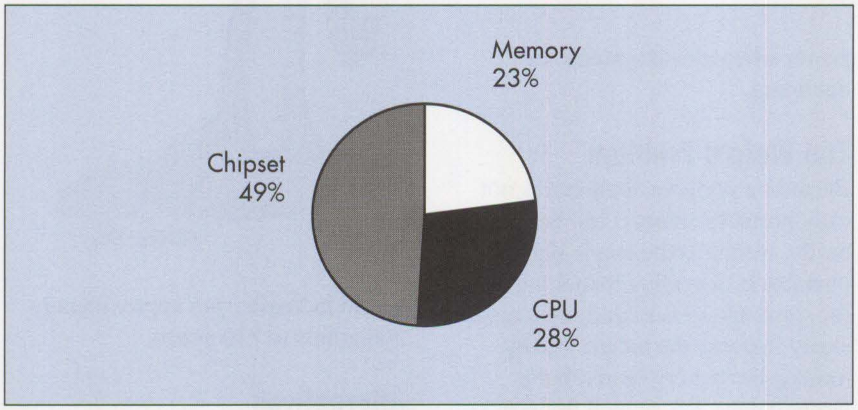


Figure 2. Motherboard Power Consumption

Integration of multiple functions on one piece of silicon will typically use fewer transistors than the equivalent multi-chip solution. Fewer transistors mean less power is required. Functional integration also reduces the total machine weight, since fewer components have to be soldered to the motherboard.

The power drain by the memory and processor can also be reduced by progressing from today's +5-volt logic to a more energy-efficient +3.3-volt logic. Because power is directly proportional to the square of the voltage used, a +3.3-volt design will consume, on average, 44% less power than the equivalent +5-volt design. Many components and subsystems are not yet available in +3.3-volt logic. However, this will change over the next two years as motherboard designers first incorporate mixed +5-volt and +3.3-volt logic, and ultimately 100% +3.3-volt designs. In the interim, subsections of the motherboard can be designed to operate at +3.3 volts. Also, semiconductors that can operate at either +3.3 volts or +5 volts should be used when available.

More Efficient LCD Design

The main consumer of power in an LCD is the backlight. The larger the area required to backlight, the more power is required. Because the pixel size determines the area of the LCD, a 12% power savings was achieved on the PS/2 Model L40 SX by using a pixel size of 0.31 mm instead of a larger 0.33 mm pixel. Also, because the power required to backlight the display is directly proportional to the area of the display, a corresponding 12% power savings was achieved.

The use of efficient fluorescent bulbs along with a mechanically efficient backlight diffuser design (a diffuser is the apparatus that distributes the light behind the LCD) can also drastically reduce power requirements. For example, the original L40 SX LCD used a 10 mm diameter bulb; by changing the bulb to a 6 mm design, the diffuser efficiency increased. In fact, the same amount of brightness could be delivered for approximately 10% less power.

The final major design consideration is the type of LCD used – color versus monochrome. A color LCD will consume approximately three times more power than the equivalent monochrome display at the same brightness. This is because the red, green, and blue color filters required to generate color allow less light to pass through than the transparent filter normally used in a monochrome LCD. Therefore, if long battery life is required, monochrome displays are beneficial because they consume significantly less power.

Denser Battery Technologies

During 1991, the most common rechargeable battery used in portables was nickel-cadmium. The energy density of a good nickel-cadmium battery during this time was approximately 55 watt-hours per kilogram or 160 watt-hours per liter. Nickel-cadmium manufacturers expect that by 1997 this technology can be improved to about 60 watt-hours per kilogram or 190 watt-hours per liter. There is little room left for improvement in this technology.

During the second half of 1991, a new battery technology, nickel-metal-hydride, was introduced in portables. This technology is more expensive and is not yet available in volume. It yields 10% to 20% more energy density than an equivalent nickel-cadmium battery. In 1992, battery suppliers are adding substantial manufacturing capacity; with the additional volumes, prices should start to come down.

While there are several other battery technologies, only one other appears to have merit. This technology is called lithium-ion. It has the potential for significant energy density improvements. The claims being made today by the two companies that have announced this technology – Toshiba® and Sony® – are very promising. For example, their functional objectives for lithium-ion technology include energy densities of 99 watt-hours per kilogram or 236 watt-hours per liter – significant improvements over nickel-cadmium and nickel-metal-hydride. Lithium-ion technology, however, is not expected to be available for portable computers until 1993. In the near term, designers of portable computers must ensure that their machines accept both nickel-cadmium and nickel-metal-hydride. A design change to lithium-ion should occur when the technology truly materializes.

Power Management

Once a portable design has been optimized for power consumption at the subsystem level, additional battery life can be achieved with a power management system. A power management system oversees the power consumption of components in a portable computer, and selectively shuts down those subsystems that are not being used. Powering only the subsystems that are needed extends battery life.

Other Areas

Although the rest of the subsystems are less critical, they all should have basic power-saving capabilities. For example, they should be able to shut down when the power management system tells them to shut down. Options such as data/fax modems and additional memory should be designed for minimum power consumption and have power management features. A power saving capability should be the minimum requirement of any subsystem; if it is not available, that subsystem should not be seriously considered for inclusion in the base design.

A portable system user can dramatically increase a nickel-cadmium or nickel-metal-hydride battery's life by understanding the power management options provided by the computer. These features, found in PS/2 portables such as the L40 SX, N51 SX, N51 SLC, and CL57 SX, enable the user to turn off devices and ports not used, and to determine how long the hard disk is kept on between periods of inactivity. Reducing the brightness of the display by using the adjustment slides or knobs will also reduce power consumption.

Nickel-cadmium and nickel-metal-hydride batteries suffer from a phenomena commonly referred to as *memory effect*. Memory effect is a condition in which a battery that is not fully discharged appears to the battery charger as if the battery is almost fully charged. When this happens, the charger will not fully recharge the battery. Allowing the battery to fully discharge before recharging will ensure the maximum possible charge during the recharging period, and minimize the memory effect.

Conclusion

A portable will be truly "portable" when the computer is lightweight, small, and – most importantly – when it can last through an entire workday on a single battery charge.

The ability for one battery to last for eight hours in the "typical" system described above is not far away. It will take a combination of higher integration, improved battery technology, and efficient subsystem designs. The eight-hour battery life will not happen suddenly. It will come about as users define the optimum combination of hardware, function, size and weight that meets their applications, and as machines are designed to meet those requirements.

***Leo L. Suarez** is a product manager for portable system strategy and requirements in IBM's Personal Systems Line of Business. He joined IBM in 1978 as a component test engineer on the Series/1 and workstations. He has since held positions ranging from technology staff at division headquarters to numerous management assignments in both development and manufacturing. Prior to his current assignment, Leo was involved in the design and development of the IBM PS/2 Models P70, P75, and L40 SX laptop computers. Leo holds a BS and an MS in electrical engineering from the University of Miami.*

Planning Guidelines for Token-Ring Cabling

Robert D. Love, Thomas Toher, and Lee C. Haas
IBM Corporation
Research Triangle Park, North Carolina

This article gives guidelines on the use of Unshielded Twisted-Pair (UTP) media for 16 Mbits per second token-ring networks. Information about Shielded Twisted-Pair (STP) media and optical fiber is included to help system administrators develop long-term cabling plans.

In the past five years, the bandwidth available for data transmission on Local Area Networks (LANs) has dramatically increased. While network designers have developed faster and faster means of transmitting data, cable designers have developed cables and terminations that meet the performance demands of the new networks. As a major supplier of LANs, IBM carefully studies developments in cable design. They want to ensure that users can reap the benefits of the latest cable technologies without sacrificing the robust characteristics that they have come to expect from IBM network products.

During the last five years the cabling industry has focused on improving the performance characteristics of Unshielded Twisted-Pair (UTP) cables. The following enhancements to IBM's 16 Mbits per second Token-Ring Network offerings allow for operation on UTP:

- A new family of token-ring 16/4 media filters and new technology in the IBM 8230 concentrator support re-clocked token rings.
- Existing IBM Token-Ring Network 16/4 adapters can be used with the new IBM 8230 concentrator by connecting through new external media filters.
- Some future IBM adapters may offer on-board media filters that may eliminate the need for external filters for those adapters.

Cabling Recommendations for 16 Mbits Per Second Token-Ring Networks

The 150-ohm shielded twisted-pair cable is the best transmission medium for token-ring networks. It allows the maximum number of stations to be attached, the maximum distance to be driven, and significant protection from external noise.

Recent testing with data-grade UTP cabling from the work area to the wiring closet has shown that it also can be used over typical lobe distances. The performance capabilities of a 16 Mbits per second Token-Ring network using UTP depend on the electrical characteristics of the cable and its terminations. The cabling and terminations must be designed as a system to effectively support higher data rates as transmission rates increase. Older cable installations should be evaluated to determine their suitability for 16 Mbits per second data transmission. Components for new installations should be selected to optimize the potential performance for both current and future use.

Historical Perspective

In late 1985, IBM announced the type 3 specification for telephone twisted pair. This was the first high-frequency specification for UTP cables available in the industry. Developed by IBM in cooperation with cable manufacturers, it was the most detailed high-frequency specification that cable manufacturers could meet at that time. Consequently, it has no crosstalk specifications and no attenuation specifications above 1 MHz. The type 3 specification was intended to help users who wanted to use their installed cabling for token-ring transmission. Therefore, the specification was written to include as much of the installed base as practical, rather than to create hard-to-meet performance criteria. Despite these limitations, it has served as a useful minimum requirement for 4 Mbits per second token-ring transmission on UTP cables.

The next major advance in specifying UTP for high-speed data transmission came with the development of EIA/TIA 568[1]. The IEEE® Project 802 Local Area Network Standards committees provided input for developing the specifications in this standard. The UTP specifications in the standard are more comprehensive than the type 3 specification – both crosstalk and attenuation characteristics through 16 MHz are specified. Consequently, some type 3 cables meet the EIA/TIA 568 category 3 specification while others do not. EIA/TIA 568's category 3 UTP cable specification should be used as an update and a replacement for IBM's type 3 specification.

[1] *Commercial Building Telecommunication Wiring Standard.* July 1991. Electronic Industries Association, Engineering Department, 2000 Pennsylvania Avenue NW, Washington, D.C. 20006.

While EIA/TIA 568 was nearing completion, cabling manufacturers were developing higher performance UTP cables. Although the standards committee recognized that these new cables would be extremely beneficial for high data rate applications, the standard was too close to completion to include them in the first edition. In November 1991, EIA/TIA issued Technical System Bulletin 36 (TSB-36), which defined five categories of UTP cables: categories 1 and 2 for voice and low-speed data only (not recommended by the standard), category 3 as described above, and categories 4 and 5 with specifications up to 20 and 100 MHz respectively, for high-speed telecommunications applications. The categories 4 and 5 cables are specified at higher frequencies and have lower attenuation and crosstalk, as shown in Figures 1 and 2.

Copies of all TIA standards, including TSBs, are available for a fee through Global Engineering Documents. In the USA and Canada, call (800) 854-7179; internationally, call (714) 261-1455.

Two Solutions Meeting the Needs of UTP Categories

The new IBM 8230 Controlled Access Unit supports higher grade categories 4 and 5 UTP cables. The new IBM 8230 Model 2 provides concentrators with retiming for each lobe to support categories 3, 4, and 5 at lobe lengths that will meet the needs of most installations.

Media Filters for 16 Mbits Per Second Token Ring on UTP

Media filters must always be used for categories 3, 4, and 5 UTP cable. Transmission of high data rate signals on UTP requires filtering to stay within national emission limits for electrical energy. IBM's 16/4 Token-Ring Adapters and corresponding concentrator ports require external filters to permit attachment to UTP lobes operating at either 4 or 16 Mbits per second. In addition to the filters that attach a token-ring adapter, a filter is needed for the IBM 8230 Controlled Access Unit for 4 or 16 Mbits per second token ring operation on UTP lobes.

Frequency in MHz	EIA/TIA Category 3	EIA/TIA Category 4	EIA/TIA Category 5
0.064	2.8	2.3	2.2
0.256	4.0	3.4	3.2
0.512	5.6	4.6	4.5
0.772	6.8	5.7	5.5
1.0	7.8	6.5	6.3
4.0	17	13	13
8.0	26	19	18
10.0	30	22	20
16.0	40	27	25
20.0	—	31	28
25.0	—	—	32
31.25	—	—	36
62.5	—	—	52
100.0	—	—	67

Note: In dB at 1000 Feet at 20° Celsius

Figure 1. Attenuation of Horizontal UTP Cable

Frequency in MHz	EIA/TIA Category 3	EIA/TIA Category 4	EIA/TIA Category 5
0.150	54	68	74
0.772	43	58	64
1.0	41	56	62
4.0	32	47	53
8.0	28	42	48
10.0	26	41	47
16.0	23	38	44
20.0	—	36	42
25.0	—	—	41
31.25	—	—	40
62.5	—	—	35
100.0	—	—	32

Note: In dB at 1000 Feet at 20° Celsius

Figure 2. Minimum Near-End Crosstalk (NEXT) Loss Worst Pair

Configurations without Retiming Concentrators

Tests were conducted using the original IBM 8230 Controlled Access Unit combined with the new 16/4

	4 Mbits Per Second Token Ring		16 Mbits Per Second Token Ring	
UTP Category	Maximum Number of Attachments	Maximum Lobe Length (Meters)	Maximum Number of Attachments	Maximum Lobe Length (Meters)
3	72	100	Not Supported	
4/5	132[1]	100	132	100

[1] The maximum attachment limit is 72 stations if there are any 4 Mbits per second-only adapter cards in the ring.

Figure 3. Maximum Lobe Lengths without Retiming Concentrators

	4 Mbits Per Second Token Ring		16 Mbits Per Second Token Ring	
UTP Category	Maximum Number of Attachments	Maximum Lobe Length (Meters)	Maximum Number of Attachments	Maximum Lobe Length (Meters)
3	132[1]	100	132	90
4/5	132[1]	100	132	100

[1] The maximum attachment limit is 72 stations if there are any 4 Mbits per second-only adapter cards in the ring.

Figure 4. Maximum Lobe Lengths with Retiming Concentrators

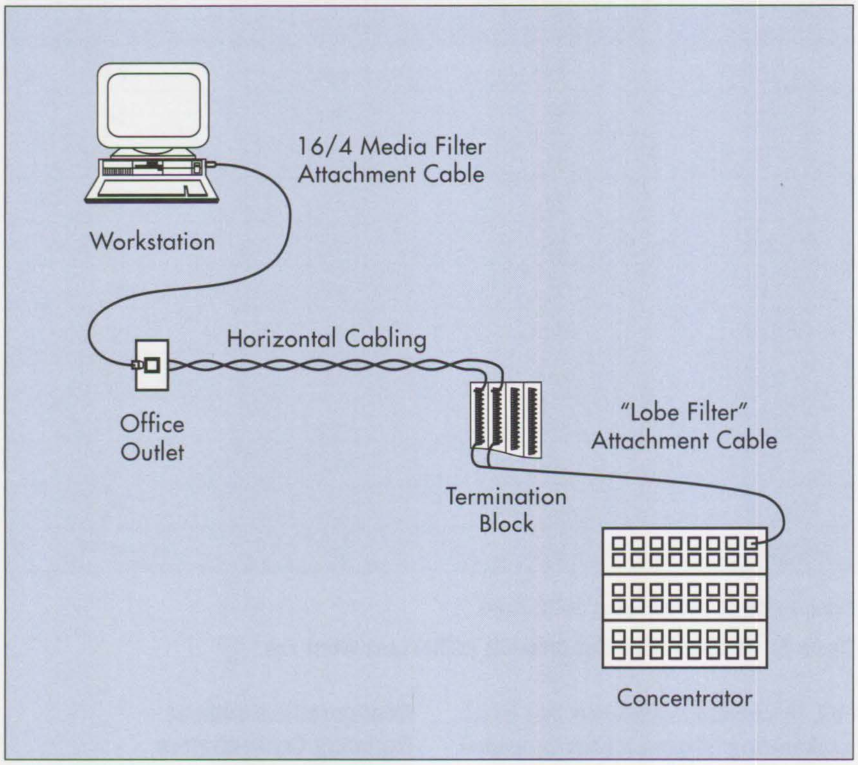


Figure 5. Block Diagram of a Token-Ring Network UTP Lobe

media filters. Figure 3 shows the maximum allowable attachment limits and maximum lobe lengths as functions of UTP grade.

Support for Categories 3, 4, and 5 UTP at 16 Mbits Per Second

To achieve desired distances and station counts over category 3 UTP cable requires an active concentrator, such as the IBM 8230 Model 2, which provides signal regeneration with retiming. Figure 4 gives the maximum supportable lobe lengths using retiming concentrators.

Multiple Wiring Closets

The main ring path of UTP 16 Mbits per second token rings should remain within a single wiring closet. However, main ring paths can span multiple wiring closets by using repeater functions. All main ring connections must use STP or multimode optical fiber.

Termination of UTP Cabling

To achieve the distances described in Figures 3 and 4, it is necessary to terminate the UTP cabling properly. Figure 5 shows a block diagram of a token-ring lobe.

Both the device and the concentrator attachment cables must be the specially designed high-quality cables specified by IBM, and must be installed following IBM's specified procedures. The horizontal cabling must be terminated at the telecommunications outlet in a high-quality modular connector. To minimize crosstalk, the cable twist should be maintained to the termination points. The termination block should be a high-quality insulation displacement-style telephone block. Again, the fixed cabling should be untwisted no more than is necessary to make the termination. To assist in limiting crosstalk and attenuation, no intermediate blocks should be used. The horizontal cabling from the termination block to the telecommunications outlet must be dedicated to the token-ring signals for that single lobe connection. No other voice or data signals are permitted in the same cable sheath, which prevents crosstalk among the pairs.

Media	4 Mbits Per Second 802.5	10 Mbits Per Second 802.3	16 Mbits Per Second 802.5	100 Mbits Per Second FDDI
Coax	C	S C	C	none
Fiber	C	S C	C	S C
STP	S C	C	S C	u C
UTP Category 3	a C	S C	u C	u
UTP Category 4	a C	S C	u C	u

S - Meets appropriate standard

C - Commercial product currently available

a - Described in appendix of a standard

u - Under investigation by standards committee

Figure 6. Horizontal Cabling and the Applications Standards

Presently Installed UTP Cabling

When planning a network on previously installed UTP cabling, you must know whether that cable was manufactured according to any of the applicable EIA/TIA UTP cabling category specifications. Fortunately, that assessment should be straightforward. Cables should be clearly labeled by both the manufacturer and the installer. Markings on the cable sheath (accessible in the wiring closet) should clearly indicate the manufacturer and the manufacturer's model number for that cable. The manufacturer can tell you if that model number meets any EIA/TIA cable category specifications.

The installer's label should provide a way to determine the length of the permanently installed cable and its termination points. If any part of the manufacturer's or installer's labeling is missing, a hand-held cable tester (available from a variety of vendors) can help determine cable length, attenuation, and crosstalk over a frequency range from 1 to 16 MHz. Some testers are designed to test the suitability of the cable for a specific application, such as 16 Mbits per second token ring on UTP. Testers can alert you to incorrect installation procedures as well as poor quality cable. A good general rule is to test any cable that has been installed for three or more years.

Selecting New Cables for Data Transmission

The EIA/TIA-568 *Commercial Building Wiring Standard* suggests four basic types of cable to use for data transmission between work areas and wiring closets: coaxial cable, shielded twisted pair, unshielded twisted pair (divided into three acceptable categories), and 62.5/125-micron multimode optical fiber. The standard mandates that two copper paths must always be present: one for data and the other for voice. Any optical-fiber cable installed should be in addition to the copper cable.

The LAN standards groups use the EIA/TIA-568 standard in defining transmission media for the LAN protocols. Figure 6 shows the relationship between these LAN standards and the *Commercial Building Wiring Standard.* Once you have estimated demand for bandwidth to the desktop over the life of the cable plant you are planning, Figure 6 can help you choose which cables you should install from the wiring closet to each work area.

Cable Choice and Standards

Figure 6 shows that 150-ohm STP cable should be considered as the premier telecommunications cabling choice when rewiring or when wiring new installations. It is often the most cost-effective solution when maintenance and future applications are considered. However, because each cabling decision is driven by its own set of unique cost considerations, installation of UTP is sometimes the appropriate choice. When this cable is chosen, only the high-performance UTP cables (categories 4 and 5) should be used. Because their performance is so much greater than that of category 3 cables, the small additional cost should not be a deterrent. Note that category 3 cable is still an acceptable choice for telephone and low-speed (less than 1 Mbit per second) applications.

The terminating hardware and standard installation practices for 150-ohm STP cabling provide end-to-end cabling integrity and quality. With UTP cabling, both the terminating hardware and the installation practices can strongly affect the overall performance of the cabling system. Wall outlets and wiring-closet terminating blocks that retain the overall transmission quality of category 4 cable are available. Category 5 cable pushes the limits of today's technology; no generally recognized hardware ensures that the benefits of category 5 cable are preserved. In addition, careful workmanship is required to prevent significant degradation of the system's electrical performance parameters from those for the category 5 cable itself. Still, category 5 cable is worth considering for new installations, even if the connecting hardware may need to be replaced later.

New Horizons for IBM's Shielded Twisted Pair

The 150-ohm STP cable, designed in the early 1980s, still stands out as a first-class cabling choice for high-speed data transmission. Its high characteristic impedance is fundamental in achieving low transmission attenuation. The shield is needed to transmit the highest possible data rates while staying within government emission standards. The 16 Mbits per second token-ring operation on shielded 150-ohm cable has been a standard since 1989. The 100 Mbits per second Fiber Distributed Data Interface (FDDI), running on that cable for at least 100 meters from the workstation to the wiring closet, is available from several manufacturers. IBM expects to support 100 Mbits per second for workstation attachment on the IBM Cabling System shielded twisted pair in 1992. In October 1991, IBM announced the F-Coupler, which allows STP to be used for broadband transmissions in the 50 to 550 MHz band while simultaneously running 4 or 16 Mbits per second token-ring signals. For further information about the F-Coupler, see the *F-Coupler Planning Guide* (GA27-3949).

The 150-ohm STP cable still stands out as a first-class cabling choice for high-speed data transmission.

As imaging applications begin to play a rapidly increasing role in our computing environments, we may ultimately reach the absolute bounds of either the physical or regulatory laws that govern data transmission. When we reach those bounds, optical fiber is waiting for wider use in office environments. But for the future, both the IBM Cabling System shielded twisted-pair and data-grade UTP have an important place in data transmission systems.

Robert D. Love *is a senior engineer at IBM's Network Systems Division in Research Triangle Park, North Carolina. He is an expert in token-ring design, and has played a crucial role in the development of IBM Token-Ring Network wiring rules for transmission on both shielded and unshielded media. Bob has been a significant contributor to the development of token-ring standards within IEEE 802.5, and as a United States Expert on the ISO/IEC JTC1/SC25/WG3 standards body developing cabling guidelines for both token ring and generic applications. He has published numerous articles about transmission design, LAN transmission comparisons, and token ring design, and has presented papers in these areas at national and international symposia. Bob received a BS in electrical engineering from Columbia University and an MS in electro-physics from the Brooklyn Polytechnic Institute.*

Thomas Toher *is an advisory information developer at IBM's Network Systems Division in Research Triangle Park, North Carolina. He received a BA and MA in English from Hobart College and Clark University, respectively. Tom joined IBM in 1984, and has been the lead writer for all IBM Cabling System, IBM Token-Ring Network, and IBM PC Network products developed at the Research Triangle Park laboratory. Tom is an IBM representative to the EIA/TIA TR41 standards concerned with building cabling, infrastructure, and administration. Tom frequently presents the subjects of cabling and planning for Token-Ring networks at GUIDE, SHARE, and other forums.*

Lee C. Haas *is a senior engineer in IBM's Local Area Network development organization at Research Triangle Park, North Carolina. He is a registered Professional Engineer, and holds BSEE and MSEE degrees in electrical engineering from Virginia Polytechnic Institute and State University, and San Jose State College, respectively. He is pursuing a PhD in electrical engineering at North Carolina State University. Lee joined IBM in 1967 and has worked in circuit design, signal processing, data communications, and modem product development. His present work is in product development for LANs, optical fiber systems, and related standards activities of American National Standards Institute (ANSI®), EIA/TIA, and IEEE. He is a national officer of Theta Tau, a professional engineering fraternity.*

Installing and Migrating Applications in OS/2 2.0

Hans Goetz
IBM Corporation
Boca Raton, Florida

This article discusses the installation of DOS and Windows applications under OS/2 2.0. It shows how to use the application migration utilities shipped with OS/2 2.0 and gives an example of how to use the PARSEDB utility to create a customized migration database.

Installing DOS and Windows applications under OS/2 2.0 is often similar to installing them in their native environments. However, because OS/2 2.0 is a true multitasking operating system, you must ensure that the installation programs do not introduce incompatibilities with existing programs. The flexibility in tailoring the virtual DOS environment for these programs also affords the opportunity to easily tune DOS settings to suit your needs.

OS/2 2.0 has a utility to help place application icons onto the Desktop after they have been installed. The utility also customizes the Settings notebook for commonly available DOS, Windows, and OS/2 applications. The utility uses information stored in a migration database that is shipped with OS/2 2.0.

System administrators can use another utility to create their own migration database to install their company's unique applications.

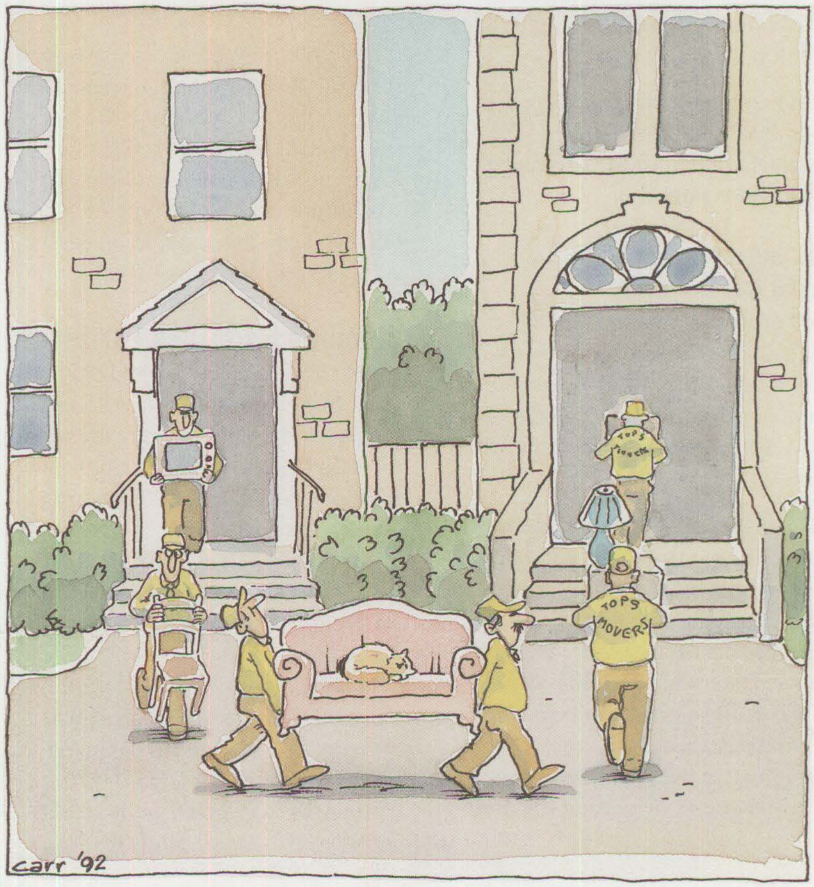
Installing DOS Programs

Application installation methods vary widely in the DOS world. Some installations involve nothing more than copying the software from diskette to the hard disk. In more complex applications, the installation procedure may check the workstation's configuration (both hardware and software), implement copy protection, and modify system files.

For most DOS applications, installation under OS/2 2.0 is simply a matter of starting a DOS full-screen or windowed session and following the instructions supplied with the package, as if the installation were taking place on a DOS system. However, some applications may not work correctly because of the special requirements of the installation program.

General Installation Procedure for DOS Programs

To install a new DOS program, use the following procedure.



Partition/Type	Drive Letter Under OS/2 2.0	Drive Letter Under DOS
Primary Partition 1, Boot Manager	none	none
Primary Partition 2, FAT	C:	C:
Extended Partition, HPFS	D:	none
Extended Partition, FAT	E:	D:

Figure 1. Possible Drive Letter Assignments on a Mixed FAT/HPFS Hard Disk

1. Read the installation instructions for the DOS program.
2. Select the OS/2 System icon.
3. Select the Command Prompts icon.
4. Select the DOS Full-Screen icon.
5. At the command prompt, enter the installation command specified in the instructions. For example, type `a:install` at the prompt.
6. Follow the instructions on the screen.
7. When installation is complete, close the Command Prompts folder.

To add an icon to the Desktop, use either the Program template in the Templates folder, or the Migrate Applications program.

Installation Programs with Special Requirements

Some programs that install DOS applications will not run properly in the OS/2 2.0 virtual DOS machine, or will not install the application correctly. Some possible problems are as follows:

- The installation program attempts to verify the version of DOS that is running, and receives a response that it cannot understand. One example is Lotus 1-2-3® Release 3.1+. Here, the OS/2 2.0 virtual DOS environment can be customized to return a DOS version in response to the installation program's query, bypassing the problem. This is done by changing the `DOS_Version` parameters in the DOS Settings facility, which is accessed by pressing the DOS Settings pushbutton on the Session page of the Settings notebook.
- The copy protection or user registration scheme implemented by some applications bypasses DOS and directly accesses the disk. OS/2 2.0 will not allow the installation program to do this, because it may interfere with other applications and terminate the virtual DOS machine in which it is running.

It may therefore be necessary to perform the installation in a native DOS environment by rebooting the workstation with a DOS boot diskette. When installation is complete, the workstation can be rebooted under OS/2 2.0 and the application can be added to the Workplace Shell.

Planning Hard Disk Partitions

If the workstation is booted from a DOS diskette to install a DOS application, the installation is restricted to those logical drives that have been formatted as the File Allocation Table (FAT). This is because logical High Performance File System (HPFS) drives cannot be accessed in a native DOS environment.

When the system is booted with DOS 5.0, HPFS drives are not assigned drive letters, and are invisible to the user. If booted with DOS 4.01 with Corrective Service Diskette (CSD) UR35284, drive letters are assigned to all drives, whether HPFS or FAT; however, the files cannot be accessed on the HPFS drives. In earlier versions of DOS, even FAT drives that lie beyond the first HPFS drive are not assigned drive letters.

Some installation programs store directory information in control files that are used at runtime. For example, WordPerfect® 5.1 records the path to its subdirectories. On a hard disk with both FAT and HPFS logical drives, this can cause the installation program, which runs in native DOS, to record drive assignments that will be wrong when the application is started from a virtual DOS machine.

Consider the example, shown in Figure 1, of a hard disk that is set up for dual boot or with Boot Manager. Figure 1 shows how drive letter assignments may differ on a mixed FAT and HPFS hard disk when the computer is booted under DOS and OS/2 2.0.

In Figure 1, the FAT drive in the extended partition appears as drive E: to the OS/2 2.0 virtual DOS machine, but it appears as drive D: when booted under DOS. Consequently, if the DOS application is installed on that partition when the system was booted under DOS, the drive letter it records in its control files will be D:. When the system is rebooted under OS/2 2.0 and the application runs in a virtual DOS machine, the application looks for drive D:. But under OS/2 2.0, drive D: is assigned to the HPFS drive of the extended partition. This inconsistency causes the application to seek the wrong drive and to miss the information it needs.

To correct the error, you may be able to change the control file information in the application from D: to E:. However, if the system is then booted from DOS, the application will look for a non-existent drive. To avoid this confusion, place the HPFS logical drives last. In Figure 1, the FAT and HPFS logical drives in the extended partition should be transposed. This allows the drive letters for the FAT partitions to be the same (that is, D:) whether the workstation is booted from DOS or from OS/2 2.0.

More details about hard disk management are contained in Chapter 4 of the *OS/2 2.0 Installation Guide*.

Installing Windows Programs

Windows programs are usually installed using the DOS command prompt or the Windows Program Manager. The following steps show how to install a Windows program from the DOS command prompt and also when using the Program Manager.

From the DOS command prompt:

1. Select the OS/2 System icon.
2. Select the Command Prompts folder.
3. Select DOS Full-Screen.
4. Enter the installation command specified in the instructions. For example, type `a:setup` at the prompt.
5. Follow the instructions on the screen to complete the installation.

Using the Program Manager:

1. Select the OS/2 System icon.
2. Select the Command Prompts folder.
3. Select WIN-OS/2 Full-Screen.
4. Select Run from the File pull-down on the action bar.
5. Enter the installation command as specified in the instructions. For example, type `a:setup` at the prompt.
6. Follow the instructions on the screen to complete the installation.

To add an icon to the Desktop, use either the Program template in the Templates folder or the Migrate Applications program. If you use the Migrate Applications option, the program icon will be placed in either the Windows Programs folder or the Additional Windows Programs folder on the Desktop.

The Migrate Applications programs always set up Windows programs to run in a WIN-OS/2 window session if possible. WIN-OS/2 window sessions cannot be used for programs, such as Windows 2.0 programs, that have to be used in real mode. If you use the Migrate Applications utility on real-mode programs, open the Settings notebook to the Sessions page and select the WIN-OS/2 Full-Screen radio button.

OS/2 2.0 has a migration database, DATABASE.DAT, which contains parameters and settings for commonly used DOS, Windows, and OS/2 programs.

AUTOEXEC.BAT and CONFIG.SYS

The installation program for a DOS or Windows application may alter the `AUTOEXEC.BAT` file (usually to modify the `PATH` statement) and the `CONFIG.SYS` file (to modify the `FILES` and `BUFFERS` statements or to add a `DEVICE` statement). We recommend that both these files be backed up before running an installation. We also recommend that if you have the option to view the changes before they are made, you should exercise that option.

The most common change made to the `AUTOEXEC.BAT` file is to the `PATH` statement, so that the program being installed can be started from any subdirectory. The equivalent function of the `PATH` statement is provided in a virtual DOS machine by using the Path and File Name field and the Working Directory field on the Program page of the Settings notebook.

Because the `CONFIG.SYS` file is used for every virtual DOS machine, the device driver added by an installation program will be loaded into all virtual DOS machines, consuming system resources unnecessarily. We recommend that when the DOS application is added to the Workplace Shell, the device driver statement should be added using the `DOS_DEVICE` setting in the DOS Settings facility. This is accessed by pressing the DOS Settings pushbutton on the Session page of the Settings notebook.

Migrating Programs

OS/2 2.0 has a migration database, `DATABASE.DAT`, which contains parameters and settings for commonly used DOS, Windows, and OS/2 programs. This binary database file is used by the Migrate Applications program to place the program icons onto the Desktop and to customize their Settings notebooks to the recommended values.

To use the Migrate Applications program, follow these instructions:

1. Select the OS/2 System icon.
2. Select System Setup.
3. Select Migrate Applications. The Find Programs window appears. The "Database Used for Find Option" field displays the default database (`\OS2\INSTALL\DATABASE.DAT`). The Migrate Applications program compares programs on the hard disk with the list of programs in the database, and

places any that match into a DOS, Windows, or OS/2 program folder on the Desktop.

4. From the Drives list, deselect the drives that should not be searched. (The default is to search all drives.)
5. In the Migrate Type check boxes, deselect the types of programs that should not be migrated. (The default is to migrate all the listed programs.)
6. Select Find. The Migrate Programs window appears. Programs are listed in the Applications list box.
7. If the program to be migrated is not in the list:
   a. Select the Add Programs pushbutton. The Add Programs window appears. Programs are listed in the Available Programs list.
   b. Highlight the program. The Working Directory and Program Title fields are filled in. Type a new title if required.
   c. Type the appropriate parameters for the selected program in the Parameters field.
   d. Select the types of programs to migrate in the Program Type field. The Migrate Applications program creates the Additional Programs folders based on the types of programs specified.
   e. Select Add. The program moves to the Selected Programs list.
   f. Select OK. The Migrate Programs window appears.
8. Select Migrate to migrate all the selected programs. When migration is complete, the Find Programs window reappears.
9. Select Exit.

The Migrate Applications program creates a DOS Programs folder and a Windows Programs folder. The programs in these folders have the recommended preselected settings for best performance.

If the Add Programs option was used, an Additional DOS Programs folder and an Additional Windows Programs folder are also created. The programs in these folders have default settings. If these programs do not run correctly, you will need to specify other settings.

Creating a Customized Migration Database

Some users have an installed base of unique or customized DOS and Windows applications. These programs are not listed in the migration database DATABASE.DAT that comes with OS/2 2.0. To accommodate these programs, the PARSEDB.EXE program in OS/2 2.0 enables a system administrator to build a customized migration database that can be used to set up these unique applications on the Workplace Shell Desktop.

PARSEDB

To create a customized migration database, first create the input text database file, then run PARSEDB to create the binary database file. To start PARSEDB, enter the following (all on one line) at a command prompt:

```
PARSEDB [path] DBTAGS.DAT
[path] text_database [path]
binary_database
```

where:

- DBTAGS.DAT is the file name that contains the definitions for the tags that define the DOS settings.
- text_database is the name of the file that contains the program settings for a specific DOS, Windows, or OS/2 program. This file is the main input file for PARSEDB.
- binary_database is the name of the new migration database file.

For example, to create a new database named MYDATA.DAT, enter:

```
PARSEDB E:\OS2\INSTALL\
    DBTAGS.DAT MYDATA.TXT
    MYDATA.DAT
```

Note: A file name must be specified for the binary database file. This prevents the PARSEDB utility program from overwriting the default database file DATABASE.DAT.

When creating the text database file, each program must have the following migration information:

- **NAME**: Name of the file that runs the program
- **TITLE**: Program object name that appears beneath the icon
- **TYPE**: DOS, Windows, or OS/2
- **ASSOC_FILE**: File name associated with the file name specified in the Name field (Use this file name to uniquely identify the program.)
- **DEF_DIR**: The directory into which the program is installed

The ASSOC_FILE and DEF_DIR can have the value NULL. You can type a file name and path into ASSOC_FILE and DEF_DIR; if the file name and path are unknown, you can type the word NULL. The NULL values must be included when defining the program if specific values for these fields cannot be provided.

When creating MYDATA.TXT, group the settings for a given program on consecutive lines. Use blank lines to mark the end of a program's settings. Begin non-blank lines with a token. The tag file DBTAGS.DAT defines valid token settings, limits, and default values for various DOS properties. Use the contents of the file DBTAGS.DAT (found in \OS2\INSTALL) shown in Figure 2 as a sample for creating your MYDATA.TXT file.

```
// dbtags.dat--DOS setting "tags" used by PARSEDB and MIGRATE.  Each "tag"
// consists of an index, a keyword, and a data type.
//
//             | DO NOT EDIT THIS FILE UNDER ANY CIRCUMSTANCES! |
//
//=============================================================================
// Allows BASIC-style comments.
// ----------------------------------------------------------------------------
1   REM                            NOP

// ----------------------------------------------------------------------------
// Required "fake" DOS settings.
// ----------------------------------------------------------------------------
2   NAME                           STR    // File name used to execute application
3   TITLE                          STR    // Icon (desktop) title
4   TYPE                           BYTE   // Application type
                                          // Valid settings: DOS
                                          //                 Windows
                                          //                 OS/2
                                          //                 custom (for Microsoft
                                          //                 Windows apps which
                                          //                 must run full-screen)
5   ASSOC_FILE                     STR    // Associated file (NULL if one is not
                                          // known)
6   DEF_DIR                        STR    // Default installation directory (NULL
                                          // if there is not one)

// ----------------------------------------------------------------------------
// Other "fake" DOS settings.
// ----------------------------------------------------------------------------
7   FOLDER                         STR    // Name of folder to create and/or put
                                          // the application icon in
8   PARAMETERS                     STR    // Application's command-line parameters
9   WORK_DIR                       STR    // Application's working directory

// ----------------------------------------------------------------------------
// "Fake" Windows settings
// ----------------------------------------------------------------------------
10  WIN_FILES                      STR    // Files to be copied to the WinOS2
                                          // directory when the application is
                                          // migrated
11  COMMON_SESSION                 BOOL   // Default: ON  -> the application is to
                                          //                 be run in the common
                                          //                 session
                                          //          OFF -> the application is to
                                          //                 be run "stand-alone"

// ----------------------------------------------------------------------------
// Real DOS settings.  NOTE: WIN_RUNMODE is not supported; all Windows apps are
// installed with SEAMLESS and WPS handles the mapping.
// ----------------------------------------------------------------------------
//  WIN_RUNMODE                    INT    // Valid settings: 10 (REAL)
                                          //                 11 (STANDARD)
                                          //                 12 (AUTO)
                                          //                 13 (SEAMLESS)
13  COM_HOLD                       BOOL   // Default: off
14  DOS_BREAK                      BOOL   // Default: off
15  DOS_DEVICE                     MLSTR  // Default: empty
16  DOS_FCBS                       INT    // Limits: 0 to 255, default 16
17  DOS_FCBS_KEEP                  INT    // Limits: 0 to 255, default 8
18  DOS_FILES                      INT    // Limits: 20 to 255, default 20
19  DOS_HIGH                       BOOL   // Default: off
```

Figure 2. Contents of DBTAGS.DAT File (continued)

```
20  DOS_LASTDRIVE               STR    // Limits: last physical drive to 'Z',
                                       //         default 'Z'
21  DOS_RMSIZE                  INT    // Limits: 128 to 640, default 640
                                       // NOTE: increments of 16
22  DOS_SHELL                   STR    // Default: "#:\OS2\MDOS\COMMAND.COM "
                                       //          "#:\OS2\MDOS\ /P" where #
                                       //          is the boot drive
23  DOS_STARTUP_DRIVE           STR    // Default: empty
24  DOS_UMB                     BOOL   // Default: off
25  DOS_VERSION                 MLSTR  // Default: DCJSS02.EXE,3,40,255
                                       //          DFIAOMOD.SYS,3,40,255
                                       //          DXMAOMOD.SYS,3,40,255
                                       //          IBMCACHE.COM,3,40,255
                                       //          IBMCACHE.SYS,3,40,255
                                       //          ISAM.EXE,3,40,255
                                       //          ISAM2.EXE,3,40,255
                                       //          ISQL.EXE,3,40,255
                                       //          NET3.COM,3,40,255
                                       //          EXCEL.EXE,10,10,4
                                       //          PSCPG.COM,3,40,255
                                       //          SAF.EXE,3,40,255
                                       //          WIN200.BIN,10,10,4
26  DPMI_DOS_API                STR    // Valid settings: AUTO (default)
                                       //                 ENABLED
                                       //                 DISABLED
27  DPMI_MEMORY_LIMIT           INT    // Limits: 0 to 512, default 3
28  EMS_FRAME_LOCATION          STR    // Valid settings: AUTO (default)
                                       //                 NONE
                                       //                 C000
                                       //                 C400
                                       //                 C800
                                       //                 CC00
                                       //                 D000
                                       //                 D400
                                       //                 D800
                                       //                 DC00
                                       //                 8000
                                       //                 8400
                                       //                 8800
                                       //                 8C00
                                       //                 9000
29  EMS_HIGH_OS_MAP_REGION      INT    // Limits: 0 to 96, default 32
                                       // NOTE: increments of 16
30  EMS_LOW_OS_MAP_REGION       INT    // Limits: 0 to 576, default 384
                                       // NOTE: increments of 16
31  EMS_MEMORY_LIMIT            INT    // Limits: 0 to 32768, default 2048
                                       // NOTE: increments of 16
32  HW_NOSOUND                  BOOL   // Default: off
33  HW_ROM_TO_RAM               BOOL   // Default: off
34  HW_TIMER                    BOOL   // Default: off
35  IDLE_SECONDS                INT    // Limits: 0 to 60, default 0
36  IDLE_SENSITIVITY            INT    // Limits: 1 to 100, default 75
37  KBD_ALTHOME_BYPASS          BOOL   // Default: off
38  KBD_BUFFER_EXTEND           BOOL   // Default: on
39  KBD_CTRL_BYPASS             STR    // Valid settings: NONE (default)
                                       //                 ALT_ESC
                                       //                 CTRL_ESC
40  KBD_RATE_LOCK               BOOL   // Default: off
41  MEM_EXCLUDE_REGIONS         STR    // Default: empty
42  MEM_INCLUDE_REGIONS         STR    // Default: empty
43  MOUSE_EXCLUSIVE_ACCESS      BOOL   // Default: off
44  PRINT_TIMEOUT               INT    // Limits: 1 to 3600, default 15
```

Figure 2. Contents of DBTAGS.DAT File (continued)

```
45  VIDEO_8514A_XGA_IOTRAP      BOOL  // Default: on
46  VIDEO_FASTPASTE             BOOL  // Default: off
47  VIDEO_MODE_RESTRICTION      ENUM  // Valid settings: NONE (default)
                                      //                 CGA
                                      //                 MONO
48  VIDEO_ONDEMAND_MEMORY       BOOL  // Default: on
49  VIDEO_RETRACE_EMULATION     BOOL  // Default: on
50  VIDEO_ROM_EMULATION         BOOL  // Default: on
51  VIDEO_SWITCH_NOTIFICATION   BOOL  // Default: off
52  VIDEO_WINDOW_REFRESH        INT   // Limits: 1 to 600, default 1
53  XMS_HANDLES                 INT   // Limits: 0 to 128, default 32
54  XMS_MEMORY_LIMIT            INT   // Limits: 0 to 16384, default 2048
                                      // NOTE: increments of 4
55  XMS_MINIMUM_HMA             INT   // Limits: 0 to 63, default 0

56  DOS_BACKGROUND_EXECUTION    BOOL  // Default: on
57  DPMI_NETWORK_BUFF_SIZE      INT   // Limits: 1 to 64, default 8
```

Figure 2. Contents of DBTAGS.DAT File

In DBTAGS.DAT, each line has the following syntax:

```
INDEX VALUE TYPE (optional
   comments)
```

where:

INDEX is a sequence number.
VALUE is the name of the setting.
TYPE is the type of value, that is, one of the following:

NOP	Comments; any line with this type is ignored
STR	A string value
INT	An integer value
BOOL	Boolean, with value ON or OFF
BYTE	Program type: DOS, Windows, or OS/2
MLSTR	A multi-line string with component lines on individual lines in the text database file

Using these types, various settings for programs can be defined.

Do *not* edit DBTAGS.DAT or create a new one; the DBTAGS.DAT file is intended solely to be a reference for creating your MYDATA.TXT file.

PARSEDB checks the validity of all entries in MYDATA.TXT and compares them to the settings definitions in DBTAGS.DAT. If all entries are valid, PARSEDB creates a binary database named MYDATA.DAT.

Errors in the text file will cause PARSEDB to exit and display one of the following types of messages:

- A message that a file is corrupted indicates embedded ASCII NULL characters in the input text file.
- A message about an invalid setting indicates the use of a setting not found in DBTAGS.DAT. The message should include a line number and the name of the input file.
- A message that an entry has missing parameters indicates the absence of the minimum settings for the entry.

PARSEDB does not check for duplicate entries in the input text file, nor does it require settings to be in any particular order. It is also not case-sensitive; that is, "Off" is treated the same as "OFF." We recommend that a copy of the input text file (DATABASE.TXT) for the default migration database file (DATABASE.DAT) be made and used as the template for your own input file. Figure 3 shows a sample input text file.

Conclusion

Installation of DOS and Windows applications under OS/2 2.0 is generally performed at a DOS command prompt or by the WIN-OS/2 Program Manager. Sometimes it may be necessary to boot from a DOS diskette to perform the installation. Any modifications to CONFIG.SYS and AUTOEXEC.BAT made by installation programs should be carefully reviewed.

If OS/2 2.0 is to be set up for Boot Manager or dual boot to DOS, the arrangement of the hard disk partitions needs to be planned.

The Migrate Applications program is used to migrate already installed DOS, Windows, and OS/2 programs; it also creates and places the program objects in a folder on the Desktop. If the DOS or Windows program is in the migrate database \OS2\INSTALL\DATABASE.DAT, the Migrate Applica-

```
REM database.txt--migration data for DOS, Windows, and OS/2 applications.
REM
REM ============================  data  begins  ==============================
REM ---------------------------------------------------------------------------
REM Windows file browser
REM ---------------------------------------------------------------------------
   NAME                         WNBROWSE.EXE
   TITLE                        File Browser
   TYPE                         Windows
   ASSOC_FILE                   WNBROWSE.HLP
   DEF_DIR                      \WNBROWSE
   MOUSE_EXCLUSIVE_ACCESS       OFF
   KBD_CTRL_BYPASS              CTRL_ESC
   COMMON_SESSION               ON
   VIDEO_SWITCH_NOTIFICATION    ON
   KBD_ALTHOME_BYPASS           ON
   DPMI_MEMORY_LIMIT            5

REM ---------------------------------------------------------------------------
REM WordPerfect 5.1 by WordPerfect
REM ---------------------------------------------------------------------------
   NAME                         WP.EXE
   TITLE                        WordPerfect
   TYPE                         DOS
   ASSOC_FILE                   WP.FIL
   DEF_DIR                      \WP51
   MOUSE_EXCLUSIVE_ACCESS       OFF
   IDLE_SENSITIVITY             88

REM ---------------------------------------------------------------------------
REM PMCAMERA OS/2 screen-capture utility
REM ---------------------------------------------------------------------------
   NAME                         PMCAMERA.EXE
   TITLE                        PM Screen Capture
   TYPE                         OS/2
   ASSOC_FILE                   PMCAMERA.HLP
   DEF_DIR                      \PMCAMERA
```

Figure 3. Sample Input Text File for PARSEDB

tions program automatically selects the recommended DOS settings for the program.

The Migrate Applications program always sets up Windows programs to run in a WIN-OS/2 window session. The Migrate Applications program is used in these situations:

- During installation of OS/2 2.0 if DOS, OS/2, or Windows programs are already installed on the hard disk
- When a DOS, OS/2, or Windows program is added to a working OS/2 2.0 system

The PARSEDB utility helps system administrators add an organization's unique applications to the migration database, or create their own.

This article is condensed from material in the "Red Book," OS/2 Version 2.0, Volume 2: DOS and Windows Environment *(GG24-3731), published by the IBM International Technical Support Center, Boca Raton, Florida. The Red Book contains the work of many contributors, such as systems engineers from around the world and developers from the Boca Raton laboratory.*

Printing Under OS/2 2.0

Jean N. Shortley
IBM Corporation
Boca Raton, Florida

This article gives an overview of the printing improvements in OS/2 2.0. There are now drivers for more than 200 printers. Customization of printer setups is simpler, because each printing device defined on a system is represented as an individual object. The Help has been significantly improved with guidelines available in the Help to diagnose printer problems. Printing can be performed from multiple DOS, Windows, and OS/2 sessions simultaneously.

In OS/2 version 1.X, adding or deleting printers and their drivers required a thorough knowledge of the OS/2 print subsystem. This is much simpler with OS/2 2.0. On the OS/2 2.0 Desktop, the printer is a device with an online notebook. The notebook records particulars about a printer, such as its driver, port, and other items specific to the printer model.

During OS/2 2.0 system installation, there is a choice of over 200 printers. Select one as the default printer, or select No Default Printer to skip installing a printer during system installation. After OS/2 2.0 system installation, it is easy to install additional printers, using printer drivers shipped with OS/2 2.0 or provided by printer vendors.

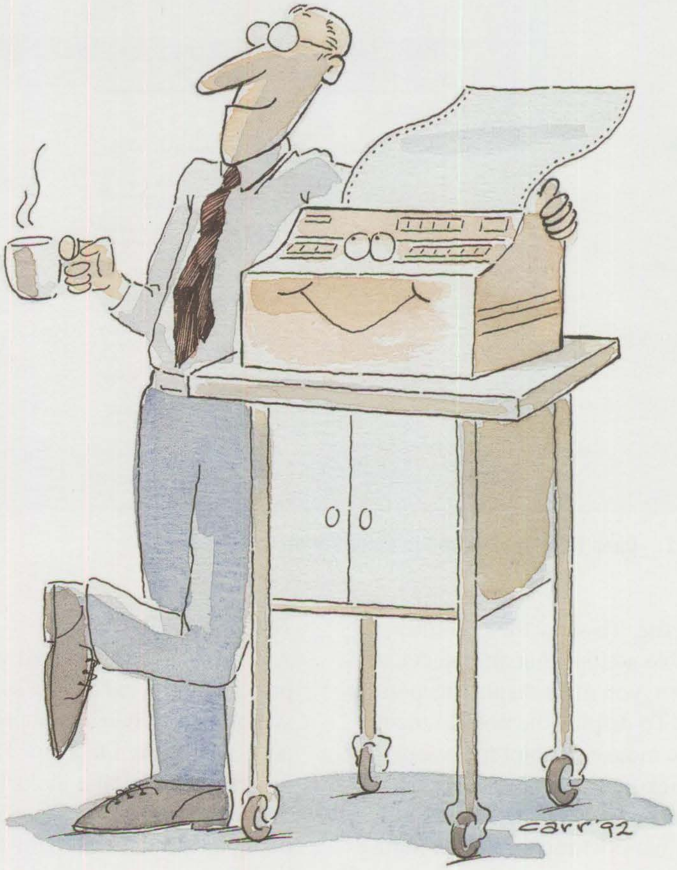
Printing from the Desktop

Using the drag-and-drop function, a data object can be dropped onto any print object, and it will be printed. Users can specify whether the data is plain text or printer-ready output. If it is printer-ready, which means it is a print job that is stored as a file, the job is spooled and printed. If the job is plain text, the job is routed through the printer driver, which then transforms the text into an output form that the printer can accept. Printing from within an application is easy using the application's printing interface or by selecting the Print option (if supported by that application) from that object's pop-up menu. Figure 1 shows the print option from a data file using drag and drop.

Printing Interfaces

In OS/2 1.X, printing devices and printing configuration were handled through the Print Manager, the Control Panel, or a combination of both, depending on the release of the operating system and which printer drivers were used. Understanding printing meant understanding the relationship among queues, drivers, ports, and their associations. In OS/2 2.0, most references to queues, print managers (in fact, all managers), and the Control Panel for configuring printer drivers, ports, and so on, have been intuitively consolidated under icons.

The Workplace Shell presents all system features as icons. An object is an icon that represents a folder (a location on the Desktop where data and program objects are stored), a program, a device (such as a printer), or



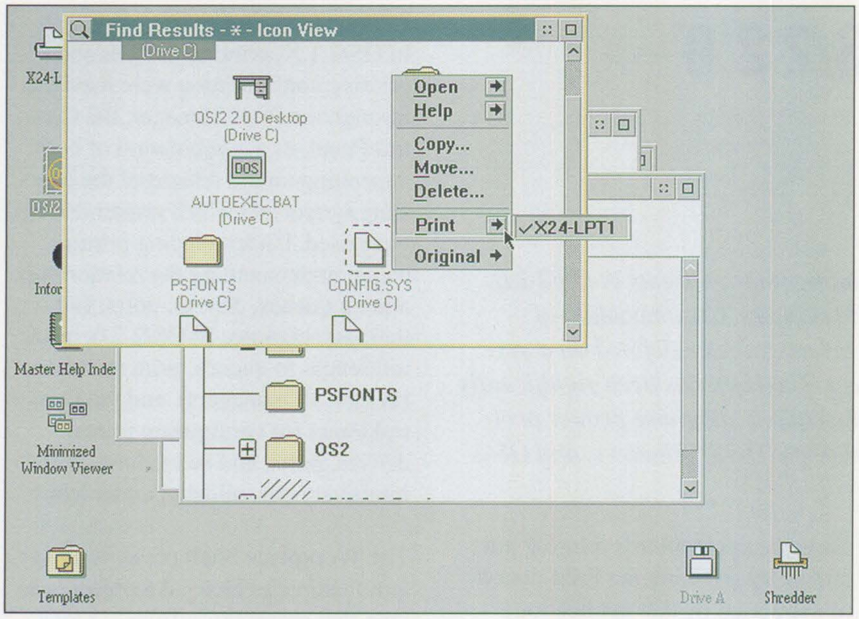


Figure 1. The Print Option Using Drag and Drop

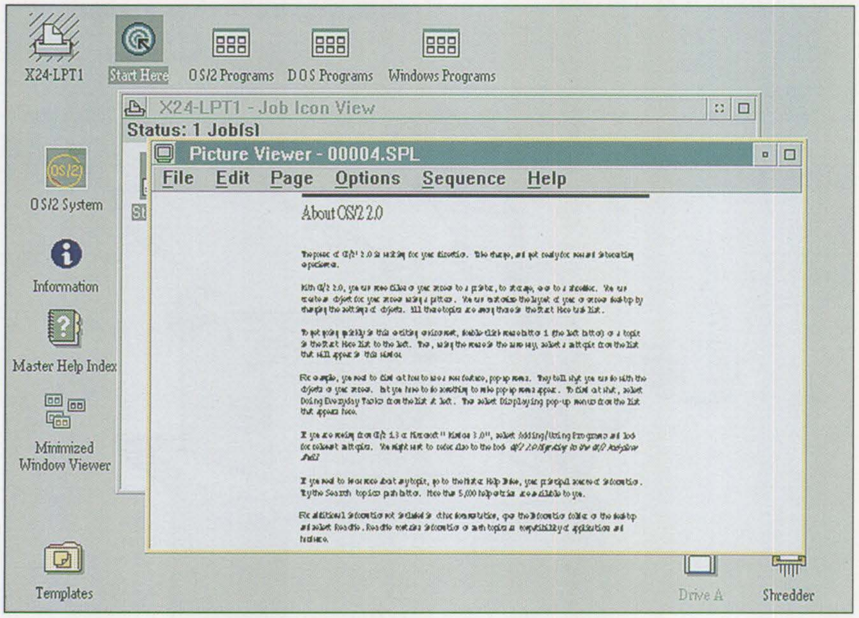


Figure 2. Open Print Job Viewed in Picture Viewer

a data file. To view the currently available actions that an object can perform, you must display its pop-up menu. To display the pop-up menu, use the mouse to point to the object, and click mouse button 2 once. To open or activate an object, select it by double-clicking on it with mouse button 1.

The Printer Object

A printer object is created when a printer is selected during system installation or when a new printer is added using the Desktop. The printer object – an icon that looks like a printer – is on the Desktop with all the other objects. Its many functions include storing a printer's setup, print spooling, and displaying print jobs and their related information.

Selecting the printer object will display, in icon view (the default), all print jobs waiting to be printed. After selecting (opening) a print object, print job information can be displayed and job status can be changed. The contents of a job, the details about a job (who sent it, when, and so on), and submission information about the job (number of copies, priority, required form, and queue options) can be displayed from an open print object. The print job can be held, copied, released, or printed next from an open print object.

To look at a job that has been sent to the printer but has not yet begun to print, open the printer object. Within the open printer object, the print job is represented as an icon. Next, select the print job's icon. The print job will then be displayed using the Picture Viewer utility. You can view individual pages, and you can either fit the displayed print job within the Picture Viewer window or make it full size. Using the Picture Viewer, you can also save the print job under a different name; it is stored by default as a spooler file `filename.SPL`. Figure 2 shows an open print job as seen in the Picture Viewer utility.

In OS/2 1.X, a print job was represented as a line of text in the Print Manager Visual Spooler. OS/2 2.0 offers text views of print jobs. For a text view of a print job instead of an icon, select Details View under the Open option of the printer object's pop-up menu.

Modifying Printer Object Settings

Selecting Open and Settings from the printer object's pop-up menu displays the printer object's notebook, containing information about the printer object. Inside the notebook, changes to various settings can be made: the system port to which the printer is connected, the type of paper in the printer, the resolution of the text, which printer driver is installed, and more. Figure 3 shows an open printer object's settings notebook.

The View tab is where the default view for the printer object is set. Here you can specify whether print jobs are displayed by default as icons or text.

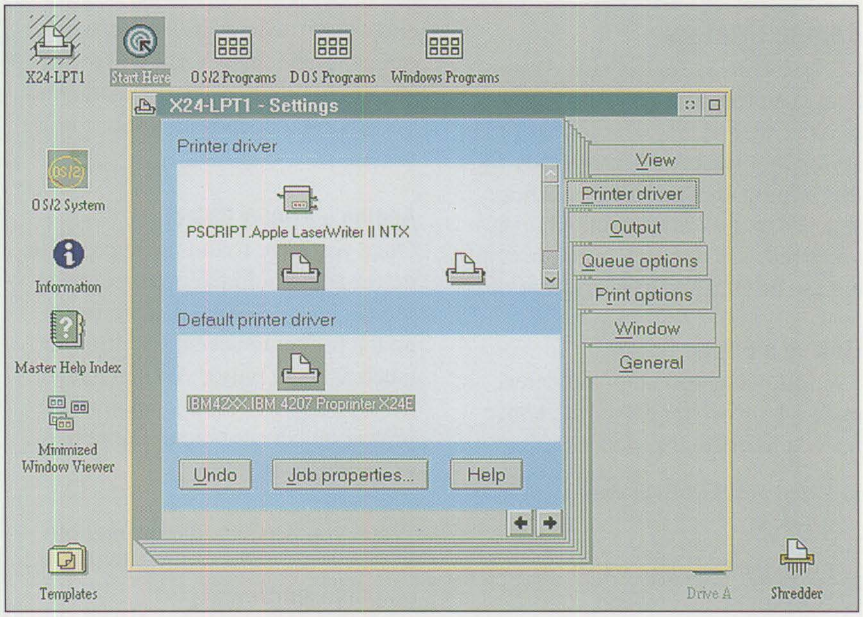
To view the printer drivers installed on the system and the selected printer driver for a particular object, select the Printer Driver tab. The default job properties (such as orientation, form, resolution, and font selections) are set through the Job Properties pushbutton inside the Printer Driver window. To view the default settings of a selected printer driver, select the driver's icon from the Printer Driver window.

From the driver's settings window, as in OS/2 1.X, it is possible to select and install soft fonts and printer-cartridge font definition files for use in specific printer models.

OS/2 2.0 supports up to four serial and three parallel ports for communications or printing devices. Configuring parallel or serial ports is done through a printer object, as follows:

1. Display the pop-up menu for any printer object.
2. Select Settings.
3. Select Output (port icons are displayed).
4. Select the icon for the port you want to configure. (Configuration changes affect whether the port is for printing or communications.)
5. Select Close from the System menu.

Selecting the Queue tab displays the icon for the queue processor (print job spooling software). The default queue processor is PMPrint. If a plotter is defined for use in the system and there is a need to remove overlapping fills, the PMPlot queue processor and its driver may need to be installed. Refer to the Help facility under "Plotter, Enabling Reverse Clipping" for additional information about plotting using the PMPlot queue processor.



X24-LPT1 - Settings

Printer driver

PSCRIPT.Apple LaserWriter II NTX

Default printer driver

IBM42XX.IBM 4207 Proprinter X24E

Figure 3. Settings Notebook in an Open Printer Object

Other selectable queue options include the Job Dialog option. This option is useful for changing the default form size, document orientation, resolution, and font in a print job before it is printed. The dialog appears only when the print option is selected from an object's pop-up menu or when using a drag-and-drop operation.

From the Print Options tab, the file name for a separator page – a file that is downloaded into a printer to separate print jobs in a networked printer environment – can be inserted and used between print jobs. Also, if the printer can be used only during a specified period, the start and stop times can be set from this window.

The Window and General tabs are used to set the Printer object's appearance, icon, and title.

Toggling the Spooler

The spooler is an object inside the System Setup object. In the default setup, the spooler is enabled. To disable the spooler, do the following:

1. Return to the Desktop.
2. Select the OS/2 System object.
3. Select the System Setup object.
4. The Spooler icon appears. Using mouse button 2, click once on the icon and select Disable. You will be prompted to reboot your system.

The Print Screen function, first available with an updated version of OS/2 1.3, can be enabled or disabled through the System object. The default setting is enabled; to disable it, do the following:

1. Return to the Desktop.
2. Select the OS/2 System object.
3. Select the System Setup object.
4. Select the System object.
5. Select the Print Screen tab.

Adding Printers

Occasionally a user may need more than one printer on a system, a network printer on a workstation, or the ability to produce a printer-ready file for printing on a printer that is not attached to the workstation. As in previous releases of OS/2, multiple printers can be attached to a workstation.

Finding a Printer Driver

The following will help you determine which printer driver to choose for your particular printer:

1. Place printer driver diskette 1 into drive A:.
2. From the Desktop, select the Drive A icon.
3. The files on printer driver diskette 1 will be represented as icons. Select the `PRDESC.LST` file.
4. A text file is displayed within the System Editor (the default setup for text files is an association, or special link, to the OS/2 System Editor), listing the printers and plotters supported by OS/2 2.0 and the specific driver name for each.
5. Once the correct printer driver is identified, select the file `PRDRV.LST` from the Drive A folder. The printer drivers and their corresponding diskette numbers are displayed in the System Editor.

Adding a Printer Driver

Once you know which printer driver to use and which diskette it is on, you can install an additional printer driver as follows:

1. Open a printer object's settings.
2. Select the Printer Driver tab.
3. Display the pop-up menu from the default printer driver.
4. Select Install.

Once the driver is installed, select it for use with the default printer object, or create a new printer object and select the new driver in the new object. There are several ways to create a new printer object. The method you choose depends on your printing needs.

Adding a Printer Object

Once you have found the appropriate driver and would like to direct print jobs to the additional printer, or set up the printer to accept specific forms (such as legal only), you can update the Desktop by creating another printer object containing the new print device or setup.

When you need another printer object with the same settings to direct output to a different port:

1. Select the Create Another option from the default printer object's pop-up menu.
2. Type the name for the new device.
3. Select the folder where you want this object to appear.
4. Select Create.
5. From the Create a Printer window, select the new port to which this object will direct output and select Create.

Use the Copy option for different printer settings (for example, a setup for landscape printing) when you have only one printer on one port. To copy an existing printer object:

1. Display the pop-up menu for the desired printer object.
2. Select Copy.
3. The OS/2 Desktop Copy window appears. Type in the name for the new printer object being created.
4. Select the folder where the new object should appear (the Desktop is the default).
5. Select Copy.
6. The printer object is created. Simply open the Settings window for this object.
7. Select the Printer Driver tab and open the printer driver you want to customize. The settings for that driver appear. From this window, you can set "landscape" as the default orientation.

If you have a printer object that is defined using LPT1 (such as an IBM 4207 printer), and you want to add a different printer to the COM1 port (for example, an IBM LaserPrinter), you can use either the template method or the drag-and-drop method to create a new printer object from which to select or install the different printer driver.

Follow these steps to create a new printer object using the template method:

1. From the Template folder located on the Desktop, grab the printer template and drop it onto a blank spot on the Desktop.
2. The system displays the Create a Printer window. Give the printer object a name or accept the system's default name.
3. Select the appropriate printer driver or install a new printer driver.
4. To install a new printer driver for a particular printer:
   a. Insert the diskette for the printer driver into drive A:.
   b. Display the pop-up menu for any printer driver icon, then select Install.
   c. The Install New Printer Driver window appears. Select Refresh; the driver files on the diskette in drive A: will be listed.
   d. Highlight the printer driver that supports the printer, and select the Install pushbutton. Once installed, a dialog box appears indicating that the printer driver

was installed successfully. Select OK in that dialog box.

5. Highlight the output port to which the printing device is physically or logically connected.
6. Select Create to display the new object.

Follow these steps to create a new printer object using the drag-and-drop method:

1. Insert the appropriate printer driver diskette into drive A:.
2. Select the Drive A object/icon from the Desktop.
3. Select the printer driver folder for your printer. In the next window that appears, select the model you want.
4. Drag the printer model from the window to either the Desktop or another folder.
5. Customize the printer properties to match those of your printer.
6. Select the port to which the additional printer is attached.

As a result of dragging and dropping the printer driver onto the Desktop, the driver is installed and a new printer object is created simultaneously.

Deleting a Printer Object or Driver

Because printing devices are represented as objects on the Desktop, removing a printer object from the Desktop is easier than removing a printer from your office. To remove a printer object from the Desktop, display the printer object's pop-up menu and select Delete. Deleting the object also removes its associated queue from the system.

After a printer object has been deleted, its printer driver is still accessible from within another printer object. Take the following steps to delete a printer driver in OS/2 2.0.

1. Deselect the driver from any print objects within the object's settings notebook. *Note*: If a driver has been loaded into memory (that is, used) since the system was booted, the driver cannot be removed. As with OS/2 1.X, you will have to reboot the system to clear memory before the driver can be removed.
2. Display the driver's pop-up menu and select Delete.

Printing from DOS Sessions

Printing from a DOS application in a Multiple Virtual DOS Machines (MVDM) environment is done the same way as printing in straight DOS. DOS applications use their own printer drivers supplied with their product packages. These must be installed just like in a regular DOS environment.

DOS applications running in the MVDMs and printing to an LPTX port have their print jobs routed through the OS/2 print spooler if it is enabled (which is the default). Capturing DOS print jobs in the OS/2 spooler prevents intermixing print jobs sent from multiple DOS sessions.

Printing from WIN-OS/2 Sessions

The WIN-OS/2 feature in OS/2 2.0 offers the ability to run most Windows programs that are written exclusively for the Windows Application Programming Interface (API) and do not jeopardize the system integrity of OS/2 2.0, as stated in the Help facility. Printing from Windows applications is improved in OS/2 2.0. The single-task print spooler in Windows is replaced with the multitasking spooler in OS/2 2.0.

If Windows support was installed during OS/2 system installation and a default printer was selected, the matching Windows printer drivers are installed automatically. When a new printer driver is added through OS/2 after system installation, there is the option to install the corresponding WIN-OS/2 printer driver.

The Help under WIN-OS/2 Printers has a list of printer drivers for WIN-OS/2 that must be installed through the WIN-OS/2 Control Panel after the operating system is installed. These drivers are for printers that do not have a corresponding OS/2 printer driver.

After the WIN-OS/2 printer drivers are installed, a printer object for WIN-OS/2 printing can be set up. The "Setting Up a WIN-OS/2 Printer" topic in the Help facility guides you through setting up a printer object for Windows printing using the IBMNULL driver. The recommended setup of using the Windows-supported printer driver and the IBMNULL driver enables print jobs to be spooled and printed with other Desktop print jobs while also being properly formatted for a printer that is not supported under OS/2.

When defining printers, through either the application or the Windows Control Panel, select the ports under WIN-OS/2 that have an OS2 suffix. For example, in the Configure option under the Printers object in the WIN-OS/2 Control Panel, select LPT1.OS2 or LPT2.OS2 as your printer port to take advantage of the job spooling in OS/2.

__Jean N. Shortley__ is a senior associate programmer in IBM's Entry Systems Technology Laboratory in Boca Raton, Florida. She was formerly a member of the printer device-driver development and test teams and the Multiple Virtual DOS Machines test team. Jean holds a BS in education and an MEd in educational microcomputer research, both from Florida Atlantic University in Boca Raton.

Installing the IBM 4029 LaserPrinter Under OS/2 1.3

Richard R. Miller
IBM Corporation
Raleigh, North Carolina

This article contains step-by-step instructions for installing the IBM 4029 LaserPrinter on an OS/2 1.3 workstation or an OS/2 1.3 LAN print server. Each section can help users customize existing printers and solve problems. See a related article, "Printing from OS/2 2.0," in this issue for a description of improvements made to the Print Manager.

The following instructions help you install the IBM 4029 Laser Printer on a workstation or print server running OS/2 1.3 or 1.30.1. They also show how to install the Automatic Emulation Switching (AES) option and the Adobe Type Manager® (ATM) PostScript® fonts. Follow these instructions in sequence to avoid omitting any steps.

Workstations using the 4029 printer must have the `IBM4019.IBM` and `PSCRIPT.IBM 4019 V52_1 (17 Fonts)`[1] printer drivers installed. You can call Lexmark® at (606) 232-3000 to obtain a diskette containing 4029-specific drivers that use the full function of the printers. Refer to the OS/2 installation procedures to install the required print drivers from the OS/2 diskettes if they are not available on your workstation or server.

Users running OS/2 1.30.2 or 2.0 will find the 4029-specific driver packaged within the new `IBM4019.DRV` driver. Installation procedures for 1.30.2 users are similar to the procedures shown here, but instead of selecting "IBM4019", select "IBM 4019.IBM 4029 LaserPrinter 10." OS/2 2.0 users should refer to "Printing from OS/2 2.0" in this issue.

As you proceed, keep in mind that COM1 refers to a serial adapter and LPT1 refers to a parallel adapter.

Installing the AES Option

The first item to install on the OS/2 1.3 workstation is the 4029 AES option. This feature enables a single 4029 to be accessed as both a Personal Printer Data Stream (PPDS) printer and a PostScript printer; it also installs the printers and queues in the Print Manager. The required diskette (part number 1183018) for this installation is in the Supplemental Utilities package shipped with the IBM 4029 LaserPrinter PostScript option. Take the following steps to install the AES option:

1. If you are installing the printer using the Printer Sharing option or the COM1 connection, edit the workstation's `CONFIG.SYS` file to include the following device statement:

   ```
   DEVICE=C:\OS2\COM02.SYS
   ```

   Note: The path must match the location of `COM02.SYS` on your workstation. After adding the `COM02.SYS` statement to `CONFIG.SYS`, reboot the workstation so that the new `CONFIG.SYS` takes effect.
2. Stop LAN Requester if it is running.
3. Complete any printing.
4. Stop or pause any jobs that will generate printing.
5. If Print Manager is running, use the following steps to stop the Print Manager Spooler:
   - Press Ctrl+Esc.
   - Select Print Manager.
   - Select Setup from the action bar in Print Manager.
   - Select Spooler.
   - Select Disable.
   - Select Set.
   - Select OK.
6. Take these steps to shut down and reboot the workstation.
   - Press Ctrl+Esc.
   - Select Desktop Manager.
   - Select Desktop.
   - Select "Shutdown the Desktop" from the action bar.
   - Select Shutdown.
   - Select Yes to terminate all running tasks. When all disk activity has ceased and the message window appears, reboot the workstation by pressing Ctrl+Alt+Del.
7. If the LAN Requester was restarted by rebooting, issue the `net stop req` command from a full-screen prompt before continuing.
8. Start an OS/2 session.
9. Insert the Supplemental Utilities for the IBM LaserPrinter diskette in drive A:.

[1] PSCRIPT.IBM 4019 V52_1 (17 Fonts) is the name that is displayed for selection in the Print Manager.

10. Type `a:` and press Enter.
11. Type `install a:` and press Enter.
12. Select Continue.
13. Read the information window, then select OK.
14. Read the information window. The requested actions were performed in steps 1 through 6, so press Enter to continue. Select Yes; the 4029 has PostScript and AES features installed. The AES Installation screen is displayed. Select the following options:
    - LPT1: 4029 LaserPrinter
    - LPT2: PostScript
    - LPT3: None
    - Printer physically connected to LPT1 (or COM1 for the Printer Sharing option)
    - Baud 9600 (this appears after COM1 is selected)

    Note: If you currently use the LPT1 or COM1 port on your workstation, you must change the previous settings to match your environment. Refer to *Supplemental Utilities for the IBM LaserPrinter* (SA40-0601) for information about changing the configuration. The COM1 port is required when using the Printer Sharing option.

    - The default installation path is `C:\AES`. You can change it if you want to install AES in a different drive or subdirectory.
    - Select OK.
15. Select Yes to change Print Manager queues for AES.
16. Select OK if the `IBM4019.IBM` print driver is highlighted for the 4029 PPDS printer.
17. Select the `PSCRIPT.IBM 4019 V52_1 (17 Fonts)` driver for PostScript.
18. Select OK.
19. Select Yes for AutoStart at power-on.
20. If you are using the Printer Sharing option, ignore the information in the next window and select OK. The physical printer is actually a parallel connection at the sharing device, so this action is not needed at the workstation.
21. Select OK on successful installation.

Installing ATM Fonts

The following diskettes are needed to install the ATM fonts, and are included with the 4029 LaserPrinter PostScript option.

- Additional ATM Fonts; Supplemental PostScript Font Software (part number 1195125)
- Additional ATM Fonts; PPDS Disk 1 (PN 1195093)
- Additional ATM Fonts; PPDS Disk 2 (PN 1195121)

Follow these steps to install the ATM fonts:

1. Press Ctrl+Esc, then select Desktop Manager.
2. Select Utilities.
3. Select Control Panel.
4. Select Installation.
5. Select Add Fonts.
6. Insert the Supplemental PostScript Font Software diskette in drive A:, then select Add.
7. Select all fonts (highlight with mouse).
8. Verify drive and path, and change if necessary.
9. Select Add.
10. Repeat the previous steps for the two remaining diskettes (PPDS Disks 1 and 2).

Starting the Print Manager

Follow these steps to start the workstation Print Manager if it is not active. Otherwise, proceed to "Printer Setup."

1. Press Ctrl+Esc.
2. Select Group-Main.
3. Select Print Manager.
4. Select Setup from the action bar in Print Manager.
5. Select Spooler.
6. Select Enable.
7. Verify Spooler Path (`C:\SPOOL`).
8. Select Set.

Printer Setup

1. Select Setup from the action bar in Print Manager.
2. Select Printer. Two printers were added during AES installation: Printer1 and Printer2.
3. Use the arrow keys to select Printer1.
4. Select Change.
5. Type a name of up to eight characters for your print queue, such as LASER, LASER20, or PPDS20.
6. Enter a description in free-form text, such as "4029 ASCII Printer." This is the name that will be displayed in the printer selection for most applications.
7. Leave Retry at 45 seconds.
8. If you are configuring the 4029 LaserPrinter on a stand-alone workstation, verify that the `IBM4019.IBM` print driver is selected for LPT1. If installing on a LAN print server, the device should be LPTX as connected to your server. If you are using the Printer Sharing option, the device should be LPT1, and LPT1 must be selected for LASER or PPDS printer.

9. Select Printer Properties. The following options are suggested for the non-PostScript PPDS or ASCII data-stream printer (Printer1):
   - Set Resolution to 300 x 300.
   - Set Defined Forms to Letter (8.5 x 11 inches).
   - Select Envelope Feed, if installed.
   - If the 500-sheet feeder is installed, select Two Paper Trays.
   - Select Orientation. Portrait is recommended.
   - Set Memory Option to 3.5 MB, if installed.
   - Set Default Spool File Type to PM_Q_STD.
   - In the Forms area:
     - Select Connections.
     - Connect Paper Tray 1 to Letter (8.5 x 11 inches).
     - Connect Paper Tray 2 to Letter (8.5 x 11 inches).
     - Connect Envelope to 10 (4.125 x 9.5 inches). To change the recommended connections, consult the *Technical Reference for the IBM LaserPrinter* (SA40-0559-00). Pages 3-1 and 3-2 describe the paper and envelope sizes to use.
     - Select OK.
10. Select Default Fonts. The Available Default Fonts window will be displayed. To print at 80 characters per line, use 437 12.0 Courier 10 as the default. To print at 132 characters per line, select 437 8.5 Courier 17.1. To install additional 4029 fonts, follow the instructions that come with the additional font package.
11. Select OK to accept font changes.
12. Select OK to accept resolution and orientation changes.
13. Select Change to accept all changes for Printer1.
14. If the warning message for printer properties change appears, select OK.
15. Select Printer2.
16. Select Change.
17. Type a name of up to eight characters for the print queue, such as POSTSCR, PS4029, or PSONE.
18. Enter a description in free-form text, such as "4029 Printer" or "Printer in Room 10."
19. Check that the print driver is `PSCRIPT.IBM 4019 V52_1 (17 Fonts)`.
20. Select Printer Properties. The following options are suggested for the PostScript printer (Printer2):

Tray	Form
Upper	Letter
Lower	Letter
Envelope	Envelope.297.684/C10 Envelope

21. Select OK. Do not change the Job Properties unless you understand the interactions and effects on the PostScript jobs intended to print on the workstation.
22. Select Change.
23. Select OK on the warning message for printer properties change.
24. Select Change to accept all changes. If you have added a printer or plotter as Printer3, customize it now.
25. Select OK when all changes are complete.

Changing the Print Queue

Before you customize a workstation's OS/2 print queues, Print Manager must be active. If it is not, take the following steps to start it. If Print Manager is active, skip to the section titled "Print Queue Setup."

1. Press Ctrl+Esc.
2. Select Group – Main.
3. Select Print Manager.
4. Select Setup from the action bar in Print Manager.
5. Select Spooler.
6. Select Enable.
7. Verify Spooler Path (`C:\SPOOL`).
8. Select Set.

Print Queue Setup

From the action bar in Print Manager, do the following:

1. Select Setup.
2. Select Queues. Two queues were added during the AES installation: LPT1Q and LPT2Q.
3. Select LPT1Q, the queue to be modified.
4. Select Change.
5. Type a name of up to eight characters, such as LASER08 or PS40294.
6. Enter a description in free-form text, such as "4029 ASCII Printer" or "4029 Printer in Room A212."
7. Check that the correct printer and driver are selected:
   - Printer1: IBM LaserPrinter 4029
   - Print Driver: `IBM4019.IBM`
8. Select Job Properties. Under Printer1 options, do the following:
   - Verify that Orientation is set to Portrait.
   - Set Resolution to 300 x 300.
   - Select OK.
   - Select Change.
9. Select LPT2Q, the next queue.
10. Select Change.

11. Type a name of up to eight characters; such as POSTSCR or PS4029x.
12. Enter a description in free-form text, such as "4029 PostScript Printer 1" or "4029 Printer in Room A214."
13. Check that the correct printer and driver are selected:
    - Printer2: 4029 PostScript
    - Print driver: `PSCRIPT.IBM 4019 V52_1 (17 Fonts)`
14. Select Job Properties. Under PostScript Job Properties:
    - Verify that Orientation is set to Portrait.
    - Verify that Scaling is 100 percent.
    - Verify that Form is Letter.
    - Select OK.
    - Select Change.
15. If you have an additional queue, go back to step 9.
16. Select OK when all print queue changes are complete.

If you are installing the 4029 printer on a stand-alone workstation, skip to the section titled "Print Test."

Verifying COM1 Options

This section helps set up a workstation's communications port for the Printer Sharing option. Skip these instructions if your printer is attached to the LPT1 port on a print server or workstation and if it is not using the 4029 Printer Sharing option.

Take the following steps to verify that a workstation's COM1 options are correct:

1. Select Desktop Manager.
2. Select Utilities.
3. Select Control Panel.
4. Select Options.
5. Select Communications Port.
6. Set options to the following:
    - COM1
    - 9600 Baud
    - Word Length 8
    - No Parity
    - 1 Stop Bit
    - Hardware Handshake
7. Select Set.
8. Close Control Panel.
9. Close Utilities.

If you are installing the 4029 printer using the Printer Sharing option, but are not installing on a print server, skip to section titled "Print Test."

Installation on a Print Server

To install the 4029 printer on a print server, log on with a LAN Administrator ID. (Any LAN Requester on the domain is acceptable; it is not necessary to log on to the server you are customizing.) Be sure to complete the previous sections before taking the following steps to access the LAN Requester for the LAN-attached printer.

1. Type `net` from an OS/2 full-screen C: prompt.
2. Select Definitions and press Enter.
3. Select Aliases and press Enter.
4. Select Printers and press Enter.
5. Select New by pressing the space bar.
6. Press F10.
7. Select Actions and press Enter.
8. Select Create and press Enter.
9. Type a name of up to eight characters for your alias, such as LASER0X or PS4029X.
10. Enter a description in free-form text, such as "4029 ASCII Printer on Print Server 27" or "4029 Printer in Room A212."
11. Enter the server name where the 4029 LaserPrinter AES and PostScript software (ATM fonts) are installed.
12. Enter the queue name previously defined in the section "Changing the Print Queue."
13. Leave Maximum Users blank to allow all users to access the printer.
14. Set When Shared to At Server Startup. Press the space bar to select. An arrow indicates selection.
15. Press Enter. You will be returned to the Manage Aliases-Printers screen. Select New and repeat the procedure to add an alias.
16. Take the following steps:
    - Select one of the aliases just created with the space bar.
    - Press F10.
    - Select Access Profile.
    - Type CP in the Universal Access Permission field and press Enter.

    Repeat the preceding steps for each alias created to allow all LAN requesters to access the printers.
17. After defining the last alias, take the following steps:
    - Press F10.
    - Select Exit and press Enter.
    - Press Shift+F3 to exit LAN Requester.
18. Log off the LAN Requester ID. The printer has been added to the LAN server.

Adding a Printer to a Workstation

Follow these steps to add a printer to each workstation that will print on a LAN-attached 4029 printer.

1. Press Ctrl+Esc if Print Manager is not the current task.
2. Select Print Manager.
3. Select Setup.
4. Select Printer.
5. Select Add to add the Text or PPDS 4029 LaserPrinter.
6. Type a name of up to eight characters, such as LASER or LASER20.
7. Enter a description in free-form text.
8. Device should be LPTX. LPT4 is recommended for the Text printer on the LAN.
9. Leave Retry at 45 seconds.
10. Check that the print driver `IBM4019.IBM` is selected for LPTX.
11. Select Printer Properties. The following options are suggested for the non-PostScript PPDS or ASCII data-stream printer:
    - Set Resolution to 300 x 300.
    - Set Defined Forms to Letter (8.5 x 11 inches).
    - Select Envelope Feed, if installed.
    - If the 500-sheet feeder is installed, select Two Paper Trays.
    - Select Orientation. Portrait is recommended.
    - Set Memory Option to 3.5 MB, if installed.
    - Set Default Spool File Type to PM_Q_STD.
    - In the Forms area:
      - — Select Connections.
      - — Connect Paper Tray 1 to Letter (8.5 x 11 inches).
      - — Connect Paper Tray 2 to Letter (8.5 x 11 inches).
      - — Connect Envelope to 10 (4.125 x 9.5 inches). To change the recommended connections, consult the *Technical Reference for the IBM LaserPrinter* (SA40-0559). Pages 3-1 and 3-2 describe the paper and envelope sizes to use.
      - — Select OK.
12. Select Default Fonts. The Available Default Fonts Window will be displayed. To print at 80 characters per line, use the 437 12.0 Courier 10 font. To print at 132 characters per line, select the 437 8.5 Courier 17.1 font. To install additional 4029 fonts, follow the instructions that come with the additional font package.
13. Select OK to accept font changes.
14. Select OK to accept resolution and orientation changes.
15. Select Change to accept all changes for Printer1.
16. Select Add to add the PostScript LAN 4029.
17. Type a name of up to eight characters, such as POSTSCR, PS40290, or PSONE.
18. Enter a description in free-form text.
19. Device should be LPTX. LPT5 is recommended for the PostScript printer on the LAN.
20. Leave Retry at 45 seconds.
21. Check that the print driver is `PSCRIPT.IBM 4019 V52_1 (17 Fonts)`.
22. Select Printer Properties. The following options are suggested for the PostScript printer.

Tray	Form
Upper	Letter
Lower	Letter
Envelope	Envelope.297.684/C10 Envelope

23. Select OK. Do not change the Job Properties unless you understand the interactions and effects on the PostScript jobs intended to print on the workstation.
24. Select Change.
25. Select OK on the warning message for printer properties change.
26. Select Change to accept all changes. If you have added a printer or plotter as Printer3, customize it now.
27. Select OK when all changes are complete.

Accessing Printers by Alias

Take the following steps for each workstation that will access the LAN printer. If you are setting up workstations for other users, leave a copy of these instructions at each workstation so that users can change, add, or delete printer assignments easily. To access the printers on a print server by alias, each workstation user must have either temporary or permanent access to the printer.

Temporary Access

A user must be logged on to the LAN as a Requester (user) to share a printer attached to a print server. The following steps provide access to the printers on a print server for the duration of one LAN session, and allow users to reaccess the printers after a print server restart. Users who retain the same printer assignments for each LAN logon should skip to the section titled "Permanent Access." The temporary procedure does not affect permanent access, and can be used whenever a user is logged on to the LAN.

1. Type `net` from an OS/2 full-screen C: prompt.
2. Select Actions and press Enter.
3. Select Resource Assignments and press Enter.
4. Select Printers and press Enter.
5. Enter the printer port to be assigned (the default is LPT1).
6. Use the Tab key to highlight the printer.
7. Press the space bar to select the printer.
8. Press Enter. A window will show the assignment.
9. Press Enter.
10. Press Esc to exit temporary assignments.
11. Select Exit and press Enter.
12. Press Shift+F3 to exit LAN Requester. The assignments made using this procedure are active until the user logs off the LAN or the print server is shut down.

Permanent Access

When creating a user ID, a LAN administrator can provide access to printers by using local procedures defined for File, Application, and Printers. The administrator should provide a copy of the following instructions to users so they can make modifications due to equipment or location changes.

To give a user permanent access to a printer attached to a print server, take the following steps. If the print server is not running at logon time, an error message will indicate that the user could not access the printer.

1. Type `net` from an OS/2 full-screen C: prompt.
2. Select Definitions and press Enter.
3. Select Logon Details and press Enter.
4. Press F10.
5. Select Actions and press Enter.
6. Select Printer Assignments. Scan the list of printers and type the LPTX port in the port field on the left side of the screen. It is possible to select multiple printers at this time, but there can be only one printer for each port.
7. Press Enter after selecting printers.
8. Press F10.
9. Select Exit and press Enter.
10. Press Shift+F3 to exit LAN Requester.
11. Press Enter. The assignments made using this procedure are obtained when a user logs on to the LAN and the print server is running. If the print server was not running when the user logged on, or the print server has been shut down and restarted, the user can employ the temporary access procedure to reaccess the LAN printers.

Print Test

A user of a stand-alone workstation can verify that a printer is properly installed by printing the `CONFIG.SYS` file on the 4029 printer in PPDS mode, as follows:

1. Press Ctrl+Esc.
2. Select an OS/2 window or full-screen task.
3. At the C: prompt, type `copy config.sys lpt1`.
4. Press Enter.

To install the 4029 printer on a LAN print server, you must be logged on as a LAN Requester to access LAN-connected 4029 LaserPrinters from your workstation. Entering the `net use` command at the C: prompt on the workstation displays a list of network devices accessed at the time the command is entered. If no network printers are displayed, use the following command to access a printer for this LAN Requester session:

```
net use lptx: aliasname
```

The x is the printer you want to use. The `aliasname` is the alias defined for the printer in the "Installation on a Print Server" section. The following command entered from an OS/2 session C: prompt will print the workstation's `CONFIG.SYS` on the 4029 printer.

```
copy config.sys lptx
```

***Richard R. Miller** is a staff systems analyst in IBM U.S. Quality Research and Development in Raleigh, North Carolina. For the last six years, Richard has been working with VM and MVS Advanced Function Printing, DOS and OS/2 workstation printing, and IBM Token-Ring printer hardware and software installation, usability, and support. His previous experience includes positions as customer engineer, program support representative, Level 2 support representative, PC application development, and VM system programmer.*

Serviceability Tools in OS/2 2.0

Allen M. Gilbert
IBM Corporation
Boca Raton, Florida

OS/2 2.0 provides a broad set of software tools that enables users, service personnel, and software developers to diagnose problems that occur in application programs and in the operating system. These tools support such functions as error logging, event tracing, resource analysis, and system dumping. This article provides an overview of the different OS/2 2.0 serviceability tools, shows users how to enable these tools, and outlines how application developers can take advantage of these valuable functions.

Serviceability means the ability to fix a defect – quickly, efficiently, and permanently – the first time the defect occurs. As a critical part of the operating system, serviceability tools have been included in OS/2 since its first release. The serviceability tools in OS/2 2.0 offer new levels of sophistication, including utilities and Application Programming Interfaces (APIs) for use within OS/2 application programs. This article primarily discusses the utilities, but also provides a brief overview of the serviceability APIs.

The OS/2 2.0 serviceability utilities are highly technical, providing a wealth of detail about the operation of OS/2 2.0 systems. They are intended for knowledgeable users, such as trained OS/2 support persons, experienced OS/2 application developers, or system administrators. Even so, this article provides background suitable for all OS/2 users.

OS/2 2.0 serviceability tools support two general serviceability functions:

First failure data capture: Data relating to a software or hardware failure is collected and retained at the time the failure is detected. If all the data relevant to a failure were captured upon detection, the cause of the failure could be deduced from the data. Unfortunately, many operating systems fail to collect sufficient data to support post-failure analysis. In such cases, developers or support persons try to reproduce the failure – an expensive and often futile effort.

General system monitoring: This approach does not collect the data directly related to a failure, but provides views of a functioning software system. It is often useful to peek into a functioning software system and quantitatively view how that system is operating. For example, a system administrator may want to study the sequences of key internal and external events through which the system moves, or to summarize the way that the system uses resources as it executes.

There are four serviceability facilities within OS/2 2.0:

- Error logging
- Event tracing
- System dumping
- Resource analysis

While OS/2 2.0 serviceability tools can be valuable to programmers during software development, they are intended mainly for analyzing problems that occur during normal OS/2 system operation.

Error Logging

The OS/2 2.0 System Error Logging facility is a single focal point for logging software and hardware errors. This facility is available to both the operating system and OS/2 application programs. Software modules can log error entries at any time.

Entries are logged in a central Error Log file. This file can be on the user's system or on an OS/2 server. OS/2 2.0 Error Log entries consist of a small standard set of error header information and a variable-length packet of error data that is unique to a particular class of error entry. The OS/2 2.0 System Error Logging facility includes a Presentation Manager®-based Error Log Formatter that can be used to browse the Error Log file.

When run in conjunction with the OS/2 Communications Manager subsystem, the OS/2 2.0 System Error Logging facility can generate Systems Network Architecture (SNA) Generic Alerts. Generic Alerts are specially formatted Error Log entries transported across an SNA network, and are interpreted and displayed at control nodes on the SNA network. Generic Alerts allow central SNA network administration personnel to remotely monitor errors on individual OS/2 workstations.

A software module that generates an Error Log entry chooses whether to format the entry as a Generic Alert. All Generic Alert Error Log entries are logged to the central Error Log file. If a software module formats an Error Log entry as a Generic Alert and the OS/2 2.0 system does not include the Communications Manager subsystem, no Generic Alert is transmitted.

The OS/2 2.0 System Error Logging facility has three main components:

Error Logging Device Driver (`LOG.SYS`): This collects the Error Log entries generated by various software modules. It forwards the Error Log entries to the Error Logging process when the process requests them. The Error Logging process then writes the Error Log entries to the system Error Log file. The Error Logging process acts as a daemon process, performing its activities in the background.

Error Logging Process (`LOGDAEM.EXE`): The system Error Log file can grow to a user-specified limit. Once the user-specified limit is reached, the Error Log file adds new Error Log entries by "wrapping around" (deleting the oldest entries to make room for new ones).

The Error Log Formatter and Control Program (`SYSLOG.EXE`): The Error Log Formatter is for viewing the Error Log file and controlling the execution of the Error Logging facility. Users can run it as a Presentation Manager utility or as a batch command within an OS/2 command file or REXX program. The *OS/2 Online Command Reference* fully explains its use.

Enabling the Error Logging Facility

To enable the Error Logging facility on an OS/2 2.0 system, users must first install the three components above, then modify the `CONFIG.SYS` file to indicate that `LOG.SYS` and `LOGDAEM.EXE` are to be initiated when the operating system boots. There are three ways to ensure that these components are installed:

- Choose the Install All Features option during system installation.
- Choose the Install Preselected Features option during system installation.
- Choose the Select Features and Install option during system installation, and select the Serviceability and Diagnostic Aids installation option.

To initiate `LOG.SYS` and `LOGDAEM.EXE` when the operating system boots, add the following two statements to the `CONFIG.SYS` file (these statements assume the two executable files are in the `C:\OS2` directory; if they are not there, modify the statements):

```
DEVICE=C:\OS2\LOG.SYS
RUN=C:\OS2\SYSTEM\LOGDAEM.EXE
```

These statements do not illustrate the various options that can be specified. For example, the `RUN=` line can also be used to set the maximum size of the System Error Log file. The *OS/2 2.0 Online Command Reference* describes the different options available for the two error logging `CONFIG.SYS` statements.

Viewing the Error Log

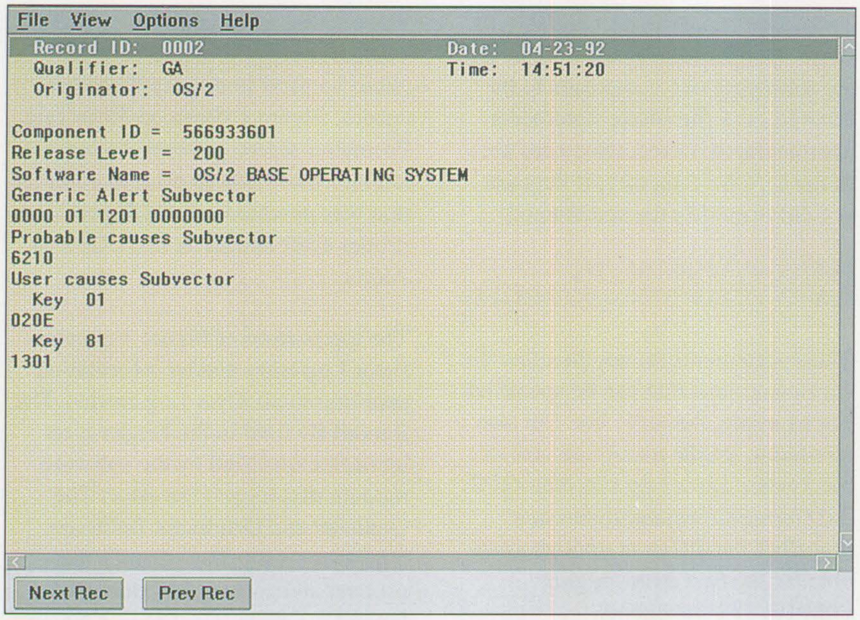
The Error Log Formatter uses its client window to display Error Log entries. Figure 1 shows an Error Log Formatter screen generated when an attempt was made to print to a printer that was powered off. OS/2 generates printer error entries as SNA Generic Alerts.

The top portion of Figure 1 contains Error Log entry header information common to all Error Log entries. The Record ID field in the header is an identifier specified by the software module that logged the error. The Qualifier and Originator fields are character strings that identify the product and software module that logged the error. The Process Name field contains the full OS/2 path name of the process that logged the error. The Date and Time fields refer to the date and time the error was logged. The Formatted By field indicates which Dynamic Link Library (DLL) file, if any, was used to format the Error Log entry. The bottom portion of Figure 1 contains information unique to the particular class of error.

In the header of the entry in Figure 1, the Qualifier field is marked as "GA," indicating a Generic Alert, and the Originator field indicates "OS/2." Within the data portion of the Error Log entry, the Software Name field is marked as "OS/2 BASE OPERATING SYSTEM" and the Release Level field indicates "OS/2 2.0."

The SNA Alert Management Services (MS) subvectors shown in Figure 1 are described below. A *subvector* is a package of information that fits within a larger message (or *vector*).

Generic Alert Subvector: Indicates permanent loss of availability not directly initiated by a user action (x0000 01 1201).



```
  Record ID:  0002                                 Date:  04-23-92
  Qualifier:  GA                                   Time:  14:51:20
  Originator:  OS/2

Component ID =  566933601
Release Level =  200
Software Name =  OS/2 BASE OPERATING SYSTEM
Generic Alert Subvector
0000 01 1201 0000000
Probable causes Subvector
6210
User causes Subvector
  Key  01
020E
  Key  81
1301
```

Figure 1. Error Log Formatter

Probable Causes Subvector: Indicates a printer output device error (x6210).

User Causes Subvector: Indicates that the printer was powered off (x020E) and that the recommended action is to ready the device and then retry (x1301).

Systems Network Architecture Formats (GA27-3136) describes Generic Alert subvectors. Typically, OS/2 support personnel interpret the Error Log.

The Error Log file is a list of chronological Error Log entries. The first Error Log entry is the newest and the last entry is the oldest. At the bottom of the window are two pushbuttons (Next Rec and Prev Rec) for moving forward or backward through the file.

The Error Log Formatter utility provides a set of menu options that allows users to control both Error Log formatting and the Error Logging process. Normally, the Error Log Formatter is used to view the currently active system Error Log file (the file where `LOGDAEM.EXE` is currently logging error entries).

Figure 2 shows the File, View, and Options menu items.

Event Tracing

The OS/2 2.0 Event Tracing facility can track the occurrence of key hardware and software events over time. An *event* is any state that can be detected within a software module and that executes within a particular section of the module.

The OS/2 2.0 Event Tracing facility allows software modules to place event tracing function calls at locations that represent significant state transitions. Because event tracing can increase system overhead, the OS/2 user can specify which events to trace. When an event occurs that is being traced, the function call associated with the event logs event-related data to an OS/2 system buffer. OS/2 provides a Trace Formatter utility so that the data can be formatted for display.

OS/2 2.0 supports two classes of function calls for event tracing:

Static tracepoints: These are OS/2 APIs placed within software modules. Each static tracepoint is associated with a major trace code and a minor trace code. An OS/2 user can turn individual static tracepoints on and off by specifying major and minor trace codes. Even when a user has not enabled them, static tracepoints require a small amount of overhead. This happens because, when a static tracepoint API call is encountered, the API must determine whether it is currently enabled.

Dynamic tracepoints: These are defined within Trace Definition Files, independent of the software module. Dynamic tracepoints can be defined for both executable (`.EXE`) files and for Dynamic Link Library (`.DLL`) files. The Trace Definition File associated with a software module contains the same root name as the corresponding `.EXE` or `.DLL` file. It has the special file suffix `.TDF` and contains all the dynamic tracepoints defined for that software module.

Sets of dynamic tracepoints can also be grouped together through the use of common Type and Group identifiers. These groupings let users easily enable or disable groups of related dynamic tracepoints. For example, the OS/2 Kernel dynamic tracepoints are divided into API and INT (Internal) Types. The API dynamic tracepoints are associated with the processing of OS/2 API calls. The INT dynamic tracepoints are associated with key OS/2 Kernel internal events.

A dynamic tracepoint can be included within more than one Type identifier. Most OS/2 APIs have two dynamic tracepoints defined for them: one tracepoint when the API is invoked and a second tracepoint at the time the API returns its results. The two classes of API dynamic tracepoints are associated with the PRE and POST Type identifiers. The PRE API tracepoints log the key parameters that call the API. The POST API tracepoints log the return code and parameters returned by the API.

The different OS/2 2.0 dynamic tracepoints are also divided into various tracepoint Groups. These Groups include SEM (semaphore-related tracepoints), FS (file system-related tracepoints) and TK (task management-related tracepoints).

When an OS/2 user enables a specific dynamic tracepoint for a software module, the tracepoint is "patched" into the software module at the location specified within the definition of the dynamic tracepoint. Until a dynamic tracepoint is enabled, the software module does not incur additional overhead. However, a dynamic tracepoint is more expensive to execute than an equivalent static tracepoint.

The OS/2 2.0 Event Tracing facility is available to both the operating system and OS/2 applications. The OS/2 2.0 operating system uses both static and dynamic tracepoints. OS/2 application software modules can use static tracepoints only.

The OS/2 2.0 Event Tracing facility has three main components:

The Event Tracing Logic within the OS/2 Kernel: This component collects the event trace entries generated by various software modules. It holds the event trace entries in a circular buffer, and forwards them to the Event Tracing Formatter utility whenever the utility requests them.

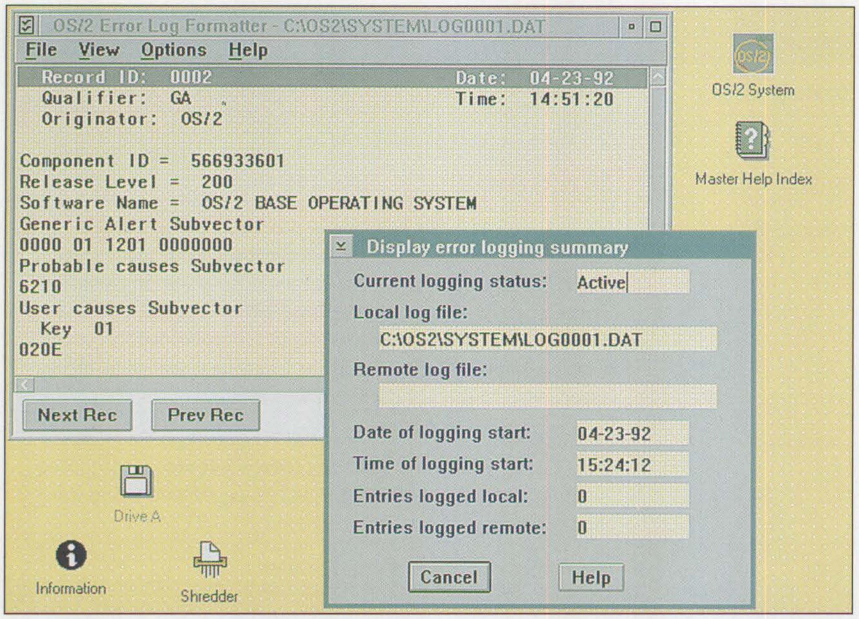
The Event Tracing Formatter utility (TRACEFMT.EXE): This is a Presentation Manager utility that formats event trace entries for display.

File	
Menu Item	**Purpose**
Open	To view another saved Error Log file.
Print	To print portions of the Error Log file.
Change Error Log file	To redirect an Error Log file to a remote system. Once system error logging has been redirected, all error entries are logged to the remote file rather than to the original local Error Log file. However, the local Error Log file serves as a backup error entry repository when communications errors make it impossible for the Error Logging daemon to log an error entry to the remote Error Log file.
Refresh	To make the Error Log Formatter reread the active Error Log file. It does not inform the user when new error entries are logged, but new error entries can be viewed using the Refresh menu.
Exit	To terminate the Error Log Formatter.
View	
Menu Item	**Purpose**
Display file header information	To display summary information about the Error Log file.
Display error logging summary	To view a summary of the current state of error logging. Figure 3 shows such a summary.
Option	
Menu Item	**Purpose**
Suspend error logging	To temporarily suspend the Error Logging facility. If a software module attempts to log an error entry while the Error Logging facility is suspended, the software will receive an error indicating that the Error Logging facility is currently suspended.
Display options	To select a subset of the Error Log records for viewing. The subset can be selected based on a date/time range, an Error Log record ID range, or both. Different classes of Error Log records have unique Error Log record IDs.

Figure 2. File, View, and Options Menu Items of the Error Log Formatter

Each time TRACEFMT.EXE retrieves a set of event trace entries from the kernel buffer, the entries are logically deleted from the kernel buffer. The utility also can copy event trace entries to a disk file. The user selects a TRACEFMT.EXE menu item to request that the currently displayed event trace entries be copied to the disk file. Unlike the OS/2 Error Logging facility, there is no automatic mechanism that continually logs event trace entries to disk files.

The Event Tracing Control utility (TRACE.EXE): This utility sends control messages to the Event Tracing Logic within the OS/2 Kernel. TRACE.EXE can enable or disable the tracing of individual tracepoints. It is also used to control both static and dynamic tracepoints.



```
  Record ID:   0002                          Date:   04-23-92
  Qualifier:   GA                            Time:   14:51:20
  Originator:  OS/2

Component ID =  566933601
Release Level =  200
Software Name =  OS/2 BASE OPERATING SYSTEM
Generic Alert Subvector
0000 01 1201 0000000
Probable causes Subvector
6210
User causes Subvector
  Key  01
020E
```

Display error logging summary

Current logging status: Active

Local log file: C:\OS2\SYSTEM\LOG0001.DAT

Remote log file:

Date of logging start: 04-23-92

Time of logging start: 15:24:12

Entries logged local: 0

Entries logged remote: 0

Figure 3. Summary of Current State of Error Logging

Enabling the Event Tracing Facility

Event tracing is enabled in much the same way as error logging. A user installs the three event tracing components, using the same steps as for the error logging components, and then modifies the CONFIG.SYS file. However, the CONFIG.SYS modifications for event tracing differ from the modifications to enable error logging. The user must include either a TRACE or a TRACEBUF statement in the CONFIG.SYS file.

The TRACE command tells OS/2 to enable the Event Tracing facility and to initialize it in either an active or suspended state. There are two acceptable forms for the TRACE statement:

```
TRACE=ON [major trace
   code(s)]
TRACE=OFF
```

A TRACEBUF command in the CONFIG.SYS file tells OS/2 to enable the Event Tracing facility and specifies the size of the circular event trace buffer in the OS/2 kernel. There is one acceptable form for the TRACEBUF statement:

```
TRACEBUF= [buffer size in
   kilobytes]
```

Users can include both TRACE and TRACEBUF statements within the CONFIG.SYS file. If neither statement is included, the Event Tracing facility is not enabled. If they include only a TRACE statement, the event trace buffer is created with a 4 KB size. If they include only a TRACEBUF statement, event tracing is enabled in a suspended state. The *OS/2 2.0 Online Command Reference* fully describes the TRACE and TRACEBUF statements.

Controlling Event Tracing

The TRACE.EXE utility controls which static and dynamic tracepoints are enabled at any given time. Users invoke this utility from the command line. It supports a rich command grammar that allows users to control individual tracepoints or groups of tracepoints. Users also can ask to trace events only if they occur on behalf of specified OS/2 processes. The *OS/2 2.0 Online Command Reference* details TRACE command syntax. Figure 4 shows four examples of this syntax.

Utility Command Line	Description
TRACE ON 15,31,22 /S	Indicates that all static tracepoints with major codes of 15, 31, and 22 are to be traced. The /S switch means that event tracing is to be suspended until a subsequent TRACE command with a /R switch resumes event tracing.
TRACE ON 17,22(4,9-13)	Says that the trace should include all static tracepoints with a major code of 17, and all static tracepoints with a major code of 22 and minor codes between 9 and 13.
TRACE ON KERNEL(FS=INT,SEM)	Inserts all dynamic tracepoints into the OS/2 kernel that are Group FS and Type INT, or are Group SEM.
TRACE ON /P:16,14,8	Indicates that events are to be traced only when the processes with Process ID 8, 14, or 16 are the current processes.

Figure 4. Trace Command Syntax

Viewing the Event Tracing Process

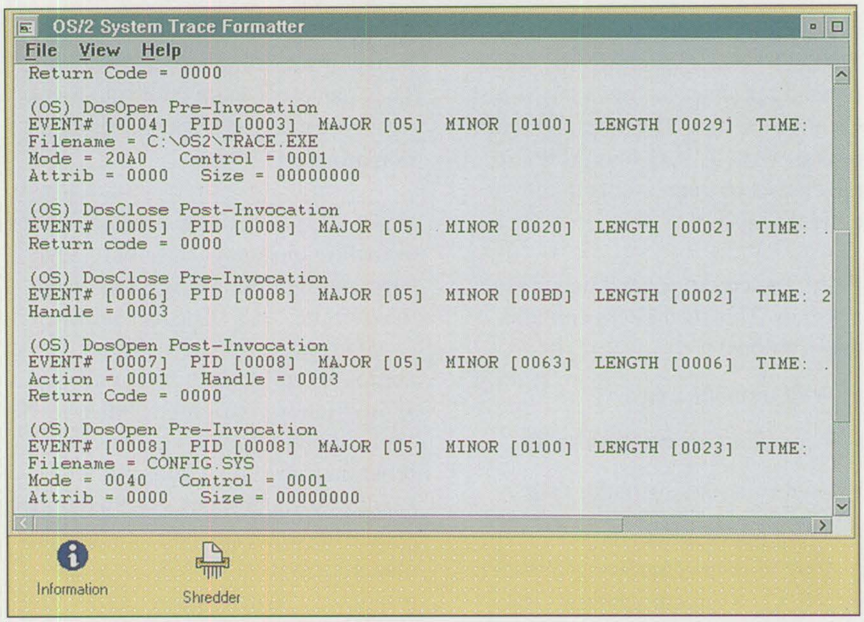
Users can invoke the Event Tracing Formatter (TRACEFMT.EXE) utility to format snapshots of the event trace buffer for display. When TRACEFMT.EXE is initiated, it acquires a snapshot of the event trace buffer. Subsequently, the user can choose a menu selection to acquire a new snapshot. The acquisition of a new event trace buffer snapshot replaces the existing snapshot (if any) within the Event Tracing Formatter, and also clears the contents of the event trace buffer within the OS/2 Kernel. This means that a particular trace entry can be acquired only one time from the OS/2 Kernel.

Figure 5 shows an Event Tracing Formatter screen. In this example, the kernel file system's dynamic tracepoints were enabled (through the use of the FS Group). The particular event was a result of executing the OS/2 command `TYPE CONFIG.SYS`.

The Event Tracing Formatter uses its client window to display formatted event trace entries. All the event trace entries acquired in a single snapshot are scrolled vertically within the client window in reverse chronological order. The top of the window contains the newest event trace entry. Typically, each formatted event trace entry has three sections:

- A descriptive title line that identifies the particular type of event trace entry
- A line that summarizes general event trace information, such as the major and minor trace codes associated with the trace entry, the Process ID (PID) of the process that generated the trace entry, and the time that the event trace entry was logged
- One or more lines that format the event-specific parameters that were logged within the trace entry

Figure 5 shows an event trace entry that corresponds to the `DosOpen` API pre-invocation tracepoint triggered by the opening of the `CONFIG.SYS` file. That event trace entry is identified by the descriptive title line "(OS) DosOpen Pre-Invocation." It is marked as event number 4 within the snapshot. The OS/2 command-line interpreter was executing as process 0x000C when the tracepoint was triggered. The major and minor tracepoint codes associated with the tracepoint are 0x0005 and 0x0100.



```
Return Code = 0000

(OS) DosOpen Pre-Invocation
EVENT# [0004]  PID [0003]  MAJOR [05]  MINOR [0100]  LENGTH [0029]  TIME:
Filename = C:\OS2\TRACE.EXE
Mode = 20A0   Control = 0001
Attrib = 0000   Size = 00000000

(OS) DosClose Post-Invocation
EVENT# [0005]  PID [0008]  MAJOR [05]  MINOR [0020]  LENGTH [0002]  TIME: .
Return code = 0000

(OS) DosClose Pre-Invocation
EVENT# [0006]  PID [0008]  MAJOR [05]  MINOR [00BD]  LENGTH [0002]  TIME: 2
Handle = 0003

(OS) DosOpen Post-Invocation
EVENT# [0007]  PID [0008]  MAJOR [05]  MINOR [0063]  LENGTH [0006]  TIME: .
Action = 0001   Handle = 0003
Return Code = 0000

(OS) DosOpen Pre-Invocation
EVENT# [0008]  PID [0008]  MAJOR [05]  MINOR [0100]  LENGTH [0023]  TIME:
Filename = CONFIG.SYS
Mode = 0040   Control = 0001
Attrib = 0000   Size = 00000000
```

Figure 5. The Event Tracing Formatter

The `DosOpen` API call that was logged contained the following parameters:

- The `FileName` requested to be opened was `CONFIG.SYS`.
- The `OpenMode` (Mode = 0x0040) indicates that the caller requested the file be opened for read-only access, and that neither Read nor Write was to be denied as a Sharing Mode.
- The `OpenFlag` (Control = 0x0001) indicates that the caller requested the file be opened, even if it already existed.
- The `FileAttribute` (Attrib = 0x0000) indicates a read-only file.
- The `FileSize` (Size = 0x00000000) is not significant because the `DosOpen` request was for read-only access.

The *OS/2 2.0 Control Program Programming Reference* (S10G-6263) documents the `DosOpen` API parameters.

The Event Trace Formatter provides a set of menu options that allows users to control the event trace formatting process and to save event trace buffer snapshots as disk files.

The File menu includes five menu items:

Open: To format a saved event trace snapshot file rather than request a new event trace snapshot from the OS/2 Kernel Trace logic.

Retrieve system trace buffer: To request a new event trace snapshot from the OS/2 Kernel Trace logic.

Save as...: To request that the current event trace snapshot be saved as a disk file. This menu item contains four options:

- Save unformatted trace data
- Save formatted trace data
- Save summary by Process ID
- Save summary by major code

The first option saves the event trace snapshot in a form that `TRACEFMT` can open later. The second option writes the ASCII formatting of the event trace snapshot into the TRACEFMT client window; this option typically produces a much larger disk file than the first. The third and fourth options write ASCII summaries of the classes of trace events in the current snapshot.

Print:To send formatted listings to a printer. This menu item contains three options:

- Print formatted data
- Print summary by Process ID
- Print summary by major code

The first option prints the ASCII formatting of the event trace snapshot in the TRACEFMT client window. The second and third options print ASCII summaries of the classes of trace events in the current snapshot.

Exit: To terminate TRACEFMT.

The View menu controls the format of the display within the TRACEFMT client window. It has three menu items:

View formatted data: To display formatted entries for each event trace entry within the snapshot. This is the standard view for the Event Trace Formatter. Figure 5 shows the TRACEFMT client window in this view.

View summary by Process ID: To display a summary for each PID of the types of events logged on behalf of the process within the snapshot.

View summary by major code: To display a summary for each major trace code of the events logged within the snapshot. For each major event trace code, the list of summarized events is subdivided by PID and then by minor trace code.

System Dumping

The OS/2 2.0 System Dump facility gives support personnel a mechanism for capturing the entire main memory image of a running OS/2 2.0 system. They can use system dumps to investigate operating system and application program failures.

OS/2 2.0 system dumps can be either asynchronous or synchronous. Asynchronous dumps are triggered when the user presses Ctrl+Alt+NumLock twice in succession. Synchronous dumps are triggered when OS/2 is appropriately configured and an application program experiences a program trap that it chooses not to handle. Asynchronous system dumps are useful when an application or the operating system "hangs" or is unresponsive. Synchronous dumps are a valuable way to capture the state of the computer at the point that an application program traps.

The System Dump facility gives support personnel a mechanism for capturing the entire main memory image of a running OS/2 2.0 system.

OS/2 2.0 system dumps are written to a series of floppy diskettes. Dump diskettes are formatted to normal diskette capacities of 1.4 MB. The system dump process tries to compact the main memory image as it is written to diskette. However, the pattern of values in memory at the time the dump is taken may prevent significant compaction, since compaction is most effective when it encounters repetitive values. A prudent user will calculate the number of diskettes necessary by dividing the size of main memory by the storage capacity of each diskette. If an OS/2 system has 8 MB of memory and the storage capacity of each dump diskette is 1.4 MB, the user will need a maximum of six dump diskettes to dump all the main memory.

Preparing for a System Dump

When a system dump is triggered, a message appears on the user's video display requesting the user to insert a system dump diskette into drive A:. This system dump diskette is a specially formatted diskette created with the CREATEDD utility. CREATEDD operates by formatting the diskette, then installing both bootstrap and stand-alone system dump software on the diskette. After the first dump diskette, the other diskettes used in an OS/2 system dump do not require special formatting. They must be formatted by the FORMAT utility.

For an asynchronous system dump, no further special preparations are necessary. For a synchronous system dump, the user must add the TRAPDUMP=ON statement to the CONFIG.SYS file.

The TRAPDUMP statement tells OS/2 2.0 to trigger a system dump if a Ring 3 trap occurs in a program and the program fails to handle the trap. *Rings* are partitions in a computer's total address space. In an OS/2 2.0 system, Rings 0–2 of the Intel 80386/80486 privilege model are reserved for use by the operating system and by privileged OS/2 software modules. Ring 3 is non-privileged and available to OS/2 application programs. The OS/2 2.0 Exception Management facility allows OS/2 programs to create exception handler routines that can gain control when an OS/2 thread causes a program trap. When the exception handler routine completes its logic, it tells the OS/2 2.0 kernel whether it handled the trap (that is, whether it corrected the conditions that caused the trap).

An OS/2 2.0 thread can define multiple exception handlers. If a trap occurs in an application program and none of its exception handlers choose to handle the trap, then OS/2 terminates the program. If the TRAPDUMP=ON statement is included within CONFIG.SYS, then OS/2 2.0 will immediately trigger a system dump.

The TRAPDUMP=ON statement should be used judiciously. It should be placed in the CONFIG.SYS file only while a particular program trap is under investigation. If the TRAPDUMP statement is left in the CONFIG.SYS when it is not necessary, it will trigger a system dump whenever any non-handled Ring 3 trap occurs. The triggering of a system dump terminates OS/2 operation. Once a system dump has been triggered, the user's only choices are to take the system dump (by feeding in the series of diskettes) or to reboot the system.

The System Dump Process

Once an OS/2 2.0 system dump has been triggered, the system dump process guides the user through its operation. First, the user receives a prompt to load the system dump diskette (created with the CREATEDD utility) into drive A: and press the Enter key. The system then loads the stand-alone dump program from the system dump diskette and copies the contents of the computer's main memory to the system dump diskette. When the system dump diskette is completely loaded, the dump program tells the user to remove the current diskette and load a new formatted diskette. This process continues until all the main memory is copied onto diskette. When the last diskette is written, the dump program tells the user to remove the final diskette and load the system dump diskette once more. The dump program needs to write summary information to the system dump diskette after writing all the main memory.

If a system dump diskette is used for an OS/2 2.0 system dump, it must be reformatted by CREATEDD before it can be reused for another system dump. This is necessary because the system dump operation overwrites the contents of the system dump diskette.

Formatting a System Dump

Once an OS/2 system dump has been taken, the diskettes that compose the system dump should be given to qualified OS/2 support personnel, who can analyze the system dump using the OS/2 2.0 System Dump Formatter utility. This utility is not included with OS/2 2.0; it is for experienced OS/2 support personnel only. The System Dump Formatter analyzes an OS/2 2.0 system dump and reconstructs the major OS/2 internal control blocks. It produces a summary of the process and thread environment at the time of the system dump. It also summarizes the memory object environment at the time of the dump.

OS/2 2.0 system dumps are critical serviceability tools because support personnel can use them to perform first failure data capture on problems that they cannot otherwise reproduce. They can also use system dumps to investigate operating system and application failures. For instance, because the program stacks for each application thread are included within a system dump, support personnel can logically unwind the stacks to determine the location of the thread at the time of the dump and the series of program calls (and their associated parameters) that led to the final thread location.

Analysis of a system dump is more difficult than the high-level language debugging of an application program. The typical application program captured in a system dump does not include the extra symbolic information needed for high-level language debugging. The analysis of a system dump takes place at the assembler language level.

Event Tracing within a System Dump

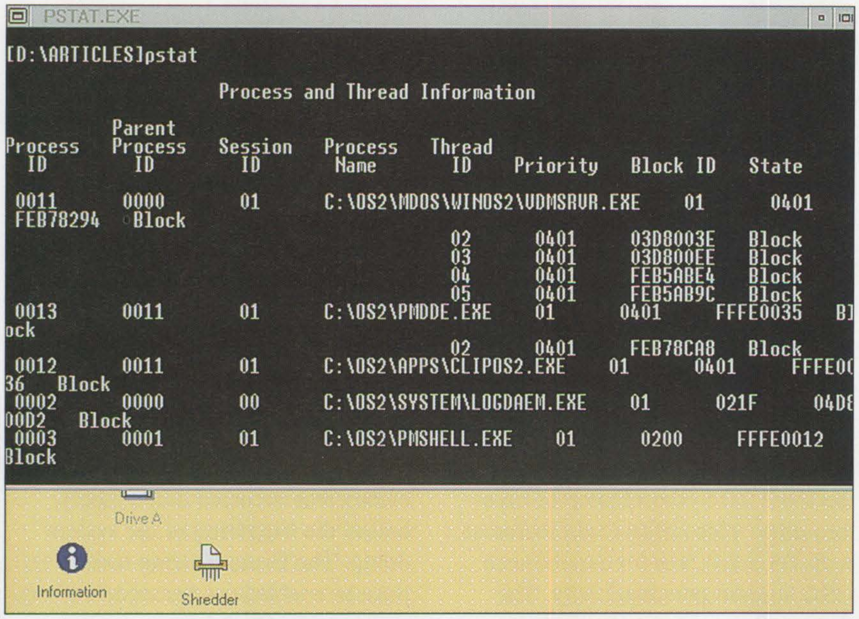
The OS/2 2.0 Event Tracing facility is a valuable tool that support personnel can use with the OS/2 2.0 System Dump facility. Since the OS/2 2.0 event trace buffer is maintained in main memory, it will be included in an OS/2 system dump.

The entries in the event trace buffer can provide a limited history of events that occurred immediately before the triggering of the system dump. The Event Tracing facility must be enabled (as described earlier), and specific OS/2 2.0 tracepoints must also be enabled. The choice of tracepoints to be enabled depends on the type of failure being analyzed. For example, if an application hang is being investigated, and there is reason to believe that the hang is due to a semaphore deadlock, then the various OS/2 2.0 semaphore tracepoints could be enabled. The system would then be run until the application hang occurs, at which point an asynchronous system dump could be taken.

The OS/2 System Dump Formatter utility allows the event trace buffer in an OS/2 2.0 system dump to be formatted in the same way that the OS/2 2.0 Event Tracing Formatter utility formats the event trace buffer on a running system.

Resource Analysis

The PSTAT text-mode utility tracks the system resources used on an OS/2 system and generates an OS/2 2.0 system-resource analysis report. Such data is important background information during a problem investigation, because it describes the general environment within which the problem occurred. PSTAT is generally used as a secondary serviceability



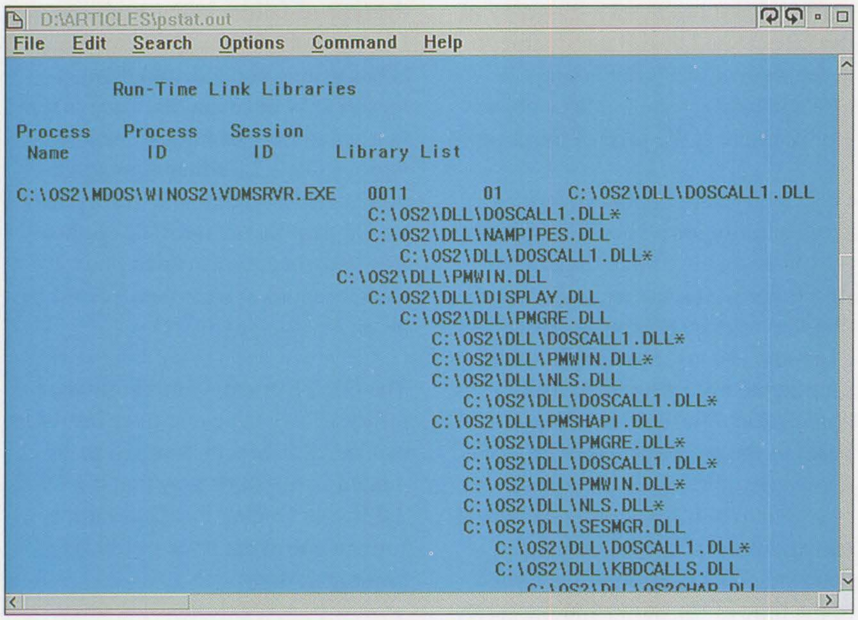
```
[D:\ARTICLES]pstat

                        Process and Thread Information

              Parent
Process       Process       Session      Process      Thread
  ID            ID            ID          Name          ID       Priority     Block ID      State

 0011          0000           01         C:\OS2\MDOS\WINOS2\VDMSRVR.EXE     01          0401
 FEB78294      Block
                                                       02         0401        03D8003E      Block
                                                       03         0401        03D800EE      Block
                                                       04         0401        FEB5ABE4      Block
                                                       05         0401        FEB5AB9C      Block
 0013          0011           01         C:\OS2\PMDDE.EXE    01          0401         FFFE0035     Bl
ock
                                                       02         0401        FEB78CA8      Block
 0012          0011           01         C:\OS2\APPS\CLIPOS2.EXE     01          0401          FFFE0
36   Block
 0002          0000           00         C:\OS2\SYSTEM\LOGDAEM.EXE      01          021F         04D
0D2    Block
 0003          0001           01         C:\OS2\PMSHELL.EXE      01          0200          FFFE0012
Block
```

Figure 6. Process and Thread Information in a PSTAT Report



```
          Run-Time Link Libraries

Process    Process    Session
 Name        ID         ID        Library List

C:\OS2\MDOS\WINOS2\VDMSRVR.EXE    0011       01       C:\OS2\DLL\DOSCALL1.DLL
                                     C:\OS2\DLL\DOSCALL1.DLL*
                                     C:\OS2\DLL\NAMPIPES.DLL
                                        C:\OS2\DLL\DOSCALL1.DLL*
                                  C:\OS2\DLL\PMWIN.DLL
                                     C:\OS2\DLL\DISPLAY.DLL
                                        C:\OS2\DLL\PMGRE.DLL
                                           C:\OS2\DLL\DOSCALL1.DLL*
                                           C:\OS2\DLL\PMWIN.DLL*
                                           C:\OS2\DLL\NLS.DLL
                                              C:\OS2\DLL\DOSCALL1.DLL*
                                           C:\OS2\DLL\PMSHAPI.DLL
                                              C:\OS2\DLL\PMGRE.DLL*
                                              C:\OS2\DLL\DOSCALL1.DLL*
                                              C:\OS2\DLL\PMWIN.DLL*
                                              C:\OS2\DLL\NLS.DLL*
                                              C:\OS2\DLL\SESMGR.DLL
                                                 C:\OS2\DLL\DOSCALL1.DLL*
                                                 C:\OS2\DLL\KBDCALLS.DLL
                                                    C:\OS2\DLL\OS2CHAR.DLL
```

Figure 7. DLL Module Information in a PSTAT Report

tool and can track five different types of OS/2 system resources:

- Processes
- Threads
- System semaphores
- Named shared memory objects
- DLL modules

PSTAT writes a report to the standard output file. This report summarizes some or all of the five types of system resources and can give valuable information about which processes are holding critical system resources. This information helps in analyzing application failures and performance problems. Details about the different PSTAT command-line options are in the *OS/2 2.0 Online Command Reference.*

Figure 6 shows an example of the process and thread information contained in a PSTAT report. In the report, processes are identified by complete path name, session, and PID, and threads within a process are identified by Thread ID (TID). The state and priority of each thread are displayed. If a thread is blocked, waiting for a system resource, the internal system ID of the resource also is listed.

The PIDs returned in this section of the PSTAT report can be used with the OS/2 2.0 TRACE utility to enable tracepoints to specific processes only.

Figure 7 shows an example of the DLL module information in a PSTAT report. This section of the report shows which DLL modules are referenced by the different active processes on the system. It also shows the DLL modules referenced, in turn, by other DLL modules.

__Allen M. Gilbert__ is a senior programmer in OS/2 development in the IBM Personal Systems Programming Center in Boca Raton, Florida. He joined IBM in 1990 and has worked as an OS/2 architect and designer. Currently he is responsible for the design of OS/2 system management facilities and is involved in the development of advanced graphic display capabilities. Allen has a BS in biology from the State University of New York at Stony Brook and an MS in genetics from the University of California at Davis.

Online Communication Using OS/2 2.0 PM Terminal

Jean N. Shortley
IBM Corporation
Boca Raton, Florida

PM Terminal is an asynchronous terminal emulator that is quick and handy to use because of the Presentation Manager's graphical user interface. It enables quick and easy access to the IBM Information Network and other networks' online information services. PM Terminal (Also known as Softerm Custom V1.00) offers a subset of features found in Softerm® Modular, available from Softronics, Inc.

OS/2 2.0 supports a large number of today's DOS applications. Communications users who access the IBM Information Network (IIN) or other networks or bulletin boards can move up to the world of Presentation Manager graphics with PM Terminal. PM Terminal is located in the Productivity folder in the OS/2 System icon under the Workplace Shell. It can be used to dial into online information services, emulate a terminal, and upload or download files. Its features include support for nine terminal emulators, ten file transfer protocols, keystroke recording for easy macro creation, keyboard remapping, screen-to-disk or printer captures, and online help. Figure 1 lists the terminal emulators and file transfer protocols supported under PM Terminal.

The Session Phonebook

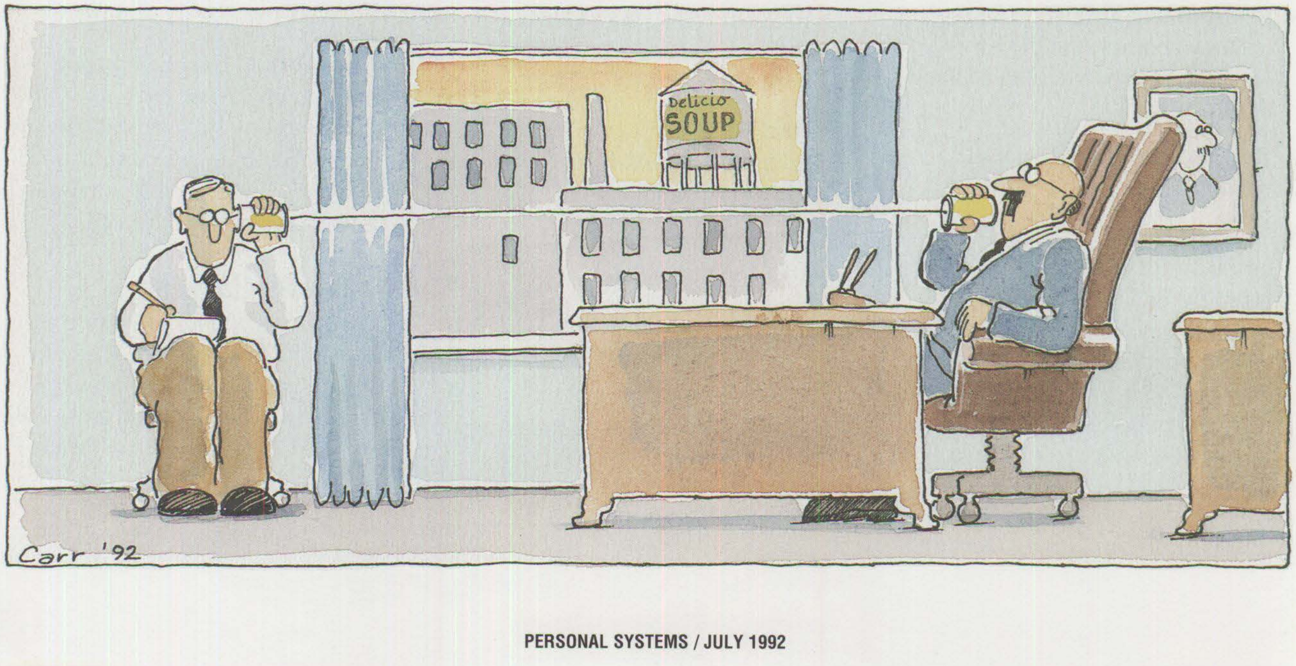
A *session profile* in PM Terminal is a bulletin board entry in the Session Phonebook. Through the Add and Change options in the Session Menu, session profiles for each session's communications, terminal emulation, file transfer, and environment settings can be set up. If new profiles need to be created, select the Setup Profiles option. PM Terminal saves the profile changes in the configuration database `CUSTOM.MDB`. PM Terminal automatically keeps a backup of the configuration database, `CUSTOM.BAK`. PM Terminal supports only one configuration database.

The Change Session Window

The following is an example of how to configure PM Terminal to access IIN. Before modifying PM Terminal, you should already be a registered user of IIN, meaning you have an account ID, a user ID, and a password. PM Terminal is already configured with an 800 number to access IIN, and it also lists the voice number at (800) 727-2222 for speaking with a technical support person

The first step is to review the default profiles, because some changes may be needed depending on your system's configuration. Initially, the IIN session is set up to communicate over a Hayes® autodial 1200-baud modem connected through COM1 using an 84-key AT-style keyboard and a customized VT100 terminal emulator. This can be changed using the following steps:

1. From the Session Phonebook, highlight the IBM Information Network entry.
2. From the Session option of the menu bar in PM Terminal, select the Change option to modify any



Terminal Emulators	File Transfer Protocols
ANSI 3.64	Character
DEC VT100	KERMIT
DEC VT52	KERMIT Server
GEnie® terminal	SOFTRANS
IBM 3101-10	XMODEM
IBM 3101-20	XMODEM 1K
IBM ANSI	YMODEM
IBM INVT100	YMODEM 1K
TTY	YMODEM-G, YMODEM-G 1K

Figure 1. Emulation and Protocols supported by PM Terminal

of the settings related to this session entry.

3. The Change Session window appears. Select the Setup Profiles pushbutton.

4. From the Setup Profiles dialog, select the Terminal pushbutton.

5. Highlight the IBM VT100 emulator (INVT100) from the Terminal Emulation Profile module, and select Change. *Note:* Selecting an emulator other than the IBM INVT100 emulator may require a lot of keyboard remapping (knowledge of an emulator's functions for each key).

6. The Terminal Emulation Settings window appears, and the Keyboard profile name dialog is blank. Select Setup.

7. The Keyboard Profile module window appears. Select Add.

8. The Add Keyboard window appears. Select which keyboard you have (the default 84-key AT-style or the enhanced 101- or 102-key styles), which terminal keyboard type you want, and the nationality if other than USA. Select OK after you have made your selections.

9. The Keyboard Settings window appears. To view the functions of the keys on the IBM INVT100 keyboard, select Change next to Keyboard Layout. To find a key's function, highlight the key. To change the key's function, select the desired "Open" key pushbutton. From the Open/Edit key window, it is easy to define a key's function by selecting the function or the character from the dialogs. Select Cancel to leave the layout without saving any changes. Figure 2 shows the Keyboard Remap window.

10. Select Save As to name and store your keyboard profile.

11. Your new keyboard profile name will appear in the dialog. Select Close to return to the Terminal Emulation Settings window. Other terminal emulation settings, such as printer definition, tab stops, and duplex, can also be changed here.

12. Select Save and Close to return to the Change Session window.

The default communication parameters for IIN are COM1, 1200 baud, even parity, 7 data bits, and 1 stop bit. You may want to change the baud rate to match that of your modem. The other parameters must remain the same. To change the baud rate, you need to modify the Modem and the Connection Path profiles. To do that:

1. Select Setup Profiles from the Change Session window or from the Session Menu option.

2. Select the Connection pushbutton in the Setup Profiles window.

3. The IBM IIN Connection Path profile is highlighted. Select Change.

4. From the Connection Path Settings window, select your COM port. Select Change next to Communication Parameters, then choose your specific baud rate.

5. Once the baud rate is set, and while in the Connection Path Settings window, select Setup next to Modem Profile to add your own modem profile or to change the Hayes profile. If your modem is not listed, but it is compatible with Hayes, you can use the Hayes entry and select Change.

6. Set the device initialization string for your modem. If you have difficulty dialing out, this is one place you may need to customize. Your modem's documentation can provide you with the required commands to put your modem into Hayes emulation mode. Other modem-specific features are the Call Failure Type and Response string, Hang Up string, Dialing string, Auto-answer strings, and modem timeout (the amount of time, between 1 and 255 seconds, to wait for the connection to complete).

7. Select Save, then Close, to store your modem profile settings. Select Save and Close again to return to the Change Session window.

8. Select OK. The Admittance data dialog appears, displaying the IIN 800 number. Within this dialog are options to add any handshake transmission data or telephone network setups. Also, the options to display the admittance data dialog at connect time, or to display the autodial dialog (the modem timeout) can be selected as well.

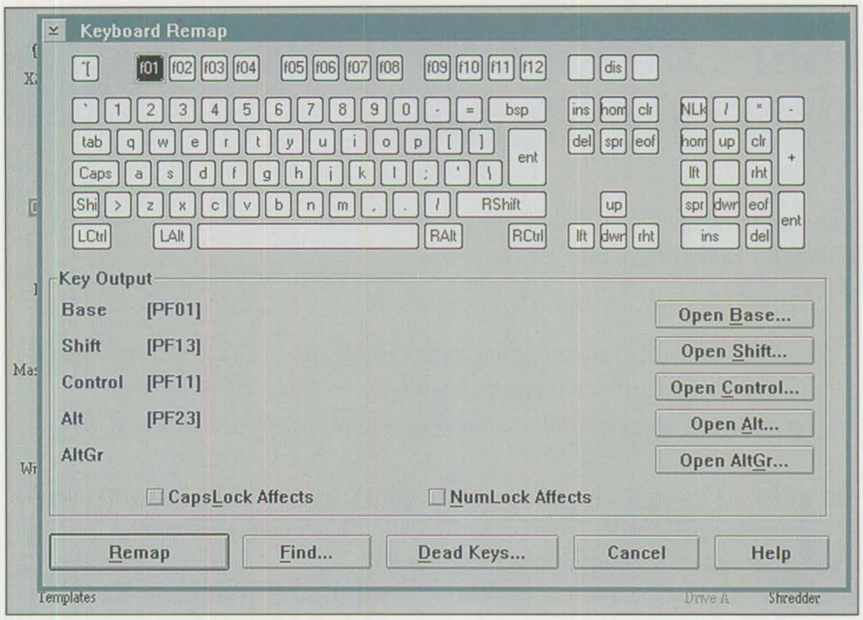
The Session Window

Once your session is configured and selected, the PM Terminal Communications Session window appears. Your modem initialization string will be displayed, and success or failure to connect to your desired online service or bulletin board is also displayed. If you are connected to IIN, enter a terminal type of 17 for VT100 emulation. The File menu contains options for the Send and Receive file transfer functions, dial for reconnection, disconnect, and data capture of on-screen text (in either binary or text format) to the disk or to the printer.

File transfers with IIN are not supported under this version of PM Terminal because of the lack of the required IND$FILE file transfer protocol. The IND$FILE transfer protocol is available in a communications package from Softronics, Inc. called Softerm Modular. For more information, contact Softronics at (800) 225-8590. Figure 3 shows a PM Terminal file transfer from the IBM National Support Center bulletin board using the XMODEM protocol.

The Edit menu displays the current keyboard to help locate keys for particular functions. From the Begin Keyboard Record option, keystrokes can be recorded with a selected ID for later playback. This feature is handy for frequently performed series of keystrokes or for setting up a logon sequence.

The Settings menu is used to modify the Session Window and its settings. Here is where on-the-fly keyboard remapping changes can be made. If permanent changes need to be made to the screen colors or to the keyboard for a session, return to the Session Phonebook and select Setup Profiles from the Session Menu option. From the Setup Profiles window, select the Environment pushbutton.

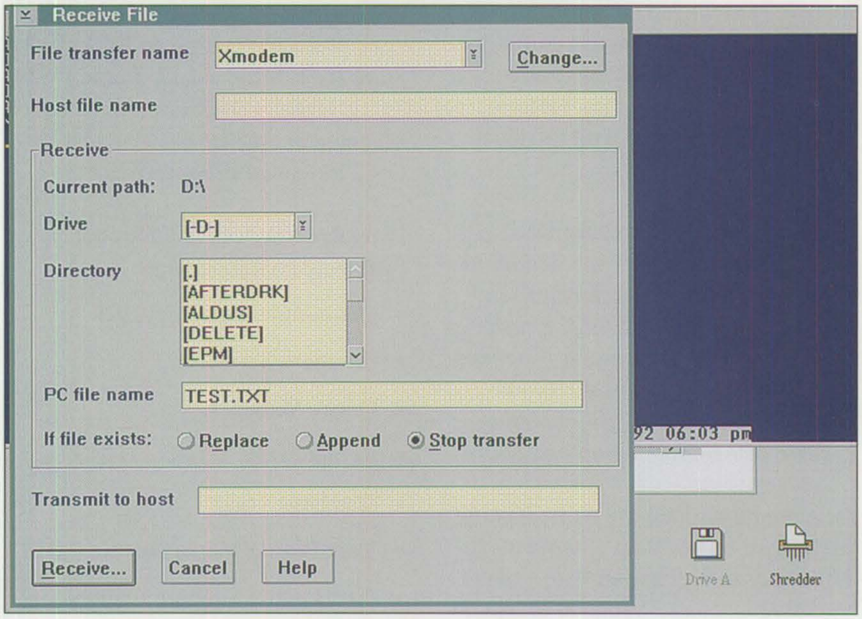

Keyboard Remap
f01 f02 f03 f04 f05 f06 f07 f08 f09 f10 f11 f12 dis
` 1 2 3 4 5 6 7 8 9 0 - = bsp ins hom clr NLk / * -
tab q w e r t y u i o p [ ] ent del spr eof hom up clr +
Caps a s d f g h j k l ; ' \ lft rht
Shi > z x c v b n m , . / RShift up spr dwr eof ent
LCtrl LAlt RAlt RCtrl lft dwr rht ins del
Open Base...
Open Shift...
Open Control...
Open Alt...
Open AltGr...
Remap
Find...
Dead Keys...
Cancel
Help
Templates
Drive A
Shredder

Key Output

Base	[PF01]
Shift	[PF13]
Control	[PF11]
Alt	[PF23]
AltGr	

□ CapsLock Affects □ NumLock Affects

Figure 2. Keyboard Remap Window



File transfer name: Xmodem

Host file name:

Receive

Current path: D:\

Drive: [-D-]

Directory:
[..]
[AFTERDRK]
[ALDUS]
[DELETE]
[EPM]

PC file name: TEST.TXT

If file exists: ○ Replace ○ Append ● Stop transfer

Transmit to host:

Figure 3. Example of PM Terminal File Transfer

***Jean N. Shortley** is a senior associate programmer in IBM's Entry Systems Technology Laboratory in Boca Raton, Florida. She was formerly a member of the printer device-driver development and test teams and the Multiple Virtual DOS Machines (MVDM) test team. Jean holds a BS in education and an MEd in educational microcomputer research, both from Florida Atlantic University in Boca Raton.*

IBM Extended Services Database Manager

Dena M. Laterza
Computer Task Group, Inc.
Dallas, Texas

This article describes the enhancements in the new OS/2 Extended Services Database Manager including backup, restore, roll-forward recovery, date/time arithmetic, and user-defined collating sequences.

Users are moving some data processing applications from mainframes to OS/2. For many, the reason is efficiency. Finding the most effective platform on which to run an application is called *right-sizing* the application. Right-sizing usually involves moving some applications from mainframes to microcomputers. Implementing right-sizing is advantageous in data processing environments because working on the local level is easier for users.

The main advantages of the OS/2 platform are its ease of use and multitasking capability, which results in high productivity. The OS/2 Database Manager engine has similar functions to IBM's family of host relational database systems and offers comparable data manipulation operations.

Because more database work is being performed on personal computers, IBM has received requests to expand the OS/2 Database Manager to handle more complex applications. Users have asked for improved database backup and restore to protect against disk failure. Users also want the data manipulation capabilities of mainframe systems to be available on less expensive personal computers. IBM has integrated some of these functions into Database Manager with several significant enhancements.

Extended Services is Here!

On October 22, 1991, IBM announced Extended Services Version 1.0. Extended Services includes an enhanced Database Manager and Communications Manager; it runs on IBM OS/2 Standard Edition (SE) 1.3[1] and OS/2 2.0. It replaces the Communications Manager and Database Manager in OS/2 Extended Edition (EE) 1.3.

Extended Services includes the following components:

- Communications Manager
- Database Services
- Query Manager
- Remote Data Services (RDS)
- Database Tools
- User Profile Management
- First Failure Support Technology/2 (FFST/2)
- Command Line Interface (CLI)

Extended Services with Database Server for OS/2 contains the above components plus database server capability. Figure 1 shows the Extended Services environment. Remote Data Services does not require LAN server/requester code on the database server or the client workstations. Extended Services clients cannot access OS/2 EE Database Manager servers. Extended Services database servers can accept requests from the following:

- Extended Services 1.0 clients using OS/2 NetBIOS, Advanced Program-to-Program Communications (APPC), or Advanced Peer-to-Peer Networking (APPN)
- OS/2 EE 1.2 and 1.3 clients using either SQLLOO or APPC
- DOS, Microsoft Windows 3.0, and OS/2 clients using NetBIOS

To benefit from the enhancements to Database Manager, Extended Services with Database Server should be installed on your database servers, and client workstations upgraded as needed. To assist in this process, an optional Administrator's Kit contains helpful publications for system administrators and database administrators.

IBM Extended Services with Database Server for OS/2 provides a distributed feature called the Database Manager client application enabler. The application enabler allows OS/2, DOS, and Microsoft Windows workstations to access Extended Services database servers using NetBIOS. Licenses for each workstation are purchased separately.

New Database Manager Features

Database Manager's enhancements discussed in this article affect the user, database administrator, and application programmer. Database Manager's ability to access data on different platforms is also introduced. This distributed database capability may change the way we think of the client/server environment.

[1] IBM recommends that Corrective Service Diskette (CSD) 5050 or higher be applied to the OS/2 SE or OS/2 EE 1.3X before running Extended Services or accessing an Extended Services server from a 1.3 client. Also, CSD 4098 is recommended for OS/2 SE 1.2 or OS/2 EE clients.

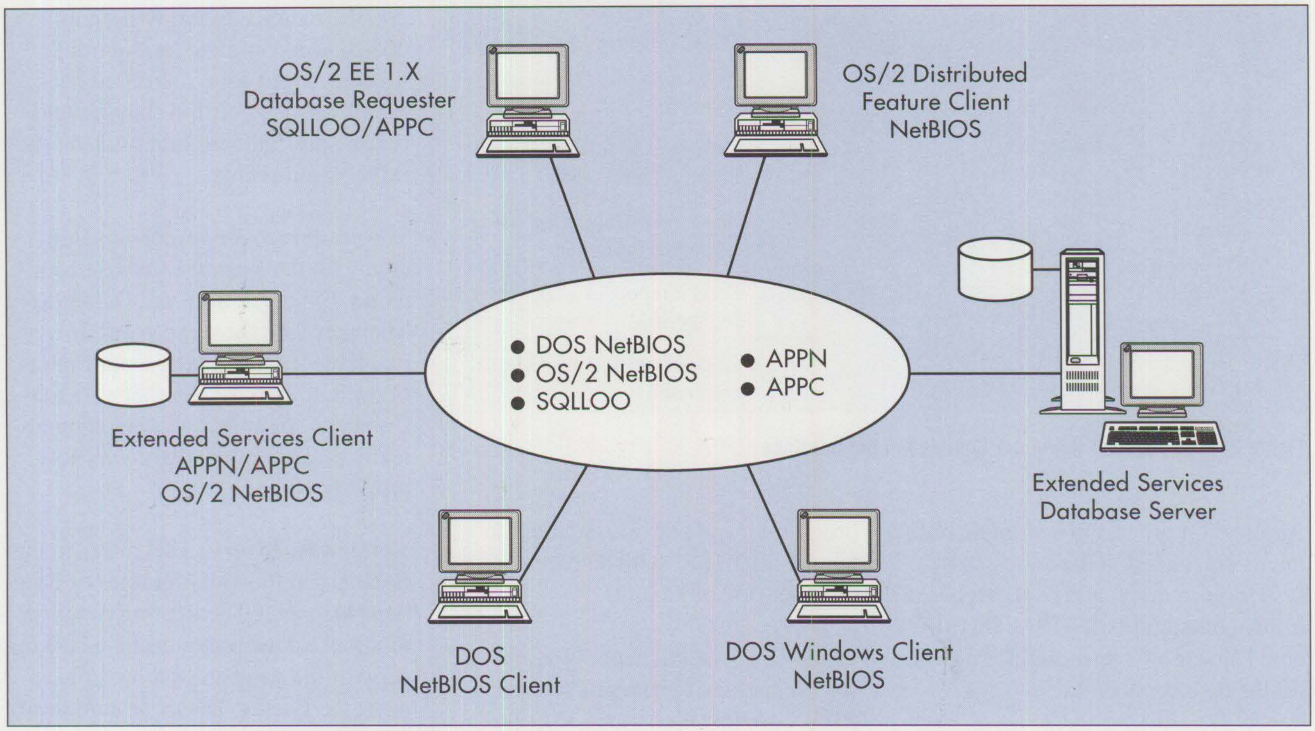


Figure 1. Extended Services Client/Server Environment

Database Tools

Extended Services provides three Presentation Manager (PM) applications for administrative functions and configuration. These applications, called Database Tools, are an installable option. Some functions were previously provided by the OS/2 EE 1.3 Query Manager and some new functions have been added. Database Tools include the following:

Directory Tool: The Directory Tool allows users to manage access to local and remote databases. There are four Database Manager directories: Database Connection Services, System, Volume, and Workstation. The operator can perform administrative functions such as Create, Drop, Catalog, and Uncatalog directory entries, and view information about all cataloged databases including host, remote, and local databases.

Configuration Tool: With the Configuration Tool, users can change database configuration parameter values and Database Manager configuration. Tuning these parameters can result in optimal database performance. The Configuration Tool conveniently indicates which databases are accessible from your workstation, displaying each database by alias, name, drive, and workstation.

Recovery Tool: The Recovery Tool enables system administrators (users with SYSADM authority) to back up, recover, or restart a database. A window lists the cataloged databases by alias, name, drive, and workstation.

Command Line Interface

CLI is an extremely useful application that processes Structured Query Language (SQL) statements and Database Manager commands entered from the OS/2 command prompt or a command file, as shown in Figure 2. The SQL statements are, by default, automatically committed, with messages of completion and output displayed on the screen. Options are available to suppress all commits and to redirect output to a file. CLI also maintains a history file of the requests it receives in the `\SQLLIB\DBM.RUN` file.

A new function that is accessible only through the CLI is `REORGCHK`. It runs statistics on a database to determine if performance can be improved by reorganizing a table in the database.

Online help is provided through the Extended Services Command Reference, which describes the syntax of SQL statements and all the Database Manager commands that can be used with the CLI.

Figure 2 shows examples of using the Command Line Interface.

Programming Language Support

Application programs can access all Database Services functions. Programming languages supported are REXX, IBM C Set/2 (32-bit compiler), Microsoft C6, MicroFocus® COBOL, and IBM FORTRAN.

Command Line	Description
`DBM START USING DATABASE SAMPLE`	Connect to the SAMPLE database
`DBM SELECT * FROM STAFF`	List the contents of all rows in the STAFF table
`DBM -R(c:\staff.out) SELECT * FROM STAFF`	Send a report to a file using `-R` report option
`DBM "SELECT * FROM STAFF" > c:\staff.out`	Send a report to a file using redirection
`DBM ? INSERT`	Obtain help for the `INSERT` command

Figure 2. Examples of Using the Command Line Interface

Application programs with SQL statements embedded in the code can access local, remote, and host databases if the workstation uses IBM Distributed Database Connection Services/2 for the host connection.

Backup Support for Non-Standard Devices

Previous versions of Database Manager supported backup to diskette or a dedicated hard disk drive partition. One of IBM's main objectives in enhancing database backup was to support more devices, such as LAN and tape drives. There are two types of backup: full backup and changes-only. Changes-only backs up only the files that changed since the last backup was made. The Database Manager's new Backup Database utility uses either the operating system backup facility or a user-supplied program called a *user exit program.* A user exit program provides the bridge between the Database Manager and storage devices that are not supported by the OS/2 backup command. User exit programs can also be written to direct logs or database backups to larger off-site host systems.

Database Manager calls a user exit program using the functions Backup, Restore, Archive, and Retrieve. Backup and Restore invoke a user exit program when "0" (zero) is specified as the drive parameter. The user exit program must be an executable file named `SQLUEXIT` with an extension of either `CMD` or `EXE`.

When Database Manager is installed, sample user exit programs written in REXX are installed in the `\SQLLIB` subdirectory. The sample programs show how to set up user exit programs; they are provided "as is," without warranty. Users can modify or otherwise use these programs in any way they wish.

Recovery Features

Routine backups are important to protect against losing data, but it is always possible to lose work done since the last backup. Recovering even one day's work is often costly. There are three types of recovery: crash, version, and roll-forward recovery. OS/2 EE 1.3 Database Manager provides crash recovery and version recovery. Extended Services adds roll-forward recovery.

Crash Recovery: Database Manager's crash recovery process protects data against the most likely problems: failing media, failing hardware, power loss, and application failure. Crash recovery uses a circular logging technique to keep track of database transactions. The logs may contain committed transactions that have not yet been written to disk as part of the database. In case of failure, the data pages are written to disk, and incomplete transactions (uncommitted data) are rolled back. A rollback cancels the changes and returns the database to its state at the last commit point.

The crash recovery process is initiated with the Restart Database command. Extended Services Database Manager uses the `auto_restart` parameter in the database configuration file to specify whether a Restart Database command will be automatically issued if a database was not properly closed.

Version Recovery: This refers to the Backup and Restore functions of Database Manager. The Restore function rebuilds a database by using a backup copy of the database that was made using the Backup Database command. The database is returned to the state it was in at the time of the backup.

Roll-forward Recovery: Roll-forward recovery eliminates the expense of reconstructing the database and protects against loss of data in case of disk failure. Transactions since the last backup are reapplied against a restored database. Roll-forward recovery is a combination of crash and version recovery that uses an archive logging technique – when active logs (or uncommitted transaction logs) are no longer needed for normal processing, they are closed and become archived logs. Archived logs ensure data protection in mission-critical database applications. Both active and archived log files can reapply the changes made to the database. The roll-forward recovery method allows a database to be rebuilt to a particular point in time or to the end of the database logs.

Once an active log file is archived, it is stored online in the database log-path directory by default. However, archiving log files to another location ensures additional protection of data and frees up space on hard drives. Archived log files can be moved offline manually (or via a user exit program) to another location, such as a tape or network drive. Enabling either the database `user_exit` parameter or the `log_retain` parameter will begin archiving the log files. If `user_exit` is enabled, Database Manager automatically calls `SQLUEXIT` to move the files offline. During roll-forward recovery, the database configuration file `user_exit` parameter also specifies whether Database Manager invokes a user exit program to retrieve required log files that are not found in the log path directory.

A full understanding of the roll-forward recovery process is necessary to decide whether it would be beneficial to a particular database environment. Maximum data protection may not be needed in all cases, and configuring every database for roll-forward recovery may not be necessary. Because roll-forward recovery is not enabled by the default settings, changes to the database configuration file must be made before archive logging takes place. Once it has been decided which databases need to have roll-forward recovery, enabling it is simple.

Using Roll-Forward Recovery: To enable a database for roll-forward recovery, use the Configuration Tool or the Update Database Configuration File routine to change the `log_retain` (or `user_exit`) parameter. Consider tuning the following related database configuration parameters: number of primary logs, number of secondary logs, log size, and log path. After roll-forward recovery is enabled, the database is inaccessible and must be entirely backed up. This is because roll-forward recovery builds upon a restored database.

There are two phases in roll-forward recovery: restore and roll-forward. The Restore Database routine or the Recovery Tool can be used to complete the first phase. After a successful restore, a database configured for roll-forward recovery enters a "roll-forward pending" state. The database is not usable while in this state.

The next step is to roll-forward the changes made to the database since the backup. To do this, use either the Roll Forward Database routine or the Recovery Tool. Roll-forward can be selected from the Recover menu to continue automatically with the roll-forward process after the restore is complete. If the Roll-Forward menu option was not selected when the database was restored (so that the database is flagged "roll-forward pending"), select the Resume button to begin the roll-forward process. The roll-forward process will restore the database changes to the end of the log files, or a stop time can be specified.

Note: If the database is rolled forward to the end of the log files, none of the transactions can be rolled back. To omit some of the last transactions made against the database, you must specify a stop time.

After restoring a database, suppose you decide *not* to roll-forward the changes made to the database. Because roll-forward is still pending, you must stop the roll-forward process to make the database usable. Using the Restore Database routine, set the Stop parameter so that the database will no longer be marked as needing a roll-forward operation. To do the same using the Recovery Tool, simply restore the database, do not select the Roll-Forward option on the menu, then click on the Stop button after restore is complete.

Note: If a database is backed up and restored with a user exit program (selecting drive 0 as the target/source drive), restoring the database does not require users to disconnect from all databases on the workstation, only the one being restored.

Greater SQL Standards Compliance

SQL enhancements include the following functions:

Date/Time Arithmetic and Scalar Functions. New SQL operations include addition and subtraction in `DATE`, `TIME`, and `TIMESTAMP` fields, and the capability to extract dates and times from timestamps into strings, as shown in Figure 3.

Translate. This function converts characters to their uppercase equivalents. One use of translate is for case-insensitive searches. For example, the following query returns "NANCY".

```
TRANSLATE('Nancy')
```

Query	Value Returned
`SELECT DATE(DATE_COL) + 1 DAY FROM TABLE1`	01-20-1992
`SELECT DAY(DATE_COL) FROM TABLE1`	19
`SELECT CHAR(DATE_COL, USA) FROM TABLE1`	01/19/1992
`SELECT MICROSECOND(TSTAMP_COL) FROM TABLE1`	35000
`SELECT CHAR(TIME(TSTAMP_COL), USA) FROM TABLE1`	12:30 PM

Note: The `DATE_COL` is datatype `DATE` and contains a value of 01-19-1992. The `TSTAMP_COL` contains `TIMESTAMP` value of 1992-01-19-12.30.00.035000.

Figure 3. Examples of Time and Date Arithmetic

The following example returns the contents of rows in the table if the NAME column contains combinations of uppercase and lowercase letters, like "SMITH", "Smith", or "SmiTH".

```
SELECT NAME, ADDRESS FROM
   TABLE1 WHERE
   TRANSLATE(NAME)='SMITH'
```

SQLSTATE. This is a common SQL statement return code among IBM's Systems Application Architecture (SAA) relational databases. It is returned to the program as a field of the SQL Communications Area (SQLCA) structure, providing diagnostic information to the application. Only SQL EXEC routines and Start/Stop Using Database fill `SQLSTATE`. No `SQLSTATE` code is generated for other Database Manager Application Programming Interface (API) calls. The messages associated with `SQLSTATE` codes can be found using the online Database Manager Messages, but there is no API that returns `SQLSTATE` code messages.

User-Defined Collating Sequence

Database Manager compares character data using a collating sequence and sorts data according to the sequence. Mainframe systems, such as DB2®, sort using the EBCDIC collating sequence, which sorts differently from OS/2. For example, Figure 4 shows sorts in ascending order.

OS/2	EBCDIC
0000	@@@@
9999	co-op
@@@@	coop
co-op	piano forte
COOP	piano-forte
coop	COOP
piano forte	PIANO-FORTE
PIANO-FORTE	0000
piano-forte	9999

Figure 4. Sort Collating Sequence for OS/2 and EBCDIC

To compensate for this difference in sort order, Extended Services Database Manager allows databases to be created with custom collating sequences. With this ability, databases stored on personal systems can sort according to the EBCDIC collating sequence, providing compatibility with mainframe SQL query results. The collating sequence is specified during Create Database and cannot be changed after the database exists. Sample collating sequences are furnished in the `\SQLLIB` subdirectory in the form of include files. Documentation is provided so that users can create their own collating sequences.

APPN and NetBIOS Support

SQLLOO (a subset of APPC) is a commonly used communications protocol in the OS/2 EE 1.2 and 1.3 environments. SQLLOO is not a supported protocol for Extended Services clients, however Extended Services servers can accept requests from this type of EE client. Support of the APPC protocol has been retained in Extended Services with added support for APPN and NetBIOS. Using NetBIOS simplifies the installation and configuration of database clients and the database server. The NetBIOS protocol is not supported on X.25 or Synchronous Data Link Control (SDLC) lines.

Although APPC and APPN require more configuration information than NetBIOS, they provide more administrative function. Features such as subsystem management, deactivation and reactivation of APPC Transaction Programs (TPs), conversation security, and session security are available when using Systems Network Architecture (SNA) Logical Unit type 6.2 (LU 6.2). Because of changes to Communications Manager, configuring APPC and APPN are simpler in Extended Services than configuring APPC using OS/2 1.2 or OS/2 1.3.

Database Manager Messages

Database Manager Messages is a PM application that assists the operator in identifying messages that occur because of system or user problems, and in obtaining information about resolving the problems. The PM interface for the Database Manager Messages allows users to search for particular error numbers and to print message information.

Most messages are identified by a prefix followed by a number. Some of the prefixes are `SQL`, `QRW`, and `DBM`. `SQLSTATE` messages do not have a prefix.

First Failure Support Technology/2

FFST/2 is a logging facility for critical application errors. It automatically logs errors and pertinent error information needed for problem resolution at the time of error occurrence, minimizing the need to duplicate error scenarios. Logging is not enabled by default. Changes must be made to the `CONFIG.SYS` file for logging to occur.

Additional Network Support

Extended Services includes support for two additional networks: 3174 Peer Communications and Token-Ring Busmaster. Database Manager can perform Remote Data Services using these network types.

Database Environment Expanded to Distributed Databases

IBM SAA Distributed Database Connection Services/2 (DDCS/2) Version 1.0 (announced October 22, 1991) enables transparent read/write access directly to mainframe databases from personal system applications. If data transfer from the mainframe into an OS/2 database is desired, using

DDCS/2 makes efficient use of resources.

DDCS/2 is the first OS/2 offering to exploit the IBM SAA direction for distributed databases by offering remote unit-of-work capabilities in a heterogeneous client/server environment.

The communications link to the mainframe uses APPC, which implements SNA LU 6.2. To use DDCS/2, you must run OS/2 1.30.1 or higher. Also, one of the following program products is required on the host:

- DB2 Version 2 Release 3
- SQL/DS Version 3 Release 3
- OS/400® Version 2 Release 1.1

Connecting to the host database with DDCS/2 is transparent to both users and applications. The host database is simply cataloged and then accessed using the Start Using Database routine, the same way an OS/2 database is accessed. Directory entries in the System, Workstation, and DCS directories resolve where the database actually resides. There is no DDCS/2-specific coding or linking of the application program. OS/2 applications can still access only one database of any type in a single unit of work.

When DDCS/2 connects to a host database, it actually connects to a database subsystem. In DB2, for example, the connection permits access to all databases that the user is authorized to access under that subsystem.

DDCS/2 data transfer can be done in a multiprocessing environment. Here, one OS/2 process SELECTs rows from the host while a second process running concurrently INSERTs the data rows into a Database Manager database. DDCS/2 also supports the functions of Export and Import using PC/IXF format against host databases.

DDCS/2 implements the Distributed Relational Database Architecture (DRDA) connection to supporting mainframe database products. DDCS/2 is available in single- and multi-user versions. The single-user version must be installed on a workstation running either Extended Services or Extended Services with Database Server. The multi-user version requires Extended Services with Database Server and enables concurrent access to mainframe databases from client machines through the server, as well as access to mainframe databases from the server on which it resides. The multi-user version is a cost-effective solution if multiple OS/2, DOS, or Windows clients on a LAN want remote access to a mainframe database.

DDCS/2 supports both static and dynamic SQL. Static SQL gives better performance because the application is bound to the database. This means that the access plan is created before the application executes. DDCS/2 provides cursor support, as expected for relational applications, to allow retrieval of multiple rows and record blocking functions from host databases.

DDCS/2 allows passing platform-specific SQL to the host; however, SAA SQL must be used if the application will run against host and OS/2 databases. Query Manager, a front-end to OS/2 Database Manager, is not supported for use with the DDCS/2 program, but the Command Line Interface provides some front-end functions.

Spanning Databases Over Disk Arrays

OASAS™ I Release 1.0 by Integra Technologies, Inc. interfaces with an IBM Small Computer Systems Interface (SCSI) adapter to manage a disk array containing three to seven IBM SCSI fixed disk drives. It provides fault-tolerant protection in the event of a single fixed-disk failure, and offers the option of having the disk drive array appear as a single large volume to the operating system. OASAS I can be run under OS/2 1.30.1 or higher, or under OS/2 2.0. Because OS/2 recognizes the disk array as one large hard disk drive, Database Manager can store very large databases that span the SCSI drives in the disk array. Databases cannot span drive partitions unless the drives are part of an OASAS I array, and databases cannot span drive partitions.

__Dena M. Laterza__ earned a bachelor's degree from the University of North Texas. She worked two years through the cooperative education program at the IBM Support Center in Dallas. Dena is currently employed by Computer Task Group, Inc., providing Database Manager technical support at IBM in Roanoke, Texas.

NetWare for SAA

Albert Juarez
IBM Corporation
Roanoke, Texas

This article discusses NetWare® for SAA, installation requirements, host connection types, and the different types of NetWare 3270 LAN workstation packages. By following the basic installation tips in this article, system administrators will be able to configure NetWare for SAA properly and efficiently.

NetWare for SAA is the first of a family of communication services to be developed by Novell®. It is a network gateway package that connects LANs to IBM host computers. NetWare for SAA enables NetWare workstations on a LAN to access host applications, and it supports distributed applications that use the LU 6.2 protocol. NetWare for SAA supports simultaneous 3270 display sessions, 3287 printer sessions, and Advanced Program-to-Program Communication (APPC) sessions. It complies with IBM Systems Application Architecture (SAA).

NetWare for SAA comes in three different packages for 16, 64, and 254 sessions. It supports up to 508 concurrent host sessions from a single server when two 254-session packages of NetWare for SAA are loaded on the file server. Different session packages of NetWare for SAA can be intermixed, as shown in the following examples:

- 64-session package + 16-session package = 80 sessions
- 64-session package + 64-session package = 128 sessions

NetWare for SAA is loaded from the NetWare server console as a set of NetWare Loadable Modules (NLMs). NLMs are programs that can be linked to, and unlinked from, the NetWare 3.11 operating system.

In NetWare for SAA, the NLM that provides the Systems Network Architecture (SNA) communication pathway to the host is `NWSAA.NLM`. This NLM formats workstation data destined for the IBM host into an IBM format. The formatted data is then transmitted to the IBM host through the Synchronous Data Link Control (SDLC) or token ring adapter. To the IBM host, the gateway appears as a Physical Unit (PU) Type 2.0 or 2.1 device, depending on configuration. NetWare for SAA supports concurrent PU Type 2.0 and 2.1 host connections. It also supports concurrent LU 1, 2, 3, and 6.2 emulations.

NetWare Communication Services

NetWare for SAA is a component within a framework called NetWare Communication Services. As shown in Figure 1, NetWare Communication Services consists of two parts:

- The Communication Executive, which is the foundation for all communication services
- Communication service modules that are controlled by the Communication Executive

NetWare Communication Services is a group of NLMs. NLMs are linked to and unlinked from the NetWare 3.11 operating system while it is running. The NetWare server console command `LOAD` links NLMs to the NetWare 3.11 operating system. Similarly, the `UNLOAD` command unlinks NLMs from the operating system.

The Communication Executive is linked to the NetWare 3.11 operating system that resides on the file server. (A *file server* is a computer with a multi-user operating system that enables its resources – such as disk space, printers, and communication services – to be shared by the workstations attached to the server.) In turn, the communication service modules are linked to the Communication Executive. Each communication service module consists of NLMs. NetWare for SAA is one of the communication service modules.

The Communication Executive

The Communication Executive, shown in Figure 2, is a set of eight NLMs that provide a core set of functions to both the communication service modules located on the file server and to the client applications located on the workstations. The eight NLMs are shown in Figure 3.

To load the Communication Executive, go to the NetWare 3.11 file server's console screen and enter the command `LOAD COMMEXEC.NLM`. The `COMMEXEC.NLM` program automatically loads the other seven NLMs. The eight NLMs in the Communication Executive are explained in the following sections.

Local Console Agent

The local console agent (`COMMEXEC.NLM`) loads the other seven NLMs and creates the NetWare Communication Services console screen on the NetWare 3.11 file server. The console screen can be viewed by hot-keying to it. The console screen is an interactive screen on the Novell 3.11 file server; there is only one such screen for all NetWare Communication Services. From the NetWare Communication Services console screen, a system administrator can run commands that govern the NetWare Communication Services. The NetWare Communication Services console screen also displays real-time error messages generated by the communication service modules.

Connection Multiplexer

The connection multiplexer (`CNSMX.NLM`) handles the communication among all eight Communication Executive NLMs. For example, the security agent (`CSSECUR.NLM`) uses the connection multiplexer to ask the database agent (`DBA.NLM`) to verify that a user requesting a host resource has the authority to use that resource.

Connection Manager

The Connection Manager (`CM.NLM`) is the interface between the workstation application (3270 emulation) and communication modules. For example, the Connection Manager is called when a workstation application requests a connection to a specific host resource called a Logical Unit (LU). The Connection Manager asks the Service Mapping Agent (`SMA.NLM`) if the requested LU is available. If it is, the Service Mapping Agent assigns a service ID to the available LU. The Connection Manager then asks the security agent (`CSSECUR.NLM`) to verify whether the client workstation requesting the LU is authorized to access it. After availability and access authority have been verified, the Connection Manager communicates again with the workstation application, giving it the service ID and establishing the connection to the requested LU.

Service Mapping Agent

The Service Mapping Agent (`SMA.NLM`) keeps track of LUs and host links. The Connection Manager uses it to verify that a workstation application's request for a host resource matches the available resources. For example, when a workstation application requests a session on a host that has a specific name (for example, CENTRAL), the Connection Manager asks the Service Mapping Agent to verify the existence of the host CENTRAL and the availability of the session on that host.

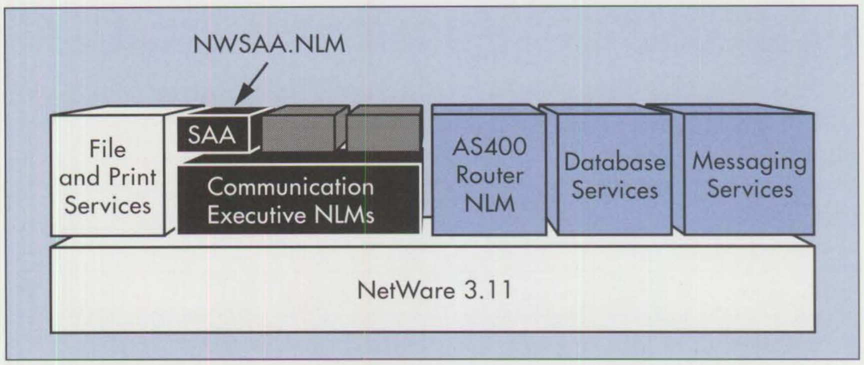


Figure 1. NetWare Communication Services

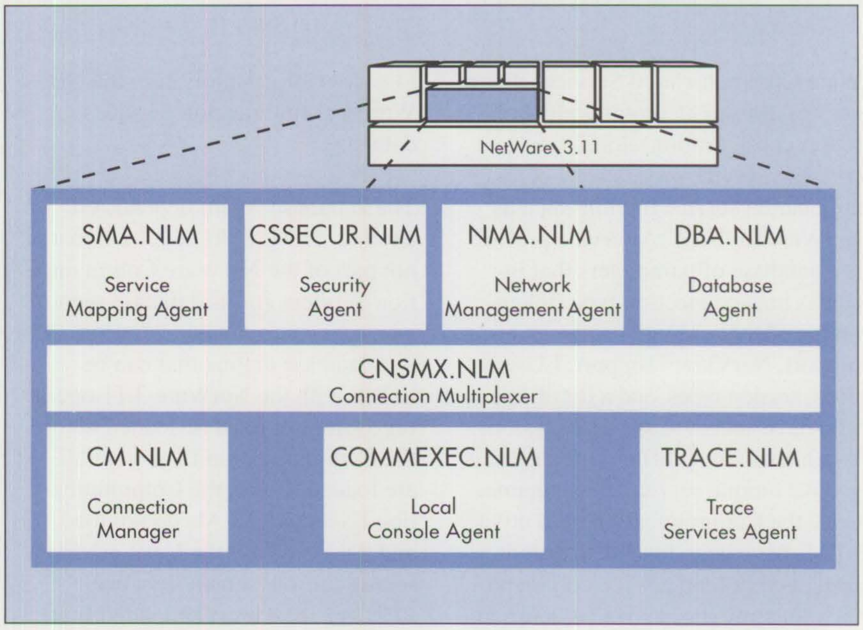


Figure 2. Communication Executive NLMs

Security Agent

The security agent (`CSSECUR.NLM`) ensures that workstation users who request communication sessions have the authority to use those sessions. The security agent checks two levels of security.

The first check involves the NetWare Binderies, a set of database-like files that contain information about users and groups. (In NetWare, a group is a user-defined set of users.) The NetWare Binderies reside on the NetWare 3.11 file server that is linked with the Communication Executive. The security agent checks the NetWare Binderies to verify that the workstation user is a NetWare user.

The second security check verifies that the workstation user is authorized to access the requested communication resource (LU). This authorization and many other configuration parameters are initially set up by the administrator with the Net-

Communication Executive NLM	Function
COMMEXEC.NLM	Local console agent
CNSMX.NLM	Connection multiplexer
CM.NLM	Connection manager
SMA.NLM	Service mapping agent
CSSECUR.NLM	Security agent
DBA.NLM	Database agent
NMA.NLM	Network management agent
TRACE.NLM	Trace services agent

Figure 3. Communication Executive NLMs

Ware Communication Services menu utility `CSCON.EXE` (communication server configuration). `CSCON.EXE` creates a service profile for a communication service module such as NetWare for SAA. A service profile is a database of parameters that includes host connection type (token ring or SDLC), security support, trace support, NetView® support, LU sessions, session types, and a list of users who have access to specific sessions. When a service profile is created, it is given a unique service profile name. Once the Communication Executive NLMs have been loaded, a system administrator can use the command `CSLOAD service_profile_name` to load the service profile. The `CSLOAD` command uses the parameters in the service profile to activate that communication service module.

After passing these security checks, the user can access the requested session, if it is available.

Database Agent

The Database Agent (`DBA.NLM`) manages the NetWare Communication Services database. Created with `CSCON.EXE`, this database contains information about host sessions and which users are authorized to use particular sessions. Other Communication Executive NLMs, such as the security agent, use the Database Agent to access information from the NetWare Communication Services database.

The Database Agent depends on another NLM (`BTRIEVE.NLM`) that is not part of the NetWare Communication Services for its database record management capability. `BTRIEVE.NLM` is a database engine that can be linked with the NetWare 3.11 operating system. It must be loaded before the Communication Executive NLMs are loaded. If not, the Communication Executive NLMs do not load, and a message on the NetWare file server console screen says that `BTRIEVE.NLM` must be loaded first.

Network Management Agent

The Network Management Agent (`NMA.NLM`) provides management functions for the NetWare Communication Services platform. It interfaces with IBM's NetView host management facility and Novell's Communication Services Manager. The latter is an application that monitors real-time alerts generated by a server and the NetWare Communication Services. NetWare for SAA monitors NetView token ring, SDLC, and Logical Link Control (LLC) alerts. Also monitored are NetView alerts for NetWare File Services. Host and LAN administrators can use these real-time alerts to quickly identify and solve problems on file servers. As an example, when an error (such as a disk-full condition or a token ring adapter malfunction) occurs on a file server, the Network Management Agent traps the error and forwards it to the Communication Executive. There, an alert is generated and transmitted to NetView on a host or to a Novell Communication Services Manager application running on a workstation connected to the network.

Trace Services Agent

The Trace Services Agent (`TRACE.NLM`) manages the diagnostic trace facilities provided by NetWare Communication Services. Using the menu utility `CSCON.EXE`, a system administrator can set the parameter `TRACE=ON` in a service profile. This means the corresponding communication service module is traceable. The trace services agent can then display, on the NetWare Communication Services console, the data flow between the host and server. It also can save the data flow information in a log file.

NetWare Communication Services Console Commands

Figure 4 lists the commands that can be entered from the Communication Services console prompt `CS` at the file server.

Installing NetWare for SAA

Installing NetWare for SAA is a three-step process. First, decide whether to configure NetWare for SAA on a dedicated machine or integrate it with an existing NetWare 3.11 file server. Next, use the NetWare for SAA installation utility (`CSINSTAL.NLM`) to load all the NetWare Communication Services files onto the file server. The third step is to configure the NetWare Communication Services using the configuration utility `CSCON.EXE`.

The NetWare for SAA package contains the complete set of NetWare

3.11 manuals and the *NetWare for SAA Communication Services Administration Guide*. NetWare 3.11 manuals are included because NetWare for SAA comes with a runtime version of NetWare 3.11. The NetWare 3.11 runtime version is a fully functional NetWare operating system, but is a single-user version, permitting only one client connection.

An additional NLM (`NLICLEAR.NLM`) comes with the runtime version of NetWare 3.11. Its purpose is to terminate unauthenticated connections to the file server. This NLM is very useful when a user attaches to the server by loading the NetWare shell (`IPX.COM` and `NETX.COM`) but never enters a login name and password. If the administrator or the `AUTOEXEC.NCF` file loads `NLICLEAR.NLM` at the NetWare server console after the default 60 seconds or after the duration specified by the user, the NLM will terminate the client connection. If `NLICLEAR.NLM` is not loaded, the only client connection available will be taken up, which means that no one can log in to the server to do administrative tasks. The `AUTOEXEC.NCF` is an ASCII file created by an administrator. This file has an administrator-specified list of commands that the NetWare operating system executes during its boot process.

Another option to use with `NLICLEAR.NLM` is `SET REPLY TO GET NEAREST SERVER = OFF` in the `AUTOEXEC.NCF`. Setting this parameter to `OFF` ensures that the runtime version of NetWare 3.11 will not respond to GetNearestServer requests from the LAN workstations.

The runtime version of NetWare 3.11 is useful when configuring NetWare for SAA as a dedicated configuration rather than an integrated configuration. If NetWare for SAA is to be installed as a dedicated configuration, the file server must be a 386SX-, 386-, or 486-based computer. It should have a minimum of 8 MB of RAM and a 20 MB hard disk to accommodate NetWare for SAA and the runtime version of NetWare 3.11. The first step is to install the runtime version of the NetWare 3.11 operating system. (The runtime version installs the same way that the full NetWare 3.11 product installs.)

Second, use the installation NLM (`CSINSTAL.NLM`) to install the NetWare Communication Services – NetWare Communication Executive and NetWare for SAA.

In an integrated configuration, the runtime version of NetWare 3.11 already exists on the file server. Only the NetWare Communication Services need to be installed, so the minimum

Command	Purpose
CSLOAD/CSUNLOAD	Loads/unloads a communication service profile created with the `CSCON.EXE` utility.
CSDOWN	Shuts down the Communication Executive, terminates all sessions, and unloads all service profiles. If there are two NetWare for SAA (254-user) modules, both would be shut down.
CSRESTART	Shuts down the Communication Executive and restarts it automatically (to help with troubleshooting). It also terminates all sessions, and unloads and reloads service profiles.
CSSTATUS	Displays information about NetWare Communication Services, such as LU sessions, their status (active or inactive), and host connection status (connected or disconnected).
CSSTART/CSSTOP	Starts or stops a trace, which records the SNA data stream between the server and a host.
CSDISPLAY SERVICES	Displays information about the NetWare Communication Services loaded.
CSBIND/CSUNBIND	Binds and unbinds transport protocols (such as SPX, TCP/IP, and AppleTalk®) to the Communication Executive.
CSNAME	Displays the Communication Executive name. This is the name of the file server on which the communication service modules run. The name is assigned when `COMMEXEC.NLM` is linked to the file server.
CSVER	Displays the Communication Executive version number.
CSSET	Sets network management options, such as NetView support, audit trail, security, and traces.
CLS/OFF	Clears the NetWare Communication Services console screen.
CSCLEAR LOG	Clears the NetWare Communication Services event log. The event log is a file created by the Communication Executive that records errors at the file server.
CSPARAM	Sets Communication Services parameters.

Figure 4. NetWare Communication Services Console Commands

1. Calculate the memory requirement (M) for each volume.

   For each DOS volume:

   M=.023 x volume size (in MB) / block size (the default block size is 4)

   For each volume with added Name Space:

   M=.032 x volume size (in MB) / block size (the default is 4)

   A Name Space is an NLM used in NetWare 3.11 to allow storage of non-DOS files (such as Macintosh® files) in a NetWare volume.

2. Add memory requirements for all volumes.

   Total volume memory = $M_{sys:}$ + $M_{vol1:}$ + $M_{vol2:}$

3. Add 2 MB for the operating system, and round the value up to the next MB. Use at least 4 MB.

4. Add 4 MB (to run NetWare for SAA) to the value determined in Step 3. Also, add 20 KB for each session and round up to the nearest MB.

Figure 5. Steps to Determine Memory Requirements for Integrated Configuration

Number of Workstations	16	32	64	80	128	253	506
Number of LUs	16	32	64	80	128	253	506
Maximum CPU Utilization	5.5%	5.6%	5.8%	6.0%	7.0%	11.6%	15.3%

Note: This data was measured using MONITOR.NLM with the -P parameter. Values will vary with different computer systems.

Figure 6. CPU Utilization in an Integrated NetWare for SAA Configuration

hard disk requirement is 10 MB. The formula in Figure 5, which comes from Novell's *NetWare for SAA 1.2 Rules of Thumb* booklet, calculates the minimum memory requirements for an integrated configuration.

If you are using the standard NetWare 3.11 (not the runtime version), use the following steps to determine the memory requirements for running NetWare for SAA.

Dedicated versus Integrated Configuration

In a dedicated NetWare for SAA configuration, the file server running the runtime version of NetWare 3.11 is dedicated solely to NetWare Communication Services. This can be an advantage if there will be many file transfers to, or many sessions on, a host system. A dedicated configuration also can be useful when existing file servers are running NetWare 2.2 or NetWare 2.1X. The disadvantage of a dedicated configuration is the cost of adding another 386SX/386/486 computer as the file server running NetWare for SAA.

In an integrated NetWare for SAA configuration, an existing NetWare 3.11 file server is used, so there is no need to invest in an additional computer. If there will be 64 or more host sessions, an integrated configuration might be appropriate. A file server's utilization is affected slightly by the number of concurrent sessions. Other factors (such as the number of users logged on to the NetWare file server, additional application NLMs, and heavy disk I/O activity) also increase file server utilization. Figure 6, also from Novell's *NetWare for SAA 1.2 Rules of Thumb*, shows the influence of the number of sessions on the maximum CPU utilization in a dedicated NetWare for SAA configuration.

NetWare for SAA Connection Types

NetWare for SAA supports a variety of connections, as illustrated in Figure 7. Each connection is discussed below.

SDLC Host Link

If a configuration requires a remote connection to a 37X5 frontend processor, a 9370 host, or an AS/400® (see Figure 7, area B), it is possible to configure NetWare for SAA to use an SDLC adapter as the interface. NetWare for SAA uses up to two SDLC adapters in one file server for concurrent access to two remote hosts. The types of SDLC adapter cards supported by NetWare for SAA are IBM's Multiprotocol Adapter/A, Novell's NetWare for SAA Synchronous Adapter (which comes with two cables: one for RS-232, the other for V.35), and Novell's Synchronous/V.35 Adapter. SDLC adapters function at different line speeds. IBM's Multiprotocol Adapter/A and Novell's NetWare for SAA Synchronous Adapter with the RS-232 cable work with NetWare for SAA at 19.2 Kbits per second. Novell's NetWare for SAA Synchronous Adapter with the V.35 cable works with NetWare for SAA at speeds of 19.2 Kbits per second to 64 Kbits per second.

Token Ring Host Link

If a configuration requires a token ring connection to a 3174 cluster controller, a 37X5 communication controller, a 9370 host, or an AS/400 (as in Figure 7, item C), it is possible to

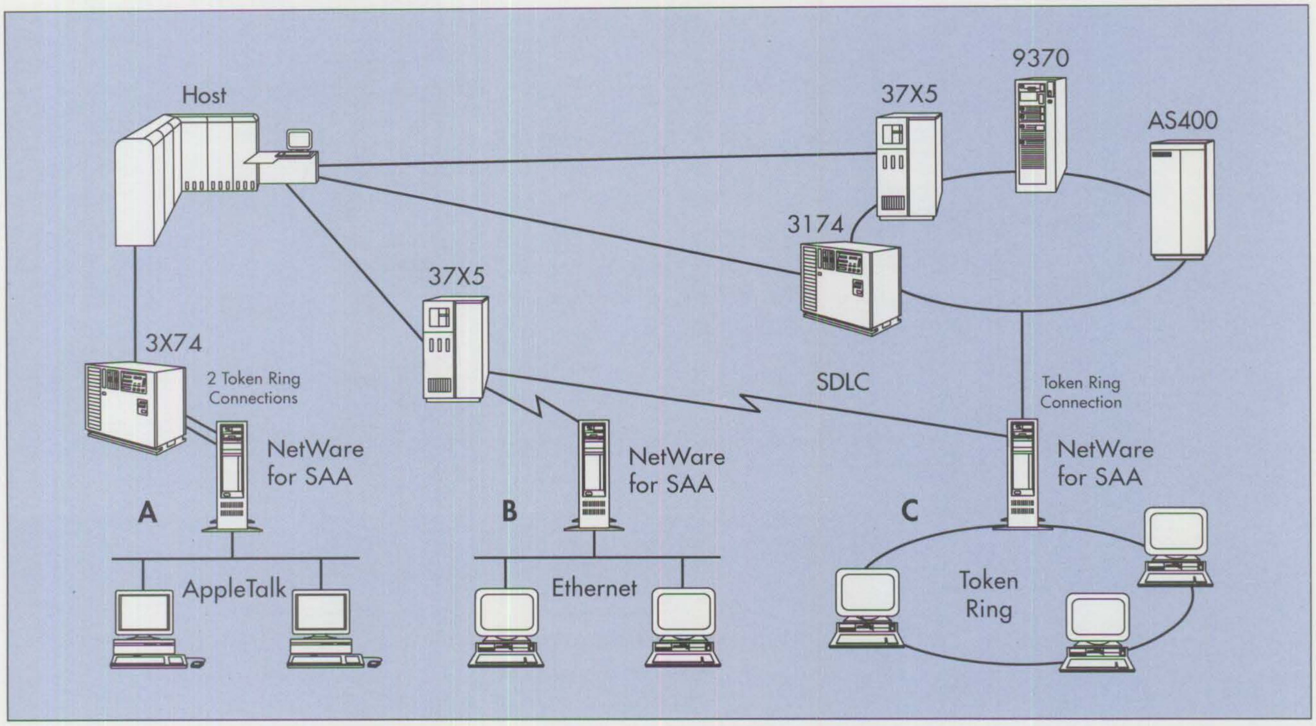


Figure 7. Netware for SAA Connection Types

configure NetWare for SAA to use a token ring adapter as the interface. The types of token ring adapter cards supported by NetWare for SAA are the IBM Token-Ring 16/4 Adapter (for the AT bus) and IBM Token-Ring 16/4 Adapter/A (Micro Channel bus). Two token ring adapters must be used when configuring NetWare for SAA for two host connections. The token ring configuration can run at data transfer speeds of 4 MB or 16 MB.

It is possible to configure a file server running NetWare for SAA to use one token ring card for SNA traffic and Inter Packet Exchange (IPX) traffic. However, for optimal performance it is preferable to use two token ring cards: one to handle the SNA traffic on the host link, and the second to handle the IPX traffic with the workstations, as illustrated in Figure 8.

Connecting to an AS/400

To allow NetWare for SAA to communicate with an AS/400 using a token ring or SDLC connection, define the communication service as a PU Type 2.0, which handles LU 2 and LU 3 display sessions. Use the `CSCON.EXE` menu utility to configure this connection when creating or editing a service profile. On the AS/400, the file server running NetWare 3.11 and NetWare for SAA is configured as if it were a cluster controller, and the workstations are configured as 3270 devices.

Unlike Twinax and 5250 SDLC gateways, which provide seven and nine concurrent host sessions, respectively, NetWare for SAA can have up to 128 concurrent AS/400 sessions with two PUs. (Although NetWare for SAA can accommodate even more concurrent sessions, the AS/400 is limited to 64 sessions per PU.) When connecting to an AS/400 with NetWare for SAA, the workstations use Novell's 3270 LAN Workstation program, rather than a 5250 emulation program, for 3270 emulation. Novell's 3270 LAN Workstation program also supports LU Type 2 3278/79 display sessions and LU Type 1 3287 printer sessions.

Mixed LAN Configurations

In NetWare for SAA, workstations running Novell's NetWare 3270 LAN Workstation program can be attached to any type of LAN topology using IPX or AppleTalk (ARCNET®, Ethernet, token ring, and LocalTalk®). This feature can be useful if users on an Ethernet NetWare LAN want to have 3270 host sessions (see Figure 7, area B), or if users on Macintosh computers attached to a NetWare file server through a LocalTalk network want to have 3270 sessions (see Figure 7, area A).

NetWare for SAA Version 1.2

Novell recently announced Version 1.2 of NetWare for SAA. Version 1.2 contains significant enhancements to AS/400 support. For connecting to an AS/400 with NetWare for SAA, Novell has added the capability of using IBM's PC Support/400 pro-

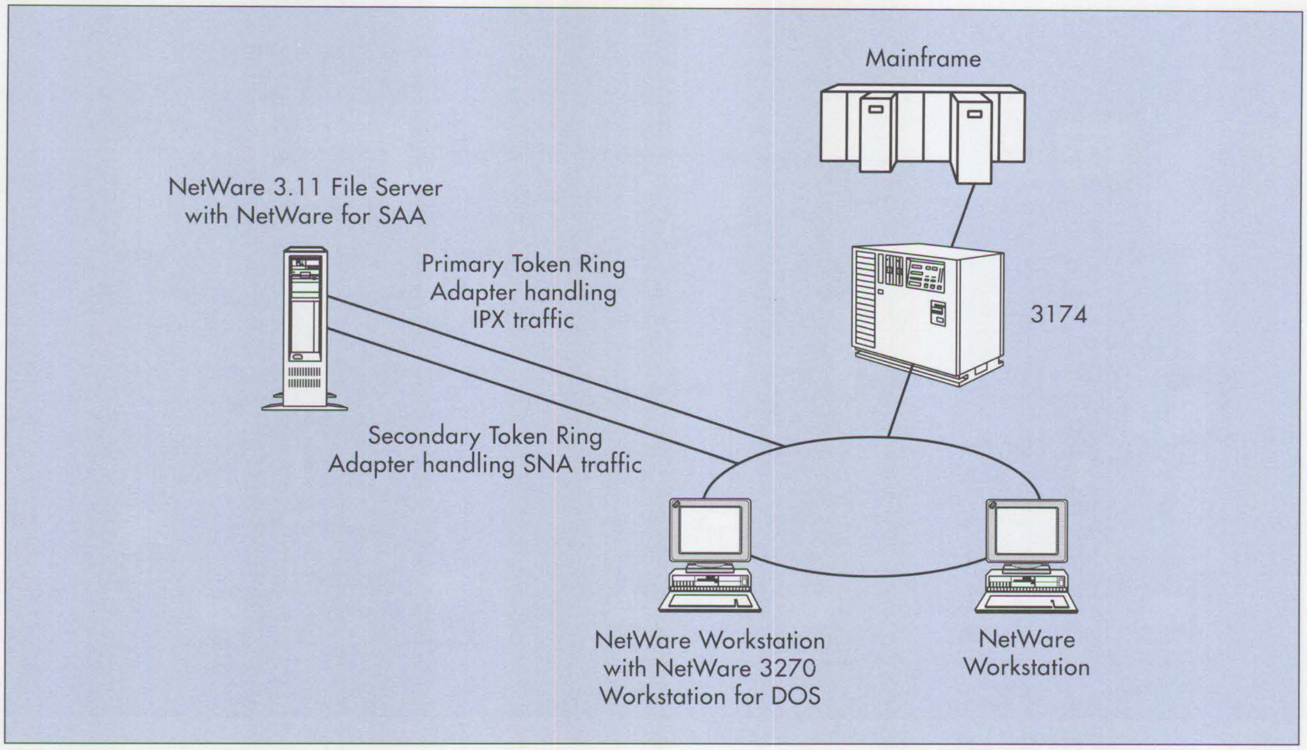


Figure 8. Dual Token Ring Adapter Configuration

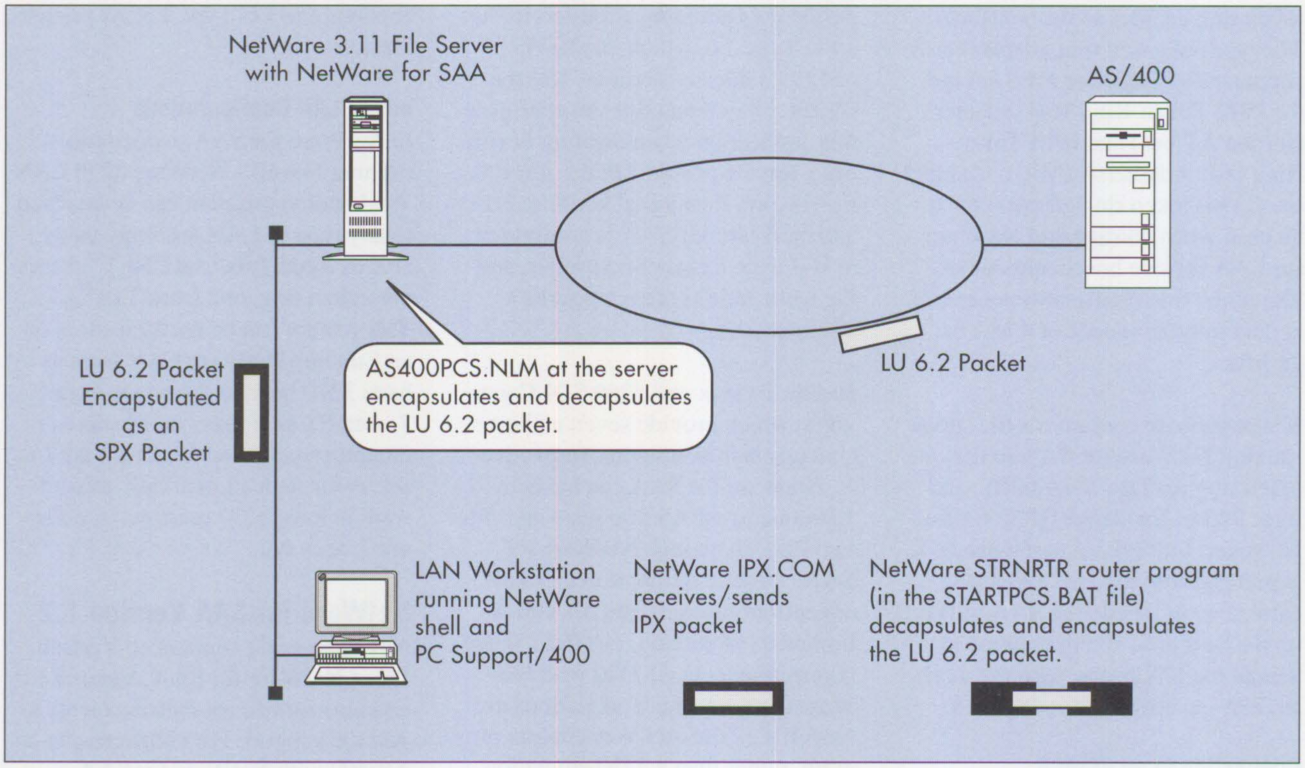


Figure 9. Novell Router for AS/400 PC Support

gram for 5250 emulation. Also, as shown in Figure 9, a new NLM (`AS400PCS.NLM`) encapsulates an LU 6.2 packet coming from the AS/400 host into a Sequential Packet Exchange (SPX) packet. The SPX packet is then routed to the LAN workstation that is running the NetWare shell and PC Support/400. Also at the workstation is the NetWare for SAA Version 1.2 router program `STRNRTR.EXE`, which replaces the `STARTRTR.EXE` program in the PC Support/400 `STARTPCS.BAT` file. `STRNRTR.EXE` uses `IPX.COM` (which is part of the NetWare shell on the workstation) to receive the SPX packet. `STRNRTR.EXE` then decapsulates the SPX packet into an LU 6.2 packet.

To put `STRNRTR.EXE` into the PC Support/400 `STARTPCS.BAT` file, first install PC Support/400 normally (without running the verification test), then run the NetWare for SAA 1.2 `CFGNRTR.EXE` program to modify the `STARTPCS.BAT` file.

Administrators can use the `CSCON.EXE` utility program to configure the NetWare for SAA 1.2 communication service as a PU Type 2.1.

Workstation Software Requirements

To interface with NetWare for SAA, the client program NetWare 3270 LAN Workstation is necessary. NetWare 3270 LAN Workstation (for DOS, Windows, or Macintosh) is a 3270 emulation package sold by Novell. It is not included with NetWare for SAA. The 3270 emulation package uses IPX/SPX or AppleTalk to communicate with NetWare for SAA.

NetWare 3270 LAN Workstation for DOS is a DOS-based workstation application that uses NetWare for SAA to provide IBM host connectivity. It supports up to five concurrent host sessions. The five host sessions can be any combination of display, printer, or APPC sessions. This application resides in memory. Through its hot-key function, a user can move among the active sessions, each of which occupies a full-screen display. IBM 3270 display models 2 through 5 with Extended Attributes (including extended, highlighted, seven colors, and APL/APL2) are supported. Additionally, IBM 3287 LU Model 2 printer sessions are supported. The package also comes with send/receive utilities and a keyboard remapping utility, `KEYDEF.EXE`. For application development, APPC support and the Application Programming Interface (API) are included. NetWare 3270 LAN Workstation for DOS uses the IPX/SPX transport protocol to communicate with NetWare for SAA.

Package	Available From	Purpose
Rumba®	Wall Data	3270 emulation for Windows
Rumba/PM	Wall Data	3270 emulation for OS/2
LINKix	Cleo Corporation	3270 emulation for UNIX
Rumba/400	Wall Data	5250 emulation for Windows
IBM PC Support/400	IBM	5250 emulation for DOS

Figure 10. 3270 and 5250 Emulation Packages Supported with NetWare for SAA 1.2

NetWare 3270 LAN Workstation for Windows is a Microsoft Windows 3.X-based workstation application that uses NetWare for SAA to provide IBM host connectivity. Up to 26 sessions can be concurrently active. Under Windows, each host session occupies a separate window that can be resized and repositioned anywhere on the screen. The host sessions can be any combination of display, printer, or APPC sessions. This application otherwise provides the same features as NetWare 3270 LAN Workstation for DOS.

NetWare 3270 LAN Workstation for Macintosh runs under Version 2 of the Macintosh operating system 6.03 or above. This workstation application uses NetWare for SAA to connect a Macintosh workstation to an IBM host computer. Up to 26 host sessions can be concurrently active. Each host session occupies a separate window that can be resized and repositioned anywhere on the screen. IBM 3270 display models 2 through 5 with Extended Attributes are supported. This package uses the AppleTalk Data Stream protocol to communicate with NetWare for SAA. Neither host printing (LU 1 and 3) nor APPC support is possible in the first release of this package. Figure 10 shows other 3270 and 5250 emulation packages supported with NetWare for SAA 1.2.

__Albert Juarez__ is a market support representative specializing in IBM LAN Server and Novell NetWare support. He joined IBM in 1991 with five years of personal systems networking experience. Albert was educated at DeVry Institute in Texas and is a Novell Certified NetWare Engineer.

Using the IBM DOS 5.0 Driver EMM386.EXE and Upper Memory

Jeanne Ann Smith
IBM Corporation
Dallas, Texas

This article contains sample statements and parameters for configuring a PS/2 386- or 486-based Micro Channel system under DOS 5.0 to use expanded memory and upper memory via the EMM386 driver.

The DOS 5.0 driver EMM386.EXE can be used in 386- and 486-based computers as an expanded memory manager, an upper memory provider, or both.

To access the upper memory area, you must specify a parameter on the EMM386.EXE statement in your CONFIG.SYS file. If you want to use upper memory but do not need expanded memory, use the NOEMS parameter; if you need both upper and expanded memory, use the RAM parameter. To use upper memory, you also must load HIMEM.SYS before EMM386.EXE, and load DOS=UMB (or DOS=HIGH,UMB) in your CONFIG.SYS file. Figure 1 shows the sequence of lines needed in the CONFIG.SYS file and some example statements for device drivers that use expanded memory and upper memory.

The HIMEM and EMM386 statements must be loaded before any DEVICEHIGH= statements can be used.

If you do not use either the NOEMS or the RAM parameter, you can access only expanded memory. If the EMM386 driver is used as an expanded memory manager, it automatically tries to configure the expanded memory to meet the LIM EMS 3.2 specification. This means that the EMM386 driver tries to find a 64 KB frame of memory in the area from C000 to DFFF. (A page is a 16 KB area of memory; a frame consists of four contiguous 16 KB pages, totaling 64 KB of contiguous memory.) EMM386 may not be able to find an available 64 KB frame of memory between C000 and DFFF, because installed devices also use that range of addresses. If no 64 KB frame of memory is free, you will receive the following error message:

```
EMM386 not installed—unable
to set page-frame base
address
```

To determine which frames and pages of memory are available, boot with the Reference Diskette for your Micro Channel PS/2 system, and select Set Configuration – Display Memory Map. This selection indicates which address locations are taken by devices and which are free. This helps to determine the available frames and pages. Addresses that can be used for frames and pages range from C000 to DC00 in increments of 400 hex (or 16 KB); that is, locations C000, C400, C800, CC00, D000, D400, D800, and DC00. The interval between any two of these locations is 16 KB. Thus, a 16 KB page can start at any of these addresses; a 64 KB frame also can start at any of these addresses, but it must be able to occupy four contiguous 16 KB pages.

If your applications require expanded memory but do not specifically need LIM EMS 3.2 expanded memory, there is no need to have a frame available. Use the parameter Pn=address to specify individual 16 KB pages for EMM386 to use. Each time you use this parameter, you must substitute a number in place of "n" and specify an actual address of a page within memory. For example, if the address range C000-C7FF is available in your system, you can use pages C000 and C400. The parameters for including these addresses would therefore

Sequence of Lines	Description

DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM
FRAME=C000
	The HIMEM.SYS driver must precede the EMM386.EXE statement.

DOS=HIGH,UMB	The HIGH parameter enables DOS to load itself into the high-memory area. The UMB parameter enables DOS to use the upper memory area.
DEVICE=C:\DOS\RAMDRIVE.SYS 1024 /A	The RAMDRIVE driver is loaded into conventional memory, but the RAMDRIVE itself is created in expanded memory (/A).
DEVICEHIGH=C:\DOS\ANSI.SYS	The ANSI.SYS driver is loaded into upper memory because the DEVICEHIGH= statement is used.

Figure 1. CONFIG.SYS for Accessing Upper Memory

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM P0=C000 P1=C400
DOS=HIGH,UMB
```

Figure 2. CONFIG.SYS for Accessing Expanded Memory Other Than LIM EMS 3.2

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM FRAME=C400
```

Figure 3. CONFIG.SYS for Accessing LIM EMS 3.2 Expanded Memory

be P0=C000 P1=C400. These parameters would be added to the end of the EMM386 statement in `CONFIG.SYS`, as shown in Figure 2.

If the application requires expanded memory that conforms to LIM EMS 3.2, it is necessary to use one of the frame parameters: `Mx`, `FRAME = address`, or `/pADDRESS`. For example, if the address range C400–D3FF is available in your system, use frame C400. The parameter to add to the end of your EMM386 statement will be either `M2`, `FRAME=C400`, or `/pC400`. The addresses that correspond to the `Mx` parameter are given on page 650 of the *IBM DOS 5.0 User's Guide and Reference*. Figure 3 shows an example of the `FRAME=C400` parameter.

The sizes of upper memory blocks are determined by the addresses taken by devices and the addresses used by the EMM386 driver. If the addresses used by installed devices are neither contiguous nor very close together, small pieces of upper memory may be available, but you may not have a 64 KB area of free memory. However, you can create larger blocks of free memory by relocating the addresses taken by your devices so that they are contiguous, very close together, or in the same memory block (either C000 or D000). To do this, boot the system with the Reference Diskette, then select Set Configuration – Change Configuration. In this selection, the devices whose addresses can be relocated are listed with square brackets around their current addresses. To change address locations, use the PF5 and PF6 keys to scroll through the alternate address locations that can be used by your devices. If an asterisk appears next to an alternate address, then there is a conflict with another device's address, so that address cannot be used.

The LOADHIGH (LH) command is used to load memory-resident programs into upper memory.

To load device drivers into upper memory, the `DEVICEHIGH=` statement (not the `DEVICE=` statement) is used in `CONFIG.SYS` as shown in Figure 1. The `LOADHIGH (LH)` command is used to load memory-resident programs into upper memory. However, before device drivers or programs can be loaded into upper memory, an upper memory block that is large enough must be available. If it is not, the driver or program will automatically be loaded into conventional memory.

Not all device drivers and programs run successfully in upper memory. For example, device drivers that allocate additional memory after startup may not run properly in the upper memory area. You should load each device driver and program into upper memory, one at a time, to ensure that they run successfully. If the system hangs or you receive errors when loading the item into upper memory, reboot using a DOS diskette and edit the `CONFIG.SYS` or `AUTOEXEC.BAT` file to load the driver or program into conventional memory.

For more information about conventional, extended, expanded, upper, and high memory, refer to the *DOS 5.0 User's Guide and Reference* (order number 84F9779), Chapter 12, "Optimizing Your System." More information about the parameters that can be used with the `EMM386.EXE` driver is on pages 649-654 of the *Guide*.

Note: This article applies only to PS/2 386- and 486-based Micro Channel computers. PS/2 models 35, 40, and L40 are 386-based, but they are AT-bus systems. To change the address locations of adapters installed in those systems, you must physically change the switch settings on the adapters. To determine which switch settings are used for each address location, consult the reference manuals provided with the adapters.

***Jeanne Ann Smith** works at IBM's OS/2 Application Assistance Center in Roanoke, Texas. She has supported DOS and OS/2 for three years in a cooperative education program. Jeanne is studying mathematics and computer science at the University of Texas at Arlington.*

Did You Know...?

- Advanced applications such as object-oriented programming, multimedia, and distributed computing require the versatility and reliability of OS/2 2.0.
- OS/2 2.0 is the first workstation operating system to fully exploit the features of the 386/486 family of processors.
- OS/2 2.0 runs Windows Versions 2.0 and 3.0 applications—something Windows 3.0 cannot do. OS/2 2.0 provides the broadest selection of applications in the industry, running more than 20,000 DOS, 5,000 Windows, and 2,500 16-bit OS/2 applications unchanged.

OS/2 Systems Migration Considerations can help!

OS/2 Systems Migration Considerations can help you take full advantage of the power of OS/2 2.0! The book provides detailed technical information about migrating to OS/2 2.0, Extended Services, and LAN Server 2.0 environments. It also covers features of previous OS/2 versions and helps you locate them in 2.0, and describes features of Microsoft Windows included in 2.0.

YES!

Mail orders to:

The TDA Group
P.O. Box 1360
Los Altos, CA 94023

Send me _____ copies of
OS/2 Systems Migration Considerations
at $18.95* plus $3.95 shipping and handling,
a total of $22.90** per copy.

□ Charge to my VISA/MasterCard

□ Check □ Money Order

*Discounts available for multiple copies
**California residents add applicable sales tax

Name ______________________________

Company ______________________________

Address ______________________________

City/State/Zip ______________________________

Phone ______________________________

VISA/MasterCard # ______________________________ Expires ____/____

For faster service, order toll-free by calling (800) 551-2832,
or fax this form to (415) 948-4280.

The Solutions Evaluation Tool

Jamie Hughey, Alan Harrison, and Boyd Jasperson
IBM Corporation
Dallas, Texas

This article describes the Solutions Evaluation Tool (SET). SET is part of Validation Services, an IBM offering that provides guidance and the necessary technology for testing applications in a LAN environment.

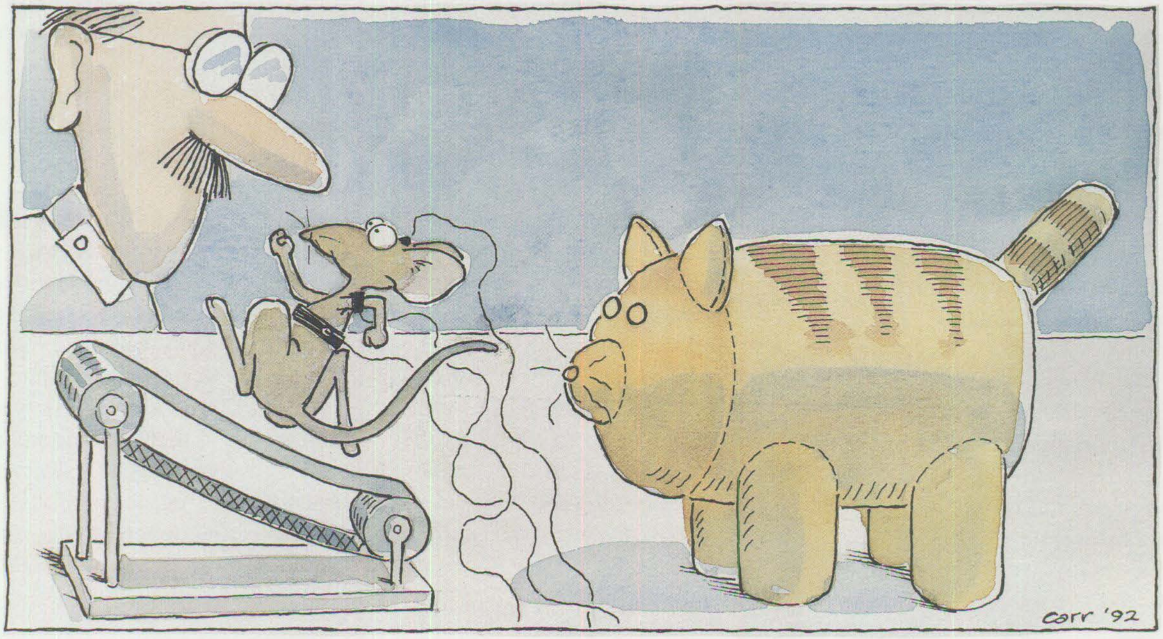
Application development in a distributed computing environment – such as a PS/2 system on a LAN communicating to a host system – can be complex and unpredictable. Automated, repeatable testing procedures can identify problems that may not be evident from manual testing.

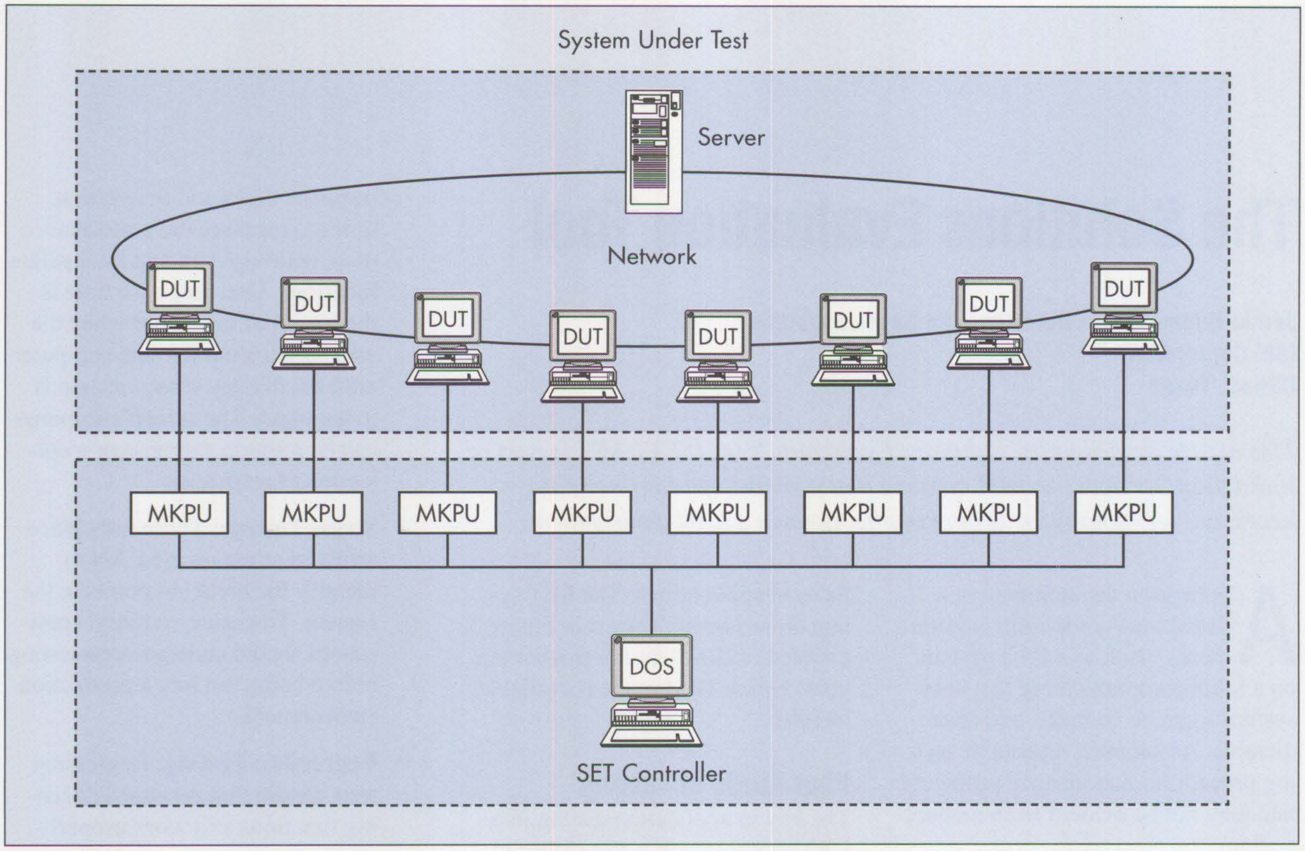
IBM created the Solutions Evaluation Tool (SET) to automate testing of applications under development and to repeatedly measure the use of mouse, keyboard, and graphical interfaces in applications. The SET system environment, shown in Figure 1, evolved in IBM laboratories for ten years before IBM made it available in 1991.

Four Kinds of Testing

The key to application testing in a LAN environment is the ability to collect repeatable measurements. SET allows developers to test applications in four ways:

- **Performance and Capacity Testing:** These tests collect user response times and transaction times to measure the performance of an entire system and its separate functions. User response time is the length of time from when the user submits input to the computer until the display screen responds to the input. The screen's response can be a single response or a collection of responses.
- **Stress Testing:** These tests place artificial stress on the LAN to identify the breaking points in the system. To ensure stability, applications should undergo stress testing before being put into a production environment.
- **Regression Testing:** Regression tests ensure that previously existing functions still work properly when changes are made in the environment.
- **Integration Testing:** Integration tests ensure smooth integration and interoperability of various applications on the network.





DUT: Device Under Test
MKPU: Mouse and Keyboard Processor Unit

Figure 1. The SET Environment

SET Fundamentals

SET measures mouse, keyboard, and graphical interfaces, and affords other advantages to developers testing applications in a LAN environment. It consists of hardware and software that are separate from the application and workstation being tested and measured. By running in this non-intrusive environment, SET does not affect the operation and performance of the application and workstation under evaluation. The measurements that SET collects are "clean" – they are not influenced by the presence of the testing tool, as in intrusive environments.

SET uses *workload scripts* – basic groups of measurable, repeatable activities – to involve the application being measured. These scripts run in the special SET environment, but no SET processes run in any part of the system under test.

SET can control and test a LAN from a single point within the network. The SET controller system connects to the keyboard, mouse, and video ports of the workstations under test. Each SET controller (a PS/2 running the SET software) can drive up to eight workstations, using independent workload scripts for each workstation. A group of SET controllers can be centrally managed using either a daisy-chain or token ring controller approach. Testing staff can add more SET controllers and test workstations, yet still manage them centrally.

Transaction rates can be fixed (to include all transactions executed) or random (to select individual transactions). Keying rates can be changed to emulate the expected keying rate of real workloads. Think times can be set by using delays that can vary throughout the scripts.

Figure 2 shows the Mouse and Keyboard Processor Unit (MKPU), which intercepts keyboard and mouse connections to the computer being tested and provides keyboard and mouse emulation. Its adapter ports are labeled J1 through J8. The keyboard and mouse of the workstation being tested connect to MKPU adapter ports J3 and J5, respectively, instead of connecting directly to the computer. Ports J1 and J4 connect the

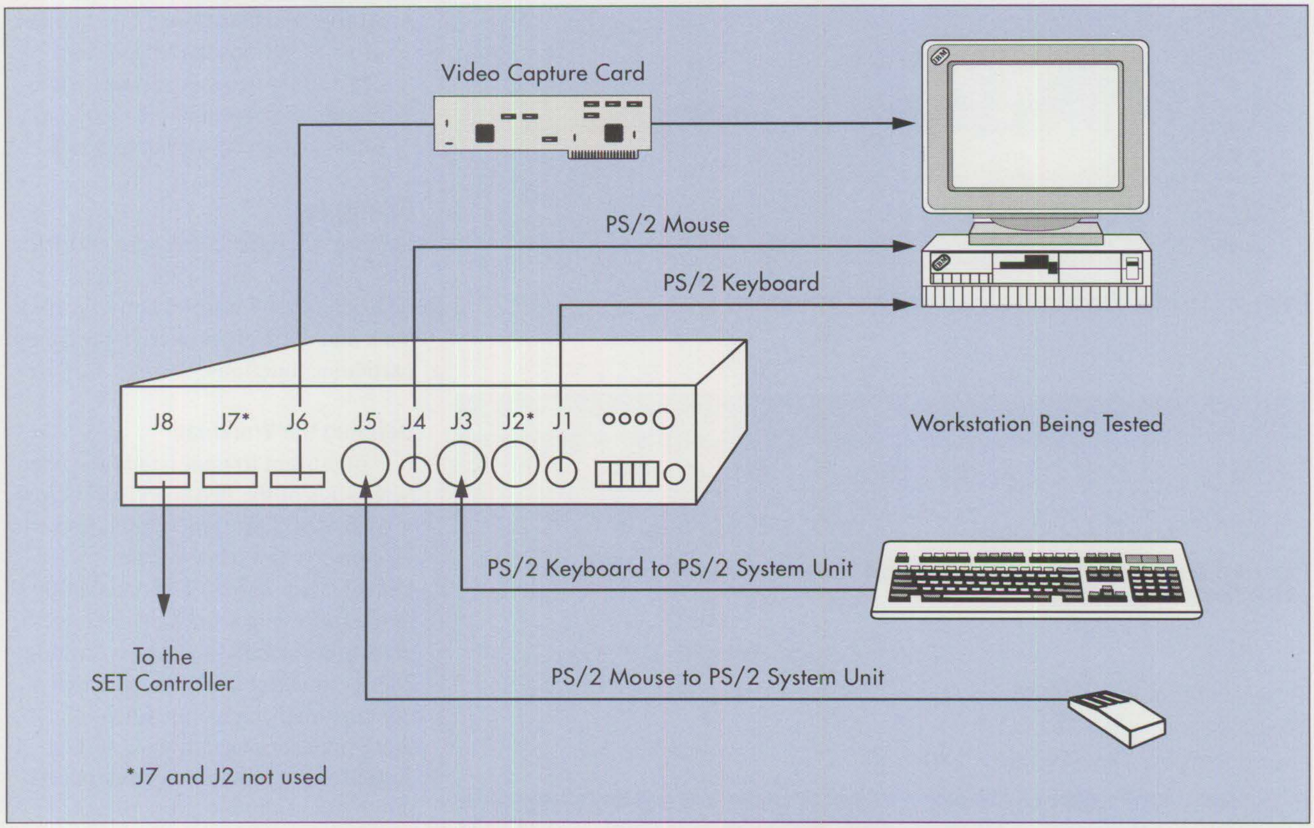


Figure 2. The MKPU Intercepting the Mouse and Keyboard Connections on the Workstation Being Tested

MKPU to the keyboard and mouse ports, respectively, of the computer under test. The simulated user's keystrokes at the workstation being tested are captured and transmitted to port J3 of the MKPU. In turn, the MKPU uses port J1 to send the captured keystrokes to the system under test. Similarly, the simulated user's mouse movements are captured and transmitted to MKPU port J5. The MKPU uses port J4 to send the captured mouse clicks to the computer being tested. Port J8 sends data back to the computer running SET. Ports J2 and J7 are not used.

Display activity is monitored by installing the SET video capture card in the auxiliary video slot of the workstation being tested, as shown in Figure 2. This display capture card is not recognized in the `CONFIG.SYS` of the workstation being tested. SET uses *screen verification* to record response times. The video capture card records a part of the display of the computer under test. SET then replays the workload script and compares this recorded image with the image on the workstation during playback. A success will occur only if the two images match pel for pel.

SET runs on any PS/2 Micro Channel computer and it can test any application on a workstation. It is not sensitive to communication protocol changes or application code changes, as long as screen images remain the same.

Basic Indicators

SET's basic indicators are response time, throughput, and number of active workstations. These indicators are defined as follows.

- **Response time:** The elapsed time for a system to complete an interactive function as viewed by the workstation user. Response time is measured in seconds and indicates the overall speed of the LAN environment.
- **Throughput:** The number of transactions or events completed in a given period while maintaining a reasonable average response time. The capability to increase transaction volumes depends on the desired response times for the total system and how the application handles longer response times. The overall system capacity is a measure of the maximum throughput that the system can achieve while providing the desired average response time.

Transaction	Frequency	Number of Send and Receive Requests to the Host (bytes)	Number of Requests Required by the Transaction
Cash Check	40%	1 (64)	2
Deposit	40%	1 (56)	2
Payments	20%	1 (51)	2

Figure 3. Sample Workload Mix in a Bank Teller Environment

Server Memory	Number of Active Workstations					Transaction Intervals (seconds)				
12 MB	4	8	12	16	20	40	60	75	90	105
16 MB	4	8	12	16	20	40	60	75	90	105

Figure 4. Test Varying Transaction Activity and Transaction Intervals for 12 MB and 16 MB Server Configurations

Total Number of Events = 396
Total Time (seconds) = 707.12
Overall Response Time Average = 1.80

Script Name	Total Events	Total Time	Script Average Response Time
Check	158	296.93	1.94
Deposit	194	333.65	1.72
Payment	44	76.54	1.74

Figure 5. Sample Script Summary Report

Script Name	Total Task Events	Total Time	Task Average Response Time
Check	Host 86	147.68	1.72
Check	Total 72	149.25	2.07
Deposit	Host 97	170.68	1.76
Deposit	Total 97	162.97	1.68
Loan	Host 24	41.37	1.72
Loan	Total 20	35.17	1.76

Note: All times are in seconds.

Figure 6. Sample Task Summary Report

- **Active workstations:** The number of workstations that the system can simultaneously support while maintaining reasonable response times at a given transaction rate.

Example

SET uses scripts to measure specific functions. The following scripts reflect a teller's normal activities in a bank and illustrate how SET measures particular functions.

Defining the Workload

It is important to understand the characteristics of the test workload before it is created. Different application environments can have different workload characteristics. The transactions in the following workload simulate a bank teller's daily activities: cashing checks, making deposits, and applying customer payments. Each of these transactions defines a single script for the teller environment containing the following functions:

- Start transaction
- Accept data input
- Start host timer 1 (to measure the host processing time)
- Send input to the host
- Process data from the host
- Stop timer 1
- Start timer 2 (to measure the total processing time on the server)
- Request database update
- Request database update (for backup)
- Stop timer 2
- End transaction

Workload Mixture

The volume of each kind of test transaction is established before the test to represent a typical workload. Workstations can run the same types or different types of transactions. Figure 3 shows a sample report of a workload mixture in the bank teller environment. The host application accesses customer balances for verifying or updating. Each database request consists of four database reads followed by four database updates.

Measurement Test Cases

Transaction intervals vary the time between transactions. A *transaction interval* is the amount of time between two transactions. The interval is varied to simulate, and measure the effect of, differing work amounts on the server and on the entire network. To test performance with different system configurations, testing staff can adjust transaction intervals, the number of active workstations, and the amount of memory on the server. For accurate results, only one of these variables should be adjusted for each test cycle. If the testing staff wants to change the amount of memory in the server from 12 MB to 16 MB, all other factors should remain the same. Figure 4 shows examples of tests that can be run by varying the amount of server memory, transaction intervals, and the number of active workstations.

Applying test workloads on small numbers of workstations is suitable for measuring the best-case response time. As workstations are added, testing staff can use the test workloads to investigate maximum system capacity. Other tests can be made to identify points between the minimum and maximum loads and to evaluate the performance characteristics of a system under an increasing workload.

Reporting Capabilities

The SET log files contain information such as workstation execution scripts, the start time of the test, and the log messages used to time stamp a function. Each SET controller manages its own log files.

Reports can be generated by script, by task, and by function. Detailed reports can present information such as percentiles, standard deviation, and minimum and maximum response times. Figure 5 shows a sample script summary report for the bank teller example.

IBM's Solutions Validation Laboratory will assist customers in establishing a test lab using specialized tools.

In Figure 5, the Check script occurred 158 times during the testing period. The total elapsed time for this script was 296.93 seconds, and the average response time was 1.94 seconds (total time divided by total events).

Figure 6 shows a sample task summary report. The task summary report is the breakdown of the tasks within a script. The Check script has two tasks: Host, which accesses the host system, and Total, which accesses the OS/2 server for customer balances. The average response time for a task is its total elapsed time divided by the number of events for that task.

Additional Information

IBM's Solutions Validation Laboratory will assist customers in establishing a test lab using specialized tools or will provide testing and consultation at IBM's Roanoke, Texas facility. For more information, call (800) 742-2493.

***Jamie Hughey** is an advisory marketing support representative in the Multivendor Systems Center, part of the Solution Validation Services located in Southlake, Texas. She has worked with SET since 1990 and began the Solutions Validation Services for IBM customers. Jamie holds a BS in data processing from Kennesaw State College in Marietta, Georgia.*

***Alan Harrison** joined IBM in 1968 and is now an advisory programmer. His previous assignments include diagnostic programming in Boca Raton and Austin System Assurance. He currently provides customer support at System Performance/Test Tools in Austin, Texas.*

***Boyd Jasperson** is a staff programmer with System Performance/Test Tools in Austin. He joined IBM in 1968 with a BS in mathematics from Brigham Young University. His previous assignments include firmware development on IBM 360s/370s and communication development. Boyd is the lead programmer in development and support of SET.*

Q Where do users get information on AIX and the RISC System/6000?

A They go straight to the source, and so should you!

Subscribe to */AIXtra*

If you use AIX or have an interest in open systems, you could use a single, comprehensive source of information devoted to your needs. */AIXtra: IBM's Magazine For AIX Professionals* is that resource. Whether you're an experienced system administrator or you're just starting to use UNIX, there are many reasons to read */AIXtra*:

- Stay informed about the latest products for AIX and the RISC System/6000.
- Enjoy many well-written, educational articles about subjects like networking, communications, system administration, graphical user interfaces, distributed computing, and much more.
- Learn how others have harnessed the power of AIX and the RISC System/6000.
- Check the */AIXtra* Field Television Network (FTN) schedule for monthly satellite broadcasts just for AIX users.
- Get detailed information about exciting trade shows and expositions.

You're interested in UNIX because it's a powerful resource. You should subscribe to */AIXtra* for the same reason. Be sure you're getting the full benefits of AIX and the RISC System/6000.

There are many reasons to read and enjoy */AIXtra*, and there also are many ways to subscribe. Choose a subscription method below that is most convenient for you. Then either copy this page and fax it to (415) 948-4280, or call The TDA Group at (800) 551-2832 and request your subscription. Another option is to write to TDA at the address below. Please provide the following information.

❑ 1 Year Subscription (4 issues) for $35*
❑ 2 Year Subscription (8 issues) for $64*
❑ 1 Year Subscription (surface delivery) Canada/Mexico $60†; other countries $80†
  ❑ Charge to my VISA/Mastercard
  ❑ Bill me Purchase Order #: ____________
  ❑ Check enclosed

Name ______________________
Company ______________________
Address ______________________
City/State/Zip ______________________
Phone ______________________
VISA/Mastercard # ______________________ Expires ___/___

/AIXtra
The TDA Group
P.O. Box 1360
Los Altos, CA 94023

*California residents add applicable sales tax. †Canada/Mexico and other international subscriptions must be prepaid in U.S. dollars.

Little Solutions

We invite you to share your "little solutions" in this column. Send them to us in care of the editor.

RIPLing Extended Services on OS/2 2.0

OS/2 2.0 Remote Initial Program Load (RIPL) provides a way to run OS/2 and the Communications Manager on a medialess workstation. The software used here is OS/2 2.0, OS/2 LAN Server 2.0 – Entry, and OS/2 Extended Services 1.0. *Note*: If you are running OS/2 1.3 base and LAN Server 2.0 – Advanced with Extended Services 1.0, omit steps 5 and 6; in step 7, the GETRPL utility will prompt you for the 1.3 base diskettes.

Follow this procedure to install RIPL:

1. Install OS/2 2.0 and OS/2 Extended Services on a workstation that you intend to use as a server.
2. Install LAN Server 2.0. During installation select ADVANCED as the type of configuration to run. At the install/remove window, select OS/2 RIPL Service.
3. Follow the remaining selections and prompts. In most cases, select the defaults.
4. After successful installation of the server, shut down and reboot the system.
5. When the system comes back up, do not start the server. At an OS/2 command prompt, run the RIPLINST.EXE utility. RIPLINST.EXE is found on disk 7 of the base code and can be unpacked by using the OS/2 2.0 UNPACK command. To unpack RIPLINST, place disk 7 of the base code in drive A: and type UNPACK A:\RIPLINST.
6. Be sure to install the OS/2 code on the drive where LAN Server is installed. For example, if LAN Server is installed on drive D:, make the "OS/2 Remote IPL Directory" D:\IBMLAN\RPL\OS2.20. Also, specify the source directory as drive A: and get ready for the diskette shuffle.
7. When RIPLINST has completed, start the server and logon as an administrator. If the Remoteboot service is started, stop it and run the GETRPL utility from an OS/2 command prompt.
8. Make a CMLIB directory under \IBMLAN\RPL. If Communications Manager is running on your server, stop it before continuing.
9. Use the following XCOPY command, where X is the drive on which Communications Manager is installed:

```
XCOPY X:\CMLIB /S /E /V IBMLAN\RPL\CMLIB
```

10. Edit the file \IBMLAN\RPL\CMLIB\SETCMTD.CMD and change the drive statements to C:. For example, if Communications Manager is installed on drive D:, the file would look as follows:

```
SET CMTD=D:
%CMTD%
CD \CMLIB
SET CMLT=D:\IBMCOM
```

    Change it to:

```
SET CMTD=C:
%CMTD%
CD \CMLIB
SET CMLT=C:\IBMCOM
```

11. Create a dummy RIPL machine to be used as a model when creating new Extended Services RIPL machines. To do this, logon as an administrator and go into the LAN Requester Full-Screen Interface. Select each of the following menu entries: Definitions, Machine Parameters, New, Create, Remote IPL workstation. Fill in the fields as follows:

    Machine ID: ESMODEL
    Description: Extended Services RIPL machine model
    Network adapter number: 10005AFFFFFC (or fill in your own number)

Remote IPL server: Your server
Server record identifier: Choose the appropriate OS/2 2.0 record for your network
File index table to model: FITS\DEFALT20

12. Create the following directories under `\IBMLAN\RPLUSER\ESMODEL`:

```
\CMLIB
\CMLIB\APPN
```

13. Copy `\CMLIB\*.DAT` to `\IBMLAN\RPLUSER\ESMODEL\CMLIB`

    Copy `\CMLIB\ACS.PRO` to `IBMLAN\RPLUSER\ESMODEL\CMLIB`

14. Copy `\OS2\EPW.EXE` to `\IBMLAN\RPL\OS2.20\OS2`

    Copy `\OS2\DLL\EPWNL001.DLL` to `\IBMLAN\RPL\OS2.20\OS2\DLL`

    Copy `\OS2\HELP\EPWNL001.HLP` to `\IBMLAN\RPL\OS2.20\OS2\HELP`

    Copy `\OS2\EPW.INI` to `\IBMLAN\RPLUSER\ESMODEL\OS2`

15. Give RPLGROUP the proper access controls to the new subdirectories that you have created. To do this, go into the LAN Requester Full-Screen Interface and select the following menu entries: Definitions, Access Control, Servers, Display Profiles By Server, and Select DC. Search until you find the `\IBMLAN\RPL` entry and "Apply" this access control.

16. Create the definitions for the machines that are going to RIPL Extended Services using the modeling feature with ESMODEL as the model. To do this, go into Definitions, Machine parameters and select New, Actions, and Create. Fill in the fields with the appropriate information. When you get to "File index table to model," press F4 and select ESMODEL.

17. Run ESCFG on the domain controller for each RIPL machine to create a configuration file for them. Select the option Create for Another Workstation, and then create the configuration file just as you would for a regular Communications Manager machine, specifying which options you would like, such as 3270 or 5250 emulation. Also, be sure to specify the use of a UAA. For example, suppose one of the machines that you created is called RPLES and you named the `CFG` file `RPLESCFG`.

    Note: Throughout the rest of these instructions, the example mentioned above is used for directory and file names. Any occurrence of RPLES will need to be replaced with the name of your RIPL workstation. Also, occurrences of `RPLESCFG` should be replaced by the name of the configuration file that you created.

18. Copy `\OS2\INSTALL\RPLESCFG.CFG` to `\IBMLAN\RPLUSER\RPLES\CMLIB`

    If the RIPL machine is called FRED, and the configuration file that you created is called `FREDCFG`, the above line should read:

    Copy `\OS2\INSTALL\FREDCFG.CFG` to `\IBMLAN\RPLUSER\FRED\CMLIB`

    Copy `\OS2\INSTALL\APPN\RPLESCFG.*` to `\IBMLAN\RPLUSER\RPLES\CMLIB\APPN`

    Copy `\IBMCOM\RPLESCFG.INI` to `\IBMLAN\RPLUSER\RPLES\IBMCOM`

19. Edit the file `\IBMLAN\RPL\FITS\RPLES.FIT` and change the line under the CMLIB section that reads:

```
C:\IBMCOM\cfgname.ini
   \\servername\WRKFILES\RPLES\IBMCOM\
   cfgname.ini
```

    to:

```
C:\IBMCOM\RPLESCFG.INI
   \\servername\WRKFILES\RPLES\IBMCOM\
   RPLESCFG.INI
```

20. Edit the file `\IBMLAN\RPL\MACHINES\RPLES\CONFIG.20`

    Add `C:\CMLIB\DLL` to the `LIBPATH` statement.

    Add `C:\CMLIB` and `C:\CMLIB\APPN` to the `SET PATH` and `DPATH` statements.

    Add `C:\CMLIB\APPN` to the `SET HELP` statement.

    Add the following lines after the statement `DEVICE=C:\IBMCOM\PROTOCOL\NETBIOS.OS2`:

```
DEVICE=C:\CMLIB\ACSLDLAN.SYS
RUN=C:\OS2\EPW.EXE
RUN=C:\IBMCOM\PROTOCOL\NETBIND.EXE
```

    Add any adapter-specific Communications Manager device drivers needed by the workstations after the above statements. For example, add the following for 3270 support via LAN:

```
DEVICE=C:\CMLIB\APPN\CMKFMDE.SYS
```

The following is an example of a RIPL machine on token ring that is running 3270 emulation.

```
PROTSHELL=C:\OS2\PMSHELL.EXE

DEVICE=C:\IBMCOM\LANMSGDD.OS2 /I:C:\IBMCOM
DEVICE=C:\IBMCOM\PROTMAN.OS2  /I:C:\IBMCOM
DEVICE=C:\IBMCOM\MACS\IBMTOK.OS2
DEVICE=C:\IBMCOM\PROTOCOL\LANDD.OS2
DEVICE=C:\IBMCOM\PROTOCOL\NETBEUI.OS2

DEVICE=C:\IBMCOM\PROTOCOL\LANDLLDD.OS2
RUN=C:\IBMCOM\PROTOCOL\LANDLL.EXE
RUN=C:\IBMCOM\LANMSGEX.EXE

DEVICE=C:\IBMLAN\NETPROG\RDRHELP.200
IFS=C:\IBMLAN\NETPROG\NETWKSTA.200
   /I:C:\IBMLAN
IFS=C:\OS2\HPFS.IFS  /CACHE:512 /CRECL:4
   /AUTOCHECK:D

DEVICE=C:\IBMCOM\PROTOCOL\NETBIOS.OS2
DEVICE=C:\CMLIB\ACSLDLAN.SYS
RUN=C:\OS2\EPW.EXE
RUN=C:\IBMCOM\PROTOCOL\NETBIND.EXE
DEVICE=C:\CMLIB\APPN\CMKFMDE.SYS

SET USER_INI=C:\OS2\OS2.INI
SET SYSTEM_INI=C:\OS2\OS2SYS.INI
.
.
.
```

21. Start the Remoteboot service on the server and turn on the RIPL machine. After the RIPL machine is finished booting, go into an OS/2 full screen, change to the `CMLIB` subdirectory, and type `STARTCM`. Communications Manager will begin to come up, and will prompt you for a configuration file name. Type in the one that you created for this machine, which in our example was `RPLESCFG`. Communications Manager should then come up. Congratulations, you have now RIPLed Communications Manager!

22. After Communications Manager is started, you will find the Communications Manager icon in the Minimized Window Viewer. If you do not like having to start Communications Manager from the command line, you can create an icon and just place `C:\CMLIB\STARTCM.CMD` in the "Path and Filename" field.

—Charles Bueche, Dallas, Texas

Update to "Remote Installation of OS/2" (*Personal Systems Technical Solutions*, April 1992)

When the article "Remote Installation of OS/2" was written, OS/2 2.0 was still under development. Please note the following change:

The name of the printer device driver directories changed from `disk_Pn` to `pmdd_n`, where `n` is the number of the printer device driver disk.

For those interested in more detailed information concerning remote installation of OS/2, you may order the book *OS/2 Version 2.0 Remote Installation and Maintenance* (GG24-3780) from your IBM representative or the IBM branch office serving your locality. This publication contains all aspects of the subject, including different communications strategies.

New Products

IBM PS/2 Model N51 SX System

The PS/2 Model N51 SX is a small, lightweight, battery/AC-adapter powered portable notebook system. It is designed for the mobile professional who requires a high-function portable, convenient to use and carry. The PS/2 Model N51 SX has the speed and capacity to support advanced operating systems and applications. Standard features include:

- 16 MHz 80386SX processor
- 2 MB of 80 ns memory (expandable to 10 MB)
- 40 MB fixed disk
- 3.5-inch 1.44 MB diskette drive
- VGA resolution Liquid Crystal Display (LCD)
- 84-key keyboard
- Two NiCd batteries and AC adapter (20 W)
- Serial and parallel ports
- Keypad/mouse port
- VGA monitor port
- External expansion I/O ports for attaching external devices
- Compatible with the Micro Channel architecture

Options announced for the Model N51 SX are:

- Internal data/fax modem for U.S./Canada/Japan
- Serial adapter
- Communication cartridge with AC adapter
- 2, 4, or 8 MB memory upgrades
- Quick charger
- Car battery adapter
- Numeric keypad (17-key)
- Miniature mouse
- NiCd and NiMH battery packs
- AC adapter (20 W)
- External rechargeable power pack

The IBM PS/2 Carrying Case is available as an accessory.

Letter # 192-065, March 24, 1992

IBM PS/2 Model N51 SLC Notebook System

The PS/2 Model N51 SLC is a small, lightweight, battery/AC-adapter powered portable notebook system. The PS/2 Model N51 SLC is designed for the mobile professional who requires a high-function portable, convenient to use and carry. The PS/2 Model N51 SLC has the speed and capacity to support advanced operating systems and applications. Standard features include:

- 16 MHz 386 SLC processor with 8 KB on-chip cache
- 2 MB of 80 ns memory (expandable to 10 MB)
- 80 MB fixed disk
- 3.5-inch 1.44 MB diskette drive
- VGA resolution LCD
- 84-key keyboard
- Two NiMH batteries and AC adapter (20 W)
- Serial and parallel ports
- Keypad/mouse port
- VGA monitor port
- External expansion I/O ports for attaching external devices
- Compatible with Micro Channel architecture

Options announced for the Model N51 SLC are:

- Internal data/fax modem for U.S./Canada/Japan
- Serial adapter
- Communication cartridge with AC adapter
- 2, 4, or 8 MB memory upgrades
- Quick charger
- Car battery adapter
- Numeric keypad (17-key)
- Miniature mouse
- NiCd and NiMH battery packs
- AC adapter (20 W)
- External rechargeable power pack

Letter # 192-072, March 24, 1992

IBM PS/2 Model CL57 SX Laptop System

The PS/2 Model CL57 SX is a lightweight, battery/AC-adapter powered laptop portable color system. The PS/2 Model CL57 SX is designed for the mobile professional who requires a high-function color portable, convenient to use and carry, with the speed and capacity to support advanced applications and operating systems.

Standard features include:

- 20 MHz 80386SX processor
- 2 MB of 80 ns memory (expandable to 16 MB)
- 80 MB fixed disk
- 3.5-inch 1.44 MB diskette drive
- VGA resolution using an active matrix, Thin Film Transistor (TFT) color LCD

- 84-key keyboard with integrated trackball pointing device
- Rechargeable NiMH battery pack
- Serial and parallel ports
- Keypad/mouse port
- VGA monitor port
- External expansion I/O ports for attaching external devices
- Compatible with Micro Channel architecture

Options announced for the Model CL57 SX are:

- PS/2 internal data/fax modem for the U.S. and Canada
- PS/2 serial adapter
- PS/2 communications cartridge II
- PS/2 2, 4, or 8 MB IC DRAM cards
- PS/2 external rechargeable power pack II
- PS/2 numeric keypad (17-key)
- PS/2 miniature mouse
- PS/2 NiMH battery pack II
- PS/2 AC adapter (60 W)

The IBM PS/2 Deluxe Carrying Case is available as an accessory.

Letter # 192-073, March 24, 1992

IBM PS/2 Model 95 XP 486

The PS/2 Model 95 XP 486 family of systems is enhanced with new, high-performance server function and increased data storage capacity through the introduction of two new model offerings. The PS/2 Model 95 XP 486 is based on Micro Channel architecture and contains Intel's i80486 50 MHz microprocessor with integrated numeric coprocessor unit. These models also use several technological leadership functions specifically designed to enhance overall LAN server function, reliability, and performance. These functions include 40 MB/second streaming data; support of Subsystem Control Blocks (SCB); Error-Checking and Correction (ECC) memory; enhanced memory optimization; improved Reliability, Availability, and Serviceability (RAS); and Vital Product Data (VPD).

A 256 KB Level 2 cache, 16 MB of ECC memory, XGA™ graphics with 512 KB video memory, and a 2.88 MB diskette drive are also provided as standard features. Both new models support a wide array of IBM keyboards.

The PS/2 Model 95 XP 486 contains a high-performance 400 MB Small Computer Systems Interface (SCSI) fixed disk drive as the standard fixed disk. The PS/2 Model 95 XP 486 offers even greater high-performance, data storage capacity through the introduction of the 1 GB SCSI hard disk drive as the standard fixed disk. With the installation of an additional PS/2 1 GB SCSI hard disk drive option, and three PS/2 400 MB SCSI fixed disk drive options, the Model 8595-0MT can achieve up to 3.2 GB of internal data storage to meet the demanding data storage needs generally associated with high-performance, full-function, LAN server applications.

The IBM PS/2 4 MB 70 ns ECC memory module and the IBM PS/2 8 MB 70 ns ECC memory module options provide system board memory expansion for the new Model 8595-0MF and 0MT systems, and for PS/2 Model 90 XP and 95 XP systems that have been upgraded with the PS/2 enhanced 486/50 processor upgrade option. These new memory options, with the ECC memory support provided on the 486/50 processor complex, enhance memory reliability and system availability by correcting memory failures without system interruption.

Both models (0MF and 0MT) support a Japanese language keyboard, the IBM Enhanced Keyboard-Japanese. With IBM DOS J5.02/V, a U.S. user can run Japanese application programs. The Japanese keyboard is selected at the time of system purchase.

Highlights:

- Intel i80486 50 MHz processor with standard 256 KB Level 2 cache
- 16 MB standard ECC memory, expandable to 64 MB on the system board
- 40 MB/second streaming data
- Enhanced memory optimization
- Subsystem control blocks
- 1 GB or 400 MB fixed disk models – up to 3.2 GB of disk storage
- Improved RAS – error log, synchronous channel check data parity, and vital product data identifier
- XGA Display Adapter/A with 512 KB video memory providing 1024 x 768 resolution and 16 colors
- 2.88 MB diskette drive

Letter # 192-096, April 28, 1992

IBM PS/2 Model 95 XP 486

IBM brings higher levels of system performance to the popular PS/2 Model 95 XP 486 family of products by incorporating Intel's i486DX2-50 50 MHz 32-bit microprocessor into an attractively priced PS/2 Model 95 XP 486 system. The PS/2 Model 95 is based on Micro Channel architecture and contains 8 MB of 70 ns memory, XGA graphics, and a 2.88 MB diskette drive. This model supports an array of IBM keyboards.

The i486DX2-50 microprocessor installed on the upgradeable processor complex card of this system operates at an internal clock speed of 50 MHz and an external speed of 25 MHz. Included are an internal memory cache controller, 8 KB memory cache, and an integrated floating-point processor unit.

This system uses a 32-bit, busmaster IBM PS/2 SCSI adapter with cache to interface with a high-performance SCSI fixed disk drive. The PS/2 Model 95 XP 486 contains a 400 MB fixed disk. The PS/2 Model 95 XP 486, with the addition of two PS/2 1 GB SCSI hard disk drive options and two additional PS/2 400 MB SCSI fixed disk drive options, can provide up to 3.2 GB of internal data storage. All other expandable attributes of the PS/2 Model 95 XP 486 platform remain unchanged.

Highlights:

- New processor complex featuring the i486DX2-50 50 MHz microprocessor
- 8 MB standard parity memory, expandable to 64 MB on the system board
- PS/2 SCSI 32-bit busmaster adapter with cache
- Up to 3.2 GB of internal high-speed data storage
- Eight 32-bit Micro Channel expansion slots (one slot is used for the PS/2 Micro Channel SCSI adapter with cache, and one is used for the PS/2 XGA Display Adapter/A)
- Seven internal storage device bays supporting a combination of 3.5-inch half-height drives and 5.25-inch full-height drives
- 2.88 MB diskette drive
- XGA Display Adapter/A with 512 KB video memory (1024 x 768 resolution with 16 colors)
- DMA serial port and DMA parallel port
- Selectable boot and easy-to-upgrade licensed system programs

Letter # 192-097, April 28, 1992

IBM PS/2 Model 90 XP 486

IBM brings higher levels of system performance to the popular PS/2 Model 90 XP 486 family of products by incorporating Intel's i486DX2-50 50 MHz 32-bit microprocessor into two attractively priced PS/2 Model 90 systems. These new models of the PS/2 Model 90 XP 486, based on Micro Channel architecture, contain 8 MB of 70 ns memory, XGA graphics, and a 2.88 MB diskette drive. The models also support a wide array of IBM keyboards.

The i486DX2-50 microprocessor, installed on the upgradeable processor complex card of these systems, operates at an internal clock speed of 50 MHz and an external speed of 25 MHz. Included are an internal memory cache controller, 8 KB memory cache, and an integrated floating-point processor unit.

Both systems use a 32-bit, busmaster IBM PS/2 SCSI adapter with cache to interface with high-performance SCSI fixed disk drives. The PS/2 Model 8590-0L9 contains a 160 MB fixed disk, while the Model 8590-0LF contains a 400 MB fixed disk. The PS/2 Model 8590-0LF with the addition of two PS/2 400 MB SCSI Fixed Disk Drive options can provide up to 1.2 GB of internal data storage. All other expandable attributes of the PS/2 Model 90 XP 486 platform remain unchanged.

IBM PS/2 2.88 MB Diskette Drive: This new 3.5-inch, 1-inch high diskette drive option features media sense capability for the standard 3.5-inch diskette capacities of 720 KB, 1.44 MB, and 2.88 MB. The diskette drive option can read and write data up to a formatted capacity of 2.88 MB, while maintaining read and write capability with 720 KB and 1.44 MB diskette drives. The PS/2 2.88 MB diskette drive option can be installed as a second diskette drive in PS/2 Models 90 XP 486 and 95 XP 486 that have the drive as a standard feature.

Highlights:

- New processor complex featuring the i486DX2-50 – 50 MHz microprocessor
- 8 MB standard parity memory, expandable to 64 MB on the system board
- Enhanced performance XGA graphics integrated on the system board providing 1024 x 768 resolution with 16 colors
- Up to 1.2 GB of internal high-speed data storage
- PS/2 SCSI 32-bit busmaster adapter with cache
- Four internal storage device "bays" supporting three 3.5-inch half-height drives and one 5.25-inch half-height drive
- Four 32-bit Micro Channel expansion slots (one used for the SCSI adapter)
- Two DMA serial ports and one DMA parallel port
- Selectable boot and easy-to-upgrade licensed system programs

Letter # 192-098, April 28, 1992

IBM PS/2 486DX2-50 Processor Upgrade Option

The PS/2 486DX2-50 Processor Upgrade Option is designed to enhance performance for PS/2 Model 90 XP 486 and 95 XP 486 systems. This announcement demonstrates IBM's commitment to bring high-quality technological leadership products to the PS/2 market quickly and provide investment protection to those who have previously purchased

PS/2 Model 90 XP 486 and 95 XP 486 systems.

The PS/2 486DX2-50 Processor Upgrade Option features the 32-bit Intel 486DX2-50 processor installed on a processor complex. The processor performs computational operations internally at a speed of 50 MHz, while operating external to the chip at 25 MHz. The processor includes a numeric coprocessor, an internal cache controller, and 8 KB of cache memory. The PS/2 486DX2-50 Processor Upgrade Option can be used to upgrade the PS/2 Model 90 XP 486 (8590-0G5, 0G9, 0J5, 0J9, 0K9, 0KD and 0KF) and PS/2 Model 95 XP 486 (8595-0G9, 0GF, 0J9, 0JD, 0JF, 0KD and 0KF) system units as follows:

- PS/2 486DX2-50 Processor Upgrade Option – from 486 SX/20 or 487 SX/20
- PS/2 486DX2-50 Processor Upgrade Option – from 486/25
- PS/2 486DX2-50 Processor Upgrade Option – from 486/33

Highlights:

- Dual-speed, 32-bit microprocessor
- Internal memory cache controller and 8 KB internal memory cache
- Improved I/O performance
- Internal floating-point processor unit

Letter # 192-099, April 28, 1992

IBM PS/2 486-25/50 Microprocessor Upgrade Option

IBM introduces a new microprocessor upgrade option, designed for installation in the second processor socket of the PS/2 Model 90 XP 486 (8590-0H5 and 0H9) and 95 XP 486 (8595-0H9 and 0HF) system units. The upgrade, consisting of the new Intel ODP486SX-25 microprocessor and a new Reference Diskette, provides higher levels of processor performance at a very attractive price for these selected PS/2 systems. The PS/2 486-25/50 Microprocessor Upgrade Option features an internal processing speed of 50 MHz and an external speed of 25 MHz. It includes an internal memory cache controller, 8 KB cache memory, and integrated floating-point processor unit.

Highlights:

- Dual-speed, 32-bit microprocessor
- Internal memory cache controller and 8 KB internal memory cache
- Internal floating-point processor unit

Letter # 192-100, April 28, 1992

IBM PS/2 Enhanced 486/50 Processor Upgrade Option

The PS/2 Model 90 XP 486 (8590) and Model 95 XP 486 (8595) family of systems can be upgraded by the new PS/2 Enhanced 486 50 MHz Processor Upgrade Option. This option allows current users of the 8590 and 8595 to obtain the higher performance and improved Reliability, Availability, and Serviceability (RAS) afforded by Error Checking and Correction (ECC) memory. In addition, it implements the 40 MB/second streaming data introduced on Models 8595-0MF and -0MT, providing significant high-speed data-transfer performance improvements when using adapters with streaming data support.

Better system performance and more efficient memory utilization are afforded through enhanced memory optimization between memory and the Micro Channel. The Intel i80486 50 MHz microprocessor includes an internal memory cache controller, 8 KB memory cache, and an integrated floating-point processor unit. Additionally, an external 256 KB Level 2 cache is provided on the processor complex as a standard feature. Memory must be either all ECC or all parity, and must be installed in matched pairs of the same size, type, and speed.

Highlights:

- 486 50 MHz 32-bit microprocessor with internal memory cache controller and 8 KB internal memory cache
- Enhanced memory optimization and 40 MB/second streaming data
- Full 32-bit DMA addressability
- Standard 256 KB Level 2 cache
- ECC or parity memory
- Internal floating-point processor unit

Letter # 192-101, April 28, 1992

IBM PS/2 1 GB SCSI Hard Disk Drive

The IBM family of SCSI hard disk drives has been enhanced with the addition of the PS/2 1 GB SCSI hard disk drive. With a 1 GB storage capacity, an 11.0 ms average seek time and a 256 KB look-ahead buffer, this drive offers the highest storage capacity and performance available today from IBM for the SCSI-based PS/2 Micro Channel systems. The 1 GB drive can be installed internally in the PS/2 8595, or attached externally to any SCSI-based PS/2 Micro Channel system when installed in the PS/2 SCSI Storage Enclosure or the PS/2 External Enclosure for SCSI devices.

This drive also offers the highest performance available today from IBM for the PS/55 Model 5580.

Highlights:

- Recommended solution for LAN servers or technical workstations because of its ability to quickly store and retrieve large amounts of data

- Can be installed in the PS/2 8595, 3510-0V0 and the 3511-003 (PS/2 SCSI enclosures), when attached to any SCSI-based PS/2 Micro Channel system; can also be installed in the PS/55 Model 5580
- Fast data transfer (11.0 ms average seek time) and 256 KB look-ahead buffer enhance user productivity

Letter # 192-102, April 28, 1992

Selected IBM PS/2 Models 56 and 57 Preconfigured with OS/2 2.0

IBM enhances selected models of the PS/2 Models 56 SX (045), 56 SLC (055, 059), 57 SX (045, 049) and 57 SLC (055, 059) by preinstalling OS/2 2.0 with new usability features, and including the IBM mouse as a standard feature. Users can reduce system installation time and substantially increase productivity by having immediate access to this new, multitasking desktop operating system, which protects current investment in existing applications and supports 32-bit software applications. By preinstalling OS/2 2.0 on these powerful desktop systems, IBM matches a high-performance, functionally comprehensive operating system with a hardware platform – ready when powered on the first time – to provide solutions for a broad range of user requirements.

Letter # 192-120, May 5, 1992

IBM Laserprinter 6A, 6P, 10A, and 10P

Four new models have been added to the IBM LaserPrinter 4029 Series. All are equipped with an Adobe PostScript interpreter, Print Quality Enhancement Technology (PQET), and additional memory. Models 021, 041, and 042 have 39 Type 1 formatted outline fonts, and Model 022 has 17 fonts. Models 021 and 022 have addressable resolution of 300 x 300 dpi with upgradeability to 600 x 600 dpi. Models 041 and 042 have 600 x 600 dpi resolution as standard. The LaserPrinter A Series, Models 021 and 041, are compatible with Apple and Macintosh systems and/or AppleTalk network connectivity. The LaserPrinter P Series, Models 022 and 042, support the IBM Personal Printer Data Stream (PPDS), Hewlett-Packard® LaserJet® Series II (PCL4) emulation, and plotter emulation.

Highlights:

All IBM LaserPrinter 4029 Series printers share the following features:

- A standard Adobe PostScript interpreter
- Print Quality Enhancement Technology (PQET)
- High-resolution, 300 x 300 dpi printing
- High-resolution, 600 x 600 dpi PostScript printing capability (standard on 10A and 10P models; 4 MB additional printer memory required on 6A and 6P)
- An easy-to-use 16-character LCD operator panel
- A full range of automatic paper-handling trays (with automatic paper-size sensing)
- Memory expandability up to 9 MB
- A small operating footprint

The following features are also provided on the LaserPrinter models specified:

- 39 Type 1 outline fonts (6A, 10A, and 10P)
- 17 Type 1 outline fonts (6P)
- Enhanced performance with an AppleTalk network interface (6A and 10A)
- A controller card with a 10 MHz Motorola™ MC68000 processor (6A and 6P)
- A high-performance controller card with a 16.7 MHz Motorola MC68020 processor (10A and 10P)
- 2 MB of RAM (6A and 6P)
- 5 MB of RAM (10A and 10P)
- Enhanced envelope hardware for reduced wrinkling of envelopes (10A and 10P)
- A 250-sheet output bin with tray-full sensing (10A and 10P)
- Standard high-speed parallel and serial interfaces (6P and 10P)
- Standard software-switchable emulations, between Adobe PostScript, IBM PPDS, HP® LaserJet Series II (PCL4) emulation, and plotter emulation (IBM 7372 and HP 7475A color plotters) (6P and 10P)
- Optional HP LaserJet Series III (PCL5) emulation (6P and 10P)

Letter # 192-078, March 31, 1992

Filters to Support 16 or 4 Mbits/second on Unshielded Twisted Pair Media

Five new filters will provide users more cabling options to meet their networking needs, including Shielded Twisted Pair (STP) and Unshielded Twisted Pair (UTP) media.

- The IBM 16/4 8230 Unshielded Media Filter is a feature of the 8230 Controlled Access Unit (CAU).
- The IBM 16/4 Workstation Filter is an accessory and is required on any device with a 16/4 Token-Ring Network Adapter installed that is operating at 16 or 4 Mbits/second on UTP media lobes. This filter assembly includes a three-meter length of UTP cable.
- The IBM 16/4 Repeater Unshielded Media Filter is an accessory and can be attached to any IBM repeater.

- — One of these filters will be required between an 8230 or a repeater and an IBM 8228 Token-Ring Network Multistation Access Unit that has UTP lobes.
- — If a segment of a ring contains only STP lobes, that segment can be isolated from the UTP portion by placing a 16/4 Repeater Unshielded Media Filter on each side of the segment. STP lobes in this isolated segment do not require lobe filtering.

See the *IBM Token-Ring Network Supplement for 16/4 Mbits/second Operation on Unshielded Twisted Pair Media* (GD21-0048) for clarification of filtering requirements on mixed STP/UTP rings. After June 26, 1992, the supplement will no longer be available, but the information will be available in the *IBM Token-Ring Network Introduction and Planning Guide*.

- The IBM 16/4 Lobe Filter A is required on all lobes of any 8228 or data-grade Lobe Attachment Module (LAM) of an 8230 operating at 16 or 4 Mbits/second on UTP media. This filter is connected directly to the concentrator and includes a six-meter length of UTP cable, which is unterminated on the other end.
- The IBM 16/4 Lobe Filter B is required on all lobes of the current RJ-45 LAM and the new 16/4 Unshielded Media Lobe Attachment Module of the 8230 operating at 4 or 16 Mbits/second. This filter is connected directly to the concentrator and includes a six-meter length of UTP cable, which is unterminated on the other end.

The above filters will allow workstations attached via the existing 8230 RJ-45 LAM or the 8228 to operate on unshielded twisted pair media (category 4 or 5 as defined in EIA/TIA TSB-36, November 1991).

IBM will continue to test different wire types and will inform users of the results of these tests as they are completed.

These filters also will be required when 16 or 4 Mbits/second operation on UTP media is being used in conjunction with the new IBM 8230 Model 002 Controlled Access Unit and 16/4 Unshielded Media Lobe Attachment Module. For detailed cabling information, consult the *IBM Token-Ring Network Supplement for 16/4 Mbits/second Operation on Unshielded Twisted-Pair Media.*

Letter # 192-082, April 21, 1992

IBM 8230 Model 002 and IBM Token-Ring Network 16/4 Unshielded Media Lobe Attachment Module

The new 8230 Model 002 and the new IBM Token-Ring Network 16/4 Unshielded Media Lobe Attachment Module (LAM), an active LAM, are used to connect adapters operating at 16 or 4 Mbits/second on unshielded twisted pair media to the main ring via the 8230. This active combination provides users with more cabling options to meet their networking needs, including Shielded Twisted Pair (STP) and Unshielded Twisted Pair (UTP) media.

The 8230 Model 002 and the 16/4 Unshielded Media LAM are used in conjunction with the 16/4 Mbits/second UTP Media Filters.

Letter # 192-083, April 21, 1992

LAN Enabler 2.0, OS/2 LAN Server 2.0 Softcopy Library

The IBM LAN Enabler 2.0 offers the OS/2 Requester, LAN Support Program, and DOS LAN Requester (DLR) – identical in function to the requesters included with the IBM OS/2 LAN Server 2.0 – in a separate licensed program. Connectivity to 80286 and 80386 workstations is enabled, including OS/2 LAN Server 2.0, Microsoft LAN Manager 2.0, and other compatible servers. This allows users of IBM OS/2 2.0 to access NetBios, NDIS-based and 802.2-based subsystems at an economical price.

IBM OS/2 LAN Server 2.0 Softcopy Library provides the publications for OS/2 LAN Server 2.0 on 3.5-inch media. These publications are displayed using BookMaster® READ products.

Highlights:

- Enables OS/2 workstations to run NetBios, NDIS, and 802.2 applications without LAN Server 2.0 or Extended Services/2
- Provides connectivity to 80286 and 80386 workstations, including LAN Server 2.0, Microsoft LAN Manager 2.0, and other compatible servers
- OS/2 LAN Server 2.0 Softcopy Library is now available in diskette form, for use with BookManager™ READ products.

Letter # 292-248, April 28, 1992

NetWare Network Computing Products from IBM

NetWare for SAA from IBM V1.2 (replacing V1.1, NetWare 3270 LAN Workstation for Windows V1.0) and NetWare for Macintosh from IBM V3.01 (200 session) are added to IBM's distribution, licensing, and support relationship with Novell, Inc.

Highlights:

- NetWare for SAA from IBM V1.2 adds NetView RUNCMD, collection point and end point support, enhanced AS/400 interoperability, and enhanced administration utilities.

- NetWare 3270 LAN Workstation for Windows from IBM V1.0 provides Windows 3.0 users with cost-effective access to SNA mainframes and AS/400 minicomputers via 3270 emulation.
- NetWare for Macintosh from IBM V3.01 is now available in a 200-user configuration.

Letter # 292-254, May 5, 1992

IBM Multimedia Presentation Manager/2 and IBM Multimedia Presentation Manager Toolkit/2

IBM Multimedia Presentation Manager/2 provides multimedia extensions to the OS/2 32-bit environment to enhance the ability of personal computers to run applications that combine sound and images.

The IBM Multimedia Presentation Manager Toolkit/2 contains C language bindings, sample programs, and documentation to assist the multimedia application developer.

Highlights:

Multimedia Presentation Manager/2

- Enhanced quality of information and communication
- New functions, devices, and multimedia data easily accommodated
- Open, extendable architecture, data standards, and consistent user interface
- Increased value of information and decreased user training costs
- Easy-to-use system setup

Multimedia Presentation Manager Toolkit/2

- Business solutions adaptable from sample programs
- Assists application developers in the use of Multimedia Presentation Manager/2

Letter # 292-192, March 31, 1992

IBM DOS 5.00 New OEM Service and Support

The service and support of IBM DOS 5.00 is expanded to include service and support for IBM-compatible personal computers ("OEM" and "IBM-compatible" are used interchangeably). A list called DOS5-OEM contains IBM-compatible personal computers tested with IBM DOS 5.00. This list is provided to worldwide IBM marketing and service channels and will be updated on a quarterly basis. Any licensee of IBM DOS 5.00 (past or current) is entitled to the list of service and support changes. IBM DOS 5.00 is refreshed in the United States with a new modification level number, 5.00.1. IBM DOS 5.00.1 has the following changes:

- A new dustcover, stating that DOS 5.00 is supported on IBM and IBM-compatible systems; and the service date extended to September 30, 1993
- Refreshed code that includes the latest CSD fixes, available through IBM Central Service, including the QBasic that works on all IBM and IBM-compatible hardware

In addition, the Upgrade IBM DOS 5.00.1 includes:

- A new SETUP module that installs over all IBM and IBM-compatible DOS 2.1, and higher, and installs across a LAN
- A new "Getting Started Upgrade" booklet that explains the new features of the SETUP module

IBM DOS 5.00.1 has the same part numbers as IBM DOS 5.00. The IBM DOS 5.00 U.K. version will be updated. The other European National Language Versions of IBM DOS 5.00 do not require any updates to allow execution and support on IBM and IBM-compatible PCs.

Letter # 292-239, April 28, 1992

Index to Past Issues of IBM Personal Systems Technical Solutions

April 1992 (G325-5015)

Comparing Architectures: Micro Channel and EISA
Portable Computer Trends and Directions
LCD Panel Technology
The OS/2 Workplace Shell
New Applications in OS/2 2.0
Unattended Installation of OS/2 2.0
OS/2 Communications Manager Trace Events
IBM and Novell LAN Software Coexistence
IBM 8209 LAN Bridge Connects Ethernet Clients to Novell and IBM Servers
Backup and Restore in an IBM NetWare Environment
The DOS Protected-Mode Environment
DOS Disk Management
Customizing Alphanumeric Screen Dimensions

January 1992 (G325-5014)

Additions to the IBM PS/1 Family
IBM LaserPrinter 4029 Series Print Quality Enhancements
OS/2 2.0: The Integrating Platform
Multiple Virtual DOS Machines
IBM OS/2 LAN Server 2.0
OS/2 2.0 Memory Management
Coding for Performance Under OS/2 Version 2.0
Extending the Functions of OS/2 REXX
Protecting User Exits Under OS/2 1.X
GDDM-OS/2 Link
IBM Upgrade Enhanced Install Utility/DOS 5.0
Advanced Peer-to-Peer Networking: An Overview
Using IBM SAA Networking Services/2
The AAI Family of Products
Securing the Enterprise Workstation

Issue 4, 1991 (G325-5013)

Power Factor: Non-Linear Loads and the Power Distribution System
Database Manager: Highlights and Direction
OS/2 Communications Manager
Improving OS/2 Application Performance
Creating PM Windows with Dialog Templates
REXX Program for OS/2 LAN Server
Micro Focus COBOL/2 and the DOS Database Requester
IBM DOS 5.0 Facts and Features
IBM DOS 5.0 Upgrade
DOS 5.0 Performance Improvements
DOS Memory Management Facilities
Disk Caching Under DOS
NetWare Client-Server Interaction
LANACS Protocols

Issue 3, 1991 (G325-5012)

PS/2 Model L40 SX Laptop Portable Computer
OS/2 2.0 Considerations
Comparing PC-DOS, OS/2, and AIX PS/2 – Part 2
Using Dual Displays with OS/2
Local Area Networks: The New Utility
And You Thought LANs Were Just for the Office!
Remote LAN Management Tools
New Horizons for IBM's Shielded Twisted Pair Cabling
Tuning and Self-Tuning Features of OS/2 LAN Server
NetWare Communications and Routing Protocols
Little Solutions for LANs

Issue 2, 1991 (G325-5011)

IBM PS/2 Model 90 XP 486 and Model 95 XP 486
Choosing an I/O Bus Architecture
The Network Is the Message
Invoking Printer Job Properties
Comparing PC-DOS, OS/2, and AIX PS/2
Programming PM Using the COBOL/2 Bindings
Installing and Using the DOS Database Requester
OS/2 LAN Server 1.3 Overview
IBM Windows Connection 2.0
SNA Definitions for 3270 Emulators – Part II
IBM THINKable™

Issue 1, 1991 (G325-5010)

XGA – Raising Video Expectations
Choosing betw. Shielded and Unshielded Wiring for Data Transmission
Compatibility of LAN Servers and Requesters
Running DOS LAN Requester and Novell NetWare Concurrently
Breaking the 640 KB DOS Memory Barrier
Understanding an OS/2 CONFIG.SYS File
OS/2 EE 1.2 Database Manager Performance
OS/2 EE 1.2 Competitive Performance
An Intelligent Front-End EASEL® Application
Enabling Software for National Language Support
SNA Definitions for 3270 Emulators
Diskette Failures Caused by Contamination

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 HPFS
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)

Introduction to Local Area Networks
IEEE 802.3 LAN Considerations
The IBM Token-Ring Network: A New Generation
How to Design and Build a 4-Mbit/s Token-Ring LAN
How to Design and Build a 16-Mbit/s Token-Ring LAN
Making the Cabling Decision
IBM Cabling System Highlights
Communication Strategy for Growth

“**The IBM PS/2 Server 295 is the first product of a long-term alliance between IBM and Parallan Computer, Inc.** (page 2)

“**Micro Channel PS/2 systems can accommodate 15 DMA adapters, 15 busmaster adapters, or any combination totaling 15.** (page 13)

“**The DMA print function on Micro Channel systems absorbs 96% of the printer management work formerly done by the processor.** (page 22)

“**The operating systems used in tablet systems are very different from those in traditional keyboard-based computers.** (page 29)

“**The 150-ohm STP cable still stands out as a first-class cabling choice for high-speed data transmission.** (page 38)

“**OS/2 2.0 has a migration database, DATABASE.DAT, which contains parameters and settings for commonly used DOS, Windows, and OS/2 programs.** (page 41)

“**During OS/2 2.0 system installation, there is a choice of over 200 printers.** (page 47)

“**The System Dump facility gives support personnel a mechanism for capturing the entire main memory image of a running OS/2 2.0 system.** (page 64)

“**The LOADHIGH (LH) command is used to load memory-resident programs into upper memory.** (page 85)

“**IBM’s Solutions Validation Laboratory will assist customers in establishing a test lab using specialized tools.** (page 91)

G325-5017-00