January 1993

Personal systems

IBM Personal Systems Technical Solutions

IBM®

$12.00

IBM Personal Systems Technical Solutions is published quarterly by the IBM National Software Systems Support Center, International Business Machines Corporation, Roanoke, Texas, U.S.A.

Editor	Libby Boyd
Assistant Editor	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.

Printed on recycled paper

Contents

Trademarks

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

Ultimedia, ThinkPad, RISC System/6000, POWERstation, SAA, CUA, Common User Access, SystemView, Workplace Shell, XGA, Series/1, THINKable, PS/ValuePoint, and S/390 are trademarks of International Business Machines Corporation.

Adobe Type Manager is a registered trademark of Adobe Systems, Inc.
Aldus is a registered trademark of Aldus Corporation.
Allen-Bradley is a registered trademark of Allen-Bradley Company.
Apple, TrueType, and Macintosh are registered trademarks of Apple Computer, Inc.
Atari is a registered trademark of Atari Corporation.
COMPAQ is a registered trademark of COMPAQ Computer Corporation.
CompuServe is a registered trademark of CompuServe, Inc.
Dell is a registered trademark of Dell Computer Corporation.
Open View is a trademark and Hewlett-Packard and HP are registered trademarks of Hewlett-Packard Company.
InfoCorp is a registered trademark of InfoCorp.
Intel is a registered trademark of Intel Corporation.
Ingram Micro D is a registered trademark of Ingram Industries, Inc.
EASEL is a registered trademark of Interactive Images, Inc.
IDC is a trademark of International Data Group, Inc.
ISO is a registered trademark of the International Organization for Standardization.
IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers.
COMDEX is a registered trademark of the Interface Group, Inc.
DeScribe is a registered trademark of Lenname Advanced Products, Inc.
Litton is a registered trademark of Litton Industries, Inc.
AGV External Control System is a trademark of Litton Industrial Automation Systems Ltd.
Lotus is a registered trademark of Lotus Development Corporation.
Micro Focus COBOL is a registered trademark of Micro Focus Limited.
Windows and Windows NT are trademarks and Microsoft is a registered trademark of Microsoft Corporation.
NCR is a registered trademark of NCR Corporation.
NetWare and Novell are registered trademarks of Novell, Inc.
Olivetti is a registered trademark of Ing. C. Olivetti & C., SPA.
Open Software Foundation, OSF, OSF/1, and Motif are registered trademarks of the Open Software Foundation.
PC Connection is a registered trademark of PC Connection, Inc.
SailRail is a trademark of SailRail Enterprises Ltd.
SCO is a registered trademark of Santa Cruz Operation, Inc.
Silicon Graphics is a registered trademark of Silicon Graphics, Inc.
SunConnect is a trademark of Sun Microsystems, Inc.
SynOptics is a registered trademark of SynOptics Communications, Inc.
Sytos Plus is a registered trademark of Sytron Corporation.
Taligent is a trademark of Taligent, Inc.
Toshiba is a registered trademark of Toshiba Corporation.
UNIVAC is a registered trademark of Unisys Corporation.
UNIX is a registered trademark of UNIX Systems Laboratories, Inc.
WordPerfect is a registered trademark of WordPerfect Corporation.
Ethernet is a trademark and Xerox is a registered trademark of Xerox Corporation.
X/Open is a trademark of X/Open Company, Limited.

PS/2 Desktop Security

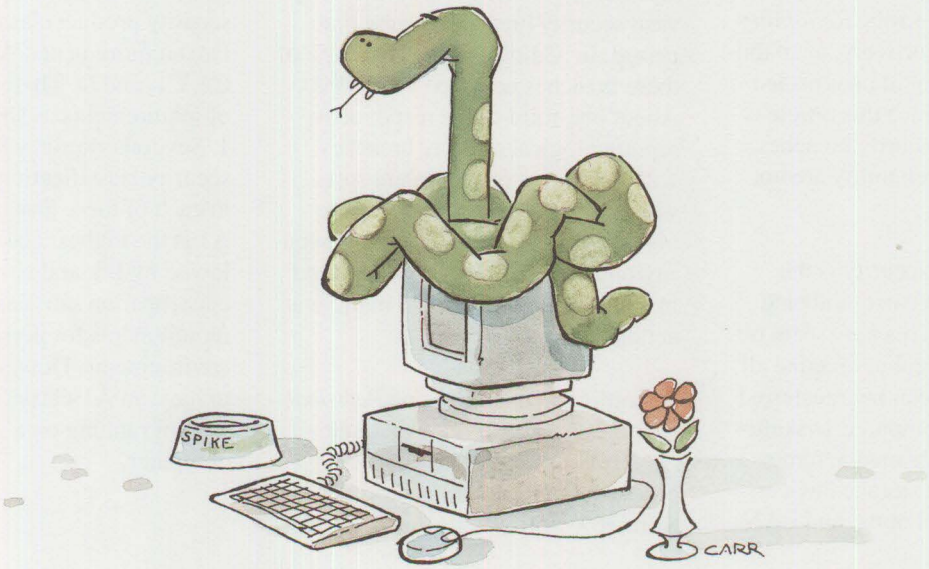
Dr. Patsy Bowlds
IBM Corporation
Basingstoke, United Kingdom

Ken Zubay
IBM Corporation
Boca Raton, Florida

This article describes the security objectives and features of IBM's PS/2® desktop computers, and is intended to assist users in their security planning. A brief description of the security features of operating systems is included as background for planning. Government requirements and security legislation are also presented.

Personal computer security is rapidly becoming a major concern. Security features are becoming important criteria for selecting hardware. Downsizing typically involves moving from larger systems that are located in controlled access areas and staffed by authorized personnel to personal computers that are located in unsecured areas with potential access by unauthorized users. This creates huge security exposures and enormous challenges in security management. In contrast to the medialess and non-intelligent terminals used with larger systems, personal computers provide local media, local software, local memory, and – perhaps most important – local data. In a highly connected environment, personal computers also provide the opportunity to access a major part of the data processing assets of an organization.

IBM has a long history of establishing security procedures and incorporating security features into products. Although users have overall responsibility for protecting their data and physical assets, IBM has a responsibility to provide systems, products, and services that help in preserving and protecting data security and privacy. In addition to the security features in IBM products, IBM's marketing groups alert and educate users about the significance of data security, and assist them in identifying specific data security needs.



Computer Security Concepts

Computer security is protecting physical assets and data from theft, unauthorized or accidental modification, destruction, and disclosure. While computer security encompasses accidental as well as intentional breaches, the primary focus of this article is on intentional security breaches.

Security breaches and protection schemes can be physical or logical. *Physical security* deals with the protection of assets or data from accidental or malicious physical damage or loss. *Logical security* is concerned with the use of software to protect data and computer assets. *System security* often refers to features that identify users, provide access controls, ensure user accountability, and enforce data confidentiality. System integrity and system security require a combination of hardware and software capabilities. IBM intends to provide secured workstation solutions that provide both system integrity and system security.

The terms data integrity and system reliability are sometimes used in security discussions. *Data integrity* is concerned with accidental damage to data that results from hardware, programming, or user errors. *Reliability* is concerned with recovery, availability, and the handling of unexpected error conditions. Since this article is about intentional security breaches, data integrity and reliability are not addressed.

Basic to computer security is the control of data movement into and out of the system. This may seem obvious on the surface, but imagine all the ways that data can be transferred into and out of a computer. Diskettes and other removable media, removable or replaceable storage devices, serial ports, parallel ports, and LAN adapters are just some of the ways. Security protection mechanisms to prevent unauthorized data transfer are a combination of physical and logical security methods.

Another basic security concept is the fact that the system owner and the system user are not necessarily the same person or entity. The person using the machine may or may not be authorized for that system. There may be multiple users, each of whom has different security authorizations.

Because of the inherent portable nature of personal computers, system security and integrity measures are not absolute. The features provided by hardware and software, however, can be effective deterrents for security breaches, and enablers for an effective computer security program.

Computer Security – Who Needs It?

Security problems are growing in today's world. A recent survey by the NCC Consultancy Group[1] indicated that over half of the respondents had suffered either a physical or logical security breach in the last five years. More than 40% of the respondents reported one or more significant physical security breaches during that period. In addition, more than 70% of these breaches occurred since 1990. About one-third of the respondents reported logical security breaches, with disruptive or fraudulent software (including viruses) being the biggest single cause. For logical security breaches, 42% of those reported in detail were attributed to malicious actions.

According to the magazine *Personal Computer*, May 1992, "An Audit Commission report showed an increase in computer-related crimes from 16% of fraud cases in 1987 to 73% in 1990. At the same time, it discovered that nearly two-thirds of these crimes were discovered accidentally – normal auditing procedures just weren't up to the job."

As pervasive and expensive as security problems are in the world of personal computing, some companies do not have defined security policies or consider security features in product selections. The NCC survey showed that only 55% of the respondents had a security policy in place.

Government Requirements and Legislation

United States Department of Defense (DoD) security requirements have been influential in defining security legislation and computer hardware and software implementations around the world. The source for these requirements is the *Department of Defense, Trusted Computer System Evaluation Criteria, DoD 5200.28 STD, 12/85*. The essence of the requirements is contained in the Assurance section, Requirement 6: "Trusted mechanisms must be continuously protected against tampering and/or unauthorized changes."

There are seven computer-system security product classifications in the DoD requirements: A1, B3, B2, B1, C2, C1, and D. There are four groups of requirements as shown in Figure 1. Several criteria, which vary by security classification, are specified in each of these groups. Currently, A1 is the highest classification, followed by B3, and so on. The C2 classification satisfies most security requirements for personal computing environments. However, it is possible to have an A1-classified operating system running on a secured personal computer.

[1] *NCC Survey of IT Security Failures or Breaches*. NCC Consultancy Group, Oxford Road, Manchester, England. 1991.

C1	C2	B1	B2	B3	A1	Criterion
Security Policy						
x	x	=	=	x	=	Discretionary Access Control
•	x	=	=	=	=	Object Reuse
•	•	x	x	=	=	Labels
•	•	x	=	=	=	Label Integrity
•	•	x	=	=	=	Exportation of Labeled Information
•	•	x	=	=	=	Exportation to Multilevel Devices
•	•	x	=	=	=	Exportation to Single-Level Devices
•	•	x	=	=	=	Labeling Human-Readable Output
•	•	x	x	=	=	Mandatory Access Control
•	•	•	x	=	=	Subject Sensitivity Labels
•	•	•	x	=	=	Device Labels
Accountability						
x	x	x	=	=	=	Identification and Authentication
•	x	x	x	x	=	Audit
•	•	•	x	x	=	Trusted Path
Assurance						
x	x	x	x	x	=	System Architecture
x	=	=	=	=	=	System Integrity
x	x	x	x	x	x	System Testing
•	•	x	x	x	x	Design Specification and Verification
•	•	•	x	x	x	Covert Channel Analysis
•	•	•	x	x	=	Trusted Facility Management
•	•	•	x	=	x	Configuration Management
•	•	•	•	x	=	Trusted Recovery
•	•	•	•	•	x	Trusted Distribution
Documentation						
x	=	=	=	=	=	Security Features User's Guide
x	x	x	x	x	=	Trusted Facility Manual
x	=	=	x	=	x	Test Documentation
x	=	x	x	x	x	Design Documentation

Key to Symbols:

= No additional requirements for this class

x New or enhanced requirements for this class

• No requirements for this class

Figure 1. Security and Integrity System Requirements

The importance of the C2 security criteria extends far beyond that of the DoD requirements. The issue of individual privacy and confidential personal information is a part of contemporary computer security requirements. Much legislation is either directly or indirectly linked to the DoD requirements. For example, the criteria have been applied to other federal agencies and to prime contractors handling confidential information (such as food stamp recipients, Medicare, Medicaid, and online tax preparation). The Computer Security Act of 1987 requires C2 security compliance for this type of information in the United States.

Many European countries have requirements similar to those of the U.S. DoD. In the United Kingdom, the 1984 Data Protection Act places responsibility for compliance with its terms on those who control the contents and use of personal data files. One of the principles of this act is that "appropriate security measures shall be taken against unauthorized access to, or alteration, disclosure or destruction of personal data, and against accidental loss or destruction of personal data." The single European market is also driving a piece of legislation designed to standardize the data protection laws of Europe within the next twelve months.

IBM has interpreted DoD's mainframe requirements for C2 security for personal computers based on long experience in dealing with government agencies requiring certification. Features that meet the C2 level for security and integrity are primarily provided by the operating system. However, to maintain a C2 system environment, users also must provide methods of ensuring that the software cannot be modified or removed. Users can do this by enclosing the computer in isolation rooms with controlled access; using monitoring devices to ensure that system modification is

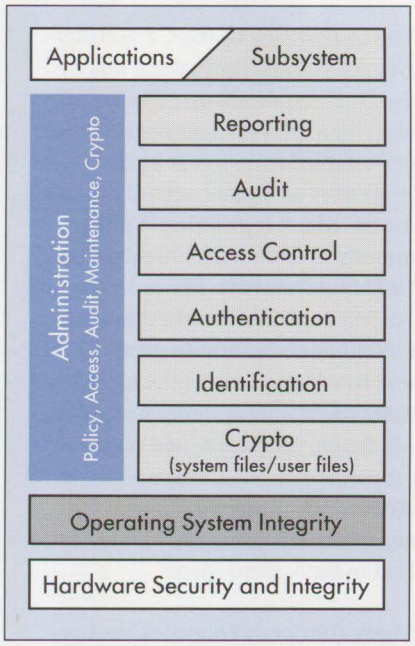


Figure 2. Elements of a Total Security Program

not possible; and employing detection methods to ensure that unauthorized media are not used. These techniques require a great deal of effort and expense to ensure system integrity, and may impose limitations on how and where personal computers are used.

IBM has determined that PS/2 hardware, microcode, and system software can be designed with features that enable a low-cost, convenient system integrity and security solution, and provide a secure platform for any operating system at all security levels.

To obtain security certification by the DoD, the specific hardware and software configuration must be submitted to the National Computer Security Center (NCSC) for testing. This organization evaluates computer system security products against DoD criteria. The process takes from 12 to 18 months. Although certification is important, the features of the secured PS/2 systems are independent of the operating system, and they provide a great deal of data and resource protection even if a non-secured operating system is used.

As shown in Figure 2, the hardware and software are a subset of the total requirements for security certification by the DoD. The total environment and procedures are essential elements of achieving and maintaining computer security and DoD certification. The system security and basic hardware integrity features provide the foundation of a total security program for personal computers.

System Security and Integrity Requirements

Certain hardware, microcode, and system software features are needed to meet DoD and most user requirements for system security and integrity. The following are descriptions of these requirements.

Tie-down capability: For workstations, the capability to tie down, bolt down, or clamp the system to permanent furniture or fixtures prevents the physical removal or theft of system and expansion units. For servers, this requirement is generally met by keeping the system in a locked room with controlled access, similar to the security requirements for mainframes and minicomputers.

Lockable covers: This is designed to prevent unauthorized access to the inside of the computer and changing the system's configuration. For example, this capability would inhibit removal of internal disk drives and memory. In addition, it prevents other security and integrity features from being disabled or bypassed.

Secure access openings: It is important to prevent foreign objects from being inserted through cabinet openings to bypass the security and integrity features. Openings such as air vents, diskette drives, cable and cord connectors, switches, and keylocks must prevent access.

Tamper-evident covers: If an unauthorized physical entry or attempted entry has been made into a secure system, there must be a method for notifying the owner. Because it is not realistically possible to prevent entry by a determined person, alerting the owner that such an entry has been made is an important security precaution. It is not sufficient to notify the user or to simply rely on detection by the user, because the majority of tampering occurs at the hands of users. The system must be able to differentiate between *owners* and *users* in order to report abuse by users.

Secure I/O cables: The rear sides of computer systems have several connectors that are used to send and receive data, such as the parallel and serial ports, LAN adapters, and Small Computer Systems Interface (SCSI) connectors. Anyone with the equipment and knowledge can use these connectors to access data on a fixed disk drive or to intercept communications. To prevent devices from being removed from or attached to a system with I/O cables, a method of securing the cables and cords and preventing attachment of devices to unused connectors must be provided.

Set or change password capability: Systems are required to have a method for setting and changing a power-on password and a privileged access password. The privileged access password is needed to differentiate between the owner and users.

Software-readable system identification: Each system and expansion unit must have a unique identification that can be read by the operating system. This prevents an unauthorized system or substitute system from participating in a network or other communications link.

Secure removable media: A mechanism is required to prevent an unauthorized person from physically

removing or inserting media (such as diskettes). If system power is lost, this feature must be able to return to its state (enabled or disabled) when power is restored.

Ability to disable BASIC in Read-Only Memory (ROM): A method of preventing unauthorized users from entering BASIC programs during Initial Program Load (IPL), both local and remote, is a requirement. If this were not done, users could enter BASIC programs and use them to bypass some other security features of the system.

Secure IPL source: A method of securing the IPL source must be provided. This is important for ensuring that the secure operating system is loaded during the IPL process. If, for some reason, the drive or LAN providing the secure operating system is unavailable, the operation of the system must be inhibited, and the system prevented from activating BASIC.

Interface to security adapter: Systems must allow a security adapter to be installed and configured. Such an adapter may provide specialized security and auditing capabilities (such as encryption or user identification using biometrics).

Permanently erasable disk and memory: This requirement is necessary to prevent data from being restored from disks after it has been logically deleted, and to provide a method to clear system memory after a change of users, or when power is turned off. Intrusions cause power to be turned off, which clears system memory. This capability is sometimes called *object reuse protection.*

Lockable I/O ports: This provides selective access to I/O ports and systems media. This is important when more than one user is authorized to access the system, but is denied access to certain disks or communications ports.

Diagnostic tests: These tests ensure that the security and integrity features are working.

Protection against securing unsecured systems: A method must be provided to ensure that an unsecured system is not converted to a secured system by an unauthorized person.

Security features user's guide: Documentation is required that describes the system's security features, how to use them, and their limitations.

PS/2 Security Features

IBM has chosen PS/2 systems as the standard-bearer for security. A broad range of security features are available in the newly announced IBM PS/2 desktop systems.

Paired with appropriate operating system support, the new, secured PS/2 systems can protect operating systems at any DoD classification. Other IBM personal computers have some of these features, but PS/2 systems have security as a primary design consideration. All PS/2 systems meet several security requirements described previously. Newly announced PS/2 models provide additional security features. Some features are available as options for earlier systems.

Implementation Notes

IBM used the following general guidelines to decide how to implement the DoD security requirements in the PS/2 product line. In some cases, these guidelines helped IBM decide whether to implement security features in hardware or in the operating system.

- The level of security, from no security to the maximum available, should be customized by the system owner.
- The cost of security should be in line with the value of the data.
- The system owner should be able to modify security as required.
- The security functions should be easy to administer and use.
- The implementation should minimize storage and performance impacts.
- The security features should be customized for each user whenever possible.

Secured PS/2 Models

In September 1992, IBM announced three PS/2 systems that have enhanced hardware integrity features:

- PS/2 Model 56 486SLC2
- PS/2 Model 57 486SLC2
- Ultimedia™ Model 57 486SLC2

The security features in the new PS/2 models include the following:

Tie-down capability: There are two small holes in the back of the computer that can be used to secure the system to a permanent fixture. A U-bolt can be installed in the holes and a cable or chain can be attached to prevent physical removal or theft.

Lockable covers with tamper-evidence feature: These systems have a keylock for their covers and internal I/O devices. In the locked position, it mechanically prevents the covers from being removed. The key has been changed to a type that can be duplicated only by the manufacturer.

If the covers are forced open, an electromechanical switch and perimeter sensor detect the intrusion. If the computer is on during the break-in attempt, it will cease working. The next time the computer is started, the Power-On Self Test (POST) routine displays a message informing the user of the intrusion, and requires that the automatic configuration pro-

gram be run before the computer can be used. This is done to flag any configuration changes that may have occurred due to the intrusion – such as the removal of a disk drive. In addition, the system cannot be used without the power-on and privileged access passwords (if they have been set). A provision for maintenance allows the system to be used without the covers in place. However, to use this feature, the key must have been used to remove the covers.

Other systems may have lockable covers. However, it is not difficult to pry off the system unit cover, disable or unplug the key mechanism, and get inside the system. The tamper-evident mechanism is the critical feature that flags the intrusion to prevent operation of the system after a forced entry. The tamper-evident mechanism is the heart of the continuous protection required by the DoD and other agencies.

This PS/2 detection feature is valuable for detecting the person most likely to break into a secured workstation – the person using the machine. Once the machine has been disabled, the system owner or administrator must be contacted to reset the system.

Secure access openings: The basic design of the keylock, cover interlock switch, diskette media lock, and other openings of the new systems prevents poking a spring hook inside to disable the battery and so on.

In the original PC AT® systems and in most of today's clones, the cover lock also locks the keyboard. If the cover lock is deactivated, the keyboard is enabled. One could easily insert a tool through the air vent in the front of the AT and deactivate the keylock, thereby gaining access to the system. Even for those systems with keyboard passwords and power-on passwords, if the battery is accessible, it is possible to disable the password protection.

The privileged access password in the new PS/2 systems is not linked to the battery. It is not disabled even if the battery is disabled. This provides an additional measure of protection.

Secure I/O cables: The rear-panel security option available for the new systems is an enclosure that is secured to the back of the computer by the cover lock. Its function is to prevent the cables from being removed and other cables from being attached. This effectively secures the serial, parallel, and SCSI cables, as well as other ports and cables provided by adapters. It prevents someone from attaching a device through these connectors and gaining access to the data in the system.

The cable cover also has a tamper-evident feature. If the cover is forced open, the system will not operate until the privileged access password is entered.

The IBM PS/2 Cable Cover 2 option is provided for the IBM PS/2 Model 56 486SLC2. The IBM PS/2 Cable Cover 3 option is provided for the PS/2 Model 57 486SLC2 and the PS/2 Ultimedia Model 57 486SLC2.

Set or change password: All PS/2 systems provide power-on password and keyboard password protection. The power-on password must be entered correctly each time the system is turned on. After three incorrect attempts, the system must be turned off and back on in order to try again. The keyboard password is used to lock the keyboard without turning off the computer. It also prevents rebooting the system by pressing the Ctrl+Alt+Del keys. PS/2 systems also provide what is called an *unattended server mode* or a *network server mode*. This mode allows other computers to access a fixed disk drive while the keyboard is locked.

Because it is necessary to differentiate between the owner and users, another level of password protection has been provided in the new systems. This security feature is called the *privileged access password*. It provides a much higher level of security when used with an operating system that controls access through the use of passwords. Systems are shipped with the privileged access password disabled. To set this password, a jumper on the system board must be moved to put the system in the change state. Once this password is set, it cannot be overridden or removed by an unauthorized person. The owner or administrator must keep a record of the privileged access password in a safe place. If the owner or administrator misplaces or forgets the privileged access password, the system board must be replaced. The privileged access password restricts access to programs, prevents the IPL source and sequence from being changed, and effectively deters unauthorized modifications to the hardware. Even after a forced entry is detected by the tamper-evident cover switch, the privileged access password (if it has been set) must be used to make the system operate.

The privileged access password is stored in a special type of read-only memory called flash EEPROM (Electrically Erasable Programmable Read-Only Memory).

System identification: The new systems provide a wealth of information stored in ROM. The system board serial number, the model and submodel byte data, the system serial number, the system board part number, the replaceable unit part number, and the manufacturing location are included in the software-readable information. This collection of data about the system is called the Vital Product

Data (VPD). The VPD information is accessible using the utility called the System Information Tool, which is part of the OS/2® version that comes preinstalled on new PS/2 systems. In addition to helping prevent substitution of an unauthorized system and aiding in matching a system with its authorized user, VPD capability is beneficial in inventory and asset management. The System Information Tool provides detailed adapter, memory, and disk configuration data for these Micro Channel® systems.

Secure removable media: A new, optional 2.88 MB diskette drive with security features is available for the new systems. The new diskette drive is a 3.5-inch, one-inch high drive with media sense capability for the standard diskette capacities of 720 KB, 1.44 MB, and 2.88 MB. It can read and write data to a formatted capacity of 2.88 MB, while maintaining read and write capability with 720 KB and 1.44 MB diskette drives. A new control signal has been added to the diskette interface that supports `LOCK`, `UNLOCK`, and `EJECT` commands issued by the operating system. If the privileged access password is not set, the diskette is unlocked during POST. If the password is set, the boot process does not unlock the diskette drive unless it is the designated IPL source. In this case, the `LOCK` and `UNLOCK` state is controlled by the Lock/Unlock system utility. For SCSI devices, there is a proposed standard `UNLOCK` command. In this case, the operating system will control the `LOCK` command if the privileged access password is set. Access to the unlocking function with specific user authorization will be controlled by future secured system software – for example, the Secured Workstation Manager.

If power loss occurs, the system retains its state (secured or unsecured) independent of the state of the battery. A diskette can be inserted in the drive, but it cannot be removed if the power is off. When the drive is turned on and locked, the media cannot be inserted or removed.

Secure IPL source: The new systems allow the system owner or administrator to select the IPL source and sequence. The IPL sequence is stored in a region of the system's EEPROM, and can be read, but not written, without the privileged access password. The setup routine ensures that at least one IPL source is specified if the privileged access password is used. This allows the system owner to control the IPL source, but prevents the user from modifying the source and sequence. For example, the diskette drive can be excluded as an IPL source. This feature helps to ensure that the system owner's specified operating system is loaded.

Earlier PS/2 models with Initial Microcode Load (IML), in which part of the system's microcode is loaded on the disk drive, could select the IPL source. However, the information was stored in Complementary Metal-Oxide Semiconductor (CMOS), which could be disabled by removing the battery. Storing the IPL sequence in EEPROM protects it from being deactivated by removing the battery.

Interface to security adapter: The Micro Channel bus in PS/2 systems provides an excellent interface for cryptographic and other security adapters. Some of these adapters are busmasters with on-board processing capabilities. The data integrity features and high data-transfer rates supported by Micro Channel architecture are important for these applications. The PS/2 Model 56 486SLC2 provides three Micro Channel slots; the PS/2 Model 57 486SLC2 and Ultimedia Model 57 486SLC2 provide five Micro Channel slots.

IBM offers a family of workstation security products called the *Transaction Security System*, which consists of the following components:

- IBM 4755 Cryptographic Adapter
- IBM 4754 Security Interface Unit
- IBM Personal Security Card

Several new models of the IBM 4755 Cryptographic Adapter have recently been announced, including enhanced versions for PS/2 systems. These adapters have a much higher performance rate, and support a larger number of cryptographic standards, including the Data Encryption Standard (DES) and the Rivest, Shamir, and Adleman (RSA) cipher.

Ability to disable ROM BASIC: A ROM version of BASIC is not provided for system boot. A full BASIC program is available as a separate software product.

Diagnostic tests: These are an integral part of the system board diagnostics. The functionality of the security features is assured during the POST routine.

Protection against securing unsecured systems: A switch under the lockable covers allows the privileged access password to be written. With the covers locked and the switch in the locked state, an unauthorized person cannot set it.

Permanently erasable disk and memory: The erasable disk capability is a feature implemented in the Secured Workstation Manager. In the new secured systems, the system memory is cleared when power is turned off. As intrusions cause power to be turned off, system memory is automatically cleared when an intrusion is detected.

Lockable I/O ports: The rear-panel security option provides an "all or none" type of access to I/O ports. In addition to this support, future software will provide I/O port manage-

ment based on the current signed-on user authorization.

Security features user's guide: The new systems come with publications that describe the security features and their use. These descriptions are included in the *Personal System/2 User's Handbook* for the new systems, and in the *Personal System/2 Micro Channel Computer Reference*.

Operating System Security Features

Hardware security features must work together with operating system security features that can be invoked by the system owner or administrator – as opposed to the person using the machine. The hardware security features must prevent corruption of the operating system. The IBM Secured Workstation Manager (SWSM) is IBM's DOS/Windows security product. It has C2 functions, including Discretionary Access Control (DAC), Object Reuse Protection (ORP), Identification and Authentication (I&A), and Audit (AUD). Key features of SWSM are as follows:

- Provides security for DOS and DOS plus Windows™, desktops, towers, laptops, notebooks, and portable workstations.
- Limits usage of workstations to authorized individuals only.
- Establishes a secure sign-on to the workstation for use in access control and auditing. Users can validate their identity by using a combination of passwords, electronic tokens, the IBM Personal Security Card, and IBM Signature Verification.
- Controls access to protected files through access control lists. Protected files are also automatically and transparently encrypted to protect confidential information. Protected files can be stored on fixed or removable media, or on LAN servers. Data file confidentiality is maintained, even if there is an attempt to access them by an unauthorized operating system.
- Allows a central site administrator to implement and distribute security rules to protected workstations. Local users can further restrict, but not reduce, the security established by this administrator.
- Maintains user accountability through audit logging of user and administrator activities. For example, all sign-on attempts and attempted security violations are monitored and recorded.
- Provides an Application Programming Interface (API) with the optional System Integration Support program toolkit. This allows users to customize their programs to take advantage of the security features.
- Establishes a secured access sign-on to allow SWSM users to sign on easily to other secured environments (such as LAN Servers and OS/400®).
- Works with the IBM AntiVirus/DOS product, which contains support for virus protection, detection, and correction.
- Available in two versions: DES and non-DES. Both versions are identical except for the algorithms used to encode user files. Protected files can be shared only between like versions.

IBM plans to use the newly announced, secured PS/2 systems with all future versions of operating system security enhancements as a platform for security certifications.

Summary

All PS/2 systems provide numerous security features to meet a variety of security needs. These features are cost-effective implementations that provide flexibility, convenience, and ease of use. The security features of the IBM PS/2 Models 56, 57, and Ultimedia Model 57 486SLC2 systems provide the hardware capabilities to meet the DoD requirements for software certification and most other security needs.

Dr. Patsy Bowlds *is a senior technical consultant on loan to the IBM European Personal Systems Center in England from the Boca Raton Development Laboratory. She has held numerous development and marketing positions in IBM, including manager of storage systems, personal systems strategy manager, and PS/2 technical marketing support. Pat is author of* Micro Channel Architecture: Revolution in Personal Computing, *published by Van Nostrand Reinhold. She has been awarded several patents. She earned her PhD at the University of Alabama.*

Ken Zubay *is a senior programmer in the IBM Personal Systems Programming Center in Boca Raton. His most recent project was co-developing the Secured Workstation Manager. He represents IBM Boca Raton at worldwide IBM security councils. Since joining IBM in 1968, Ken has worked in operating system development, special-bid marketing, and PC workstation development. He is co-holder of three U.S. patent disclosures about security features in PS/2 workstations.*

IBM 486SLC2: System Performance Implications

Walt Adams
IBM Corporation
Boca Raton, Florida

Patsy Bowlds and Tikiri Wanduragala
IBM Corporation
Basingstoke, United Kingdom

This article gives an overview of the IBM 486SLC2 processor, which is used in PS/2 Models 56 and 57 486SLC2, in processor upgrades for the PS/2 56 and 57, and in certain IBM ThinkPad™ models.

The IBM 486SLC2 processor is a high-performance, 32-bit microprocessor that provides performance similar to Intel® 486SX microprocessors. The IBM 486SLC2 chip has the same exterior dimensions as the IBM 386SLC and Intel 386SX microprocessors. Internally, its many advanced features provide outstanding performance. It supports all the 486SX instructions, and is fully compatible with the large installed base of 8086, 286, 386, and 486 software.

Internally, the 486SLC2 microprocessor has 32-bit data and address paths. These 32-bit paths maximize throughput between the internal cache and the instruction execution unit. Externally, the 486SLC2 has a 16-bit data path and a 24-bit address path. These external paths contribute to its small footprint.

The primary features that enable the 486SLC2 processor to perform at 80486 levels are its internal 16 KB cache and its clock-doubling technology. An IBM 486SLC2 chip running at 40/20 MHz (where 40 MHz denotes the internal processing speed, and 20 the external speed) can achieve performance comparable to an Intel 486SX chip running at 25 MHz. An IBM 486SLC2 processor at 50/25 MHz can achieve performance comparable to a 486DX processor running at 33 MHz.

Low power consumption and power management instructions are also key features of the IBM 486SLC2 processor. In addition to being used in desktop environments, this processor is ideal for portable computers, where power and size are critical factors.

Comparing IBM SLC Processors with Others

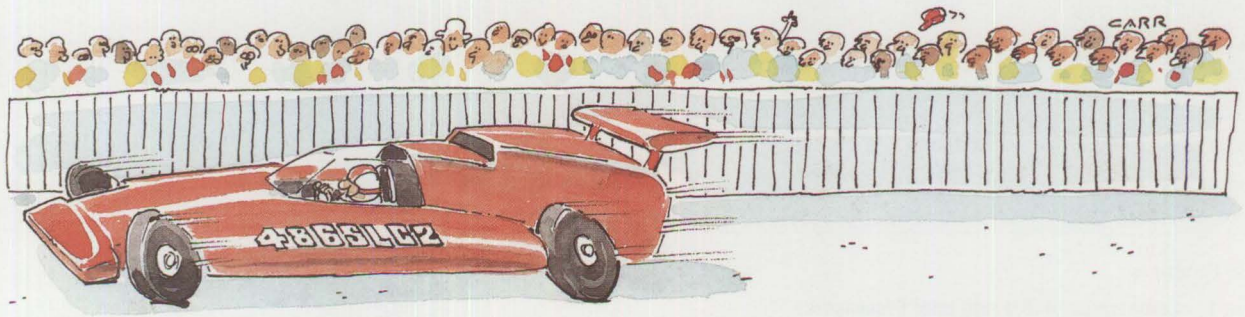
IBM SLC processors share features with both the Intel 386SX and 486SX microprocessors. The most important feature is software compatibility. Application software and operating system software for the 386 processor will run on the IBM 386SLC chip, and software for the 486 processor will run on the IBM 486SLC2 chip. This includes OS/2, DOS, Windows, SCO® UNIX®, and Novell® NetWare®. Application software for these operating systems – whether 16- or 32-bit applications – runs unmodified.

Figure 1 summarizes the features of IBM SLC processors compared to Intel processors.

Performance Implications

All processors listed in Figure 1 have 32-bit internal data paths. However, the external data path is 16 bits wide for some and 32 bits for others. Because the external data path in the IBM 486SLC2 is 16 bits wide, the natural question is whether the IBM 486SLC2 can be as fast as the Intel 486SX processor, which has a 32-bit external data bus. The answer is yes, depending on the application.

The large internal cache (16 KB) of the IBM 486SLC2 processor – an advantage compared to the 8 KB cache in a 486SX chip – reduces the processor's need to use its data path. In addition, the IBM 486SLC2 and



386SLC processors access their internal cache via a full 32-bit data path.

The effectiveness of the cache depends on each individual application's instruction and data locality – that is, the pattern of referencing previously accessed memory. It is not uncommon for applications to have an average cache hit rate in excess of 90%. This means that 90% of the memory reads will not need to access memory, because the cache already contains the information being fetched.

The type of cache used in the Intel 486SX and IBM 486SLC2 chips, called a *write-through* cache, is effective when the processor is reading data or fetching instructions. When a write is done to memory, it causes data to be sent on the processor's external bus. Write buffers allow the processor to continue executing without waiting for the memory write to finish. The larger the number of write buffers, the less likely the processor will have to wait for a write to complete (unless the application is performing many writes in quick succession).

The size of the address bus in IBM SLC chips – 24 bits – is not a performance issue unless it becomes necessary to use more than 16 MB of physical memory to reduce paging in a virtual memory environment. Then, a 32-bit address bus is necessary to provide more than 16 MB of physical memory. Both the Intel 486SX and the IBM 486SLC2 processors have the same virtual address space of 64 terabytes (TB). In practice, the amount of virtual memory is much less than 64 TB because of operating system constraints.

Like the Intel 486DX2, the IBM 486SLC2 has clock-doubling technology. When driven by an external frequency of 20 MHz, the internal frequency will be 40 MHz; a 25 MHz external clock is doubled to 50 MHz internally. It is primarily this feature that gives the IBM 486SLC2 microprocessor a performance comparable to that of a 486SX microprocessor. Clock doubling also gives the Intel 486DX2 and OVERDRIVE microprocessors an advantage over Intel 486SX and 486DX microprocessors.

Like the Intel 486SX, the IBM 486SLC2 processor does not contain a math coprocessor. To gain this function, an 80387SX coprocessor is used with the IBM 486SLC2 chip. In applications that are almost exclusively floating point, the internal floating-point unit in an Intel 486DX, 487SX, or OVERDRIVE processor has an advantage over the external 387SX that is used with the IBM 486SLC2. Many applications that use floating point also execute many instructions that are not floating point. These applications can enjoy excellent performance with the 486SLC2 plus 387SX combination.

486SLC2 Processor Upgrade

IBM Personal System/2® Models 56 SX and 56 SLC systems (and their Model 57 counterparts) can be upgraded to use the IBM 486SLC2 processor. In the upgraded systems, the 486SLC2 runs at 40 MHz internally and 20 MHz externally.

The upgrade consists of a small circuit board and a new reference diskette. The circuit board plugs into the

CPU Designation	Intel 386SX	IBM 386SLC	Intel 386DX	Intel 486SX	IBM 486SLC2	Intel 486DX	Intel DX2 and OVERDRIVE
Internal Data Path	32-bit	32-bit	32-bit	32-bit	32-bit	32-bit	32-bit
External Data Path	16-bit	16-bit	32-bit	32-bit	16-bit	32-bit	32-bit
Internal Cache	None	8 KB	None	8 KB	16 KB	8 KB	8 KB
Write Buffers	0	2	0	4	2	4	4
Address Interface	24-bit	24-bit	32-bit	32-bit	24-bit	32-bit	32-bit
Physical Addressable Memory	16 MB	16 MB	4 GB	4 GB	16 MB	4 GB	4 GB
Virtual Addressable Memory	64 TB[1]	64 TB	64 TB	64 TB	64 TB	64 TB	64 TB
Math Coprocessor	External 387SX	External 387SX	External 387DX	487SX or OVERDRIVE	External 387SX	Internal	Internal
Clock Doubled	No	No	No	No	Yes	No	Yes

[1] TB=Terabytes

Figure 1. Comparison of IBM and Intel Processors

PS/2 56 486SLC2	PS/2 56 SX
486SLC2 processor at 50/25 MHz	386SX processor at 20 MHz
Optional 387SX math coprocessor at 25 MHz	Optional 387SX math coprocessor at 20 MHz
Micro Channel architecture	Micro Channel architecture
Higher performance SCSI interface on the local bus	SCSI interface on the local bus
XGA-2 graphics	16-bit VGA graphics
2.88 MB diskette drive	2.88 MB diskette drive
8 MB, 70-ns memory; 16 MB maximum	4 MB, 70-ns memory; 16 MB maximum
DMA parallel and two serial ports	DMA parallel and one serial port
OS/2 2.0 Preload	OS/2 2.0 Preload
104 MB and 212 MB, 12-ms SCSI disks	80 MB and 160 MB, 16-ms SCSI disks

Figure 2. Comparison of PS/2 56 486SLC2 and 56 SX Systems

system board (where the math coprocessor socket is located), and the reference diskette is used to update the Initial Machine Load (IML). A math coprocessor socket on the upgrade card allows users to add a 387SX-20 coprocessor. This processor upgrade card is very similar to the IBM 386SLC cached processor option, which is available for the PS/2 56 SX and PS/2 57 SX systems.

PS/2 56 486SLC2 and 57 486SLC2 Systems

The PS/2 Models 56 486SLC2 and PS/2 57 486SLC2 systems have many significant enhancements over the PS/2 56/57 SX and PS/2 56/57 SLC systems. Key differences are that the subsystems of the PS/2 56 486SLC2 have been upgraded or have higher performance. Figure 2 compares PS/2 56 SX and 486SLC2 systems.

Structural Overview of PS/2 56 486SLC2 System

The Micro Channel connectors, the video subsystem, and other system board Input/Output (I/O) are located on the Micro Channel bus. The Small Computer Systems Interface (SCSI) controller is on the local bus. Both the local bus and the Micro Channel bus are 16-bit implementations. The clock frequency of the local bus is higher than that of the Micro Channel bus, which translates into performance gains in some applications. Figure 3 shows an overview of the components of the PS/2 56 486SLC2 system.

Processor/memory subsystem: The 486SLC2 processor's 24 address lines can address up to 16 MB of memory. Memory is located on the system board using three connectors. Each connector can support 2 MB, 4 MB, or 8 MB of 70-ns memory.

XGA-2 video subsystem: The XGA-2 video subsystem has enhancements specifically for today's Graphical User Interface- (GUI-) based operating systems. At 640 x 480 resolution, the XGA-2 graphics subsystem improves application performance up to 67% over 16-bit VGA. IBM's new XGA-2 subsystem sets new heights of performance and function. Even at 1024 x 768 resolution, the XGA-2 subsystem can be more than twice as fast as the original XGA™.

New functions in XGA-2 include a flicker-free refresh rate of 75 Hz at both 640 x 480 and 1024 x 768 resolutions, as well as hardware and software support for 16-bit direct color, to display up to 65,536 colors simultaneously. Enhancements that boost performance include a standard 1 MB of 100-ns VRAM (compared to 512 KB of 150-ns VRAM, standard in XGA graphics), 32-bit internal data paths, and significant device driver enhancements.

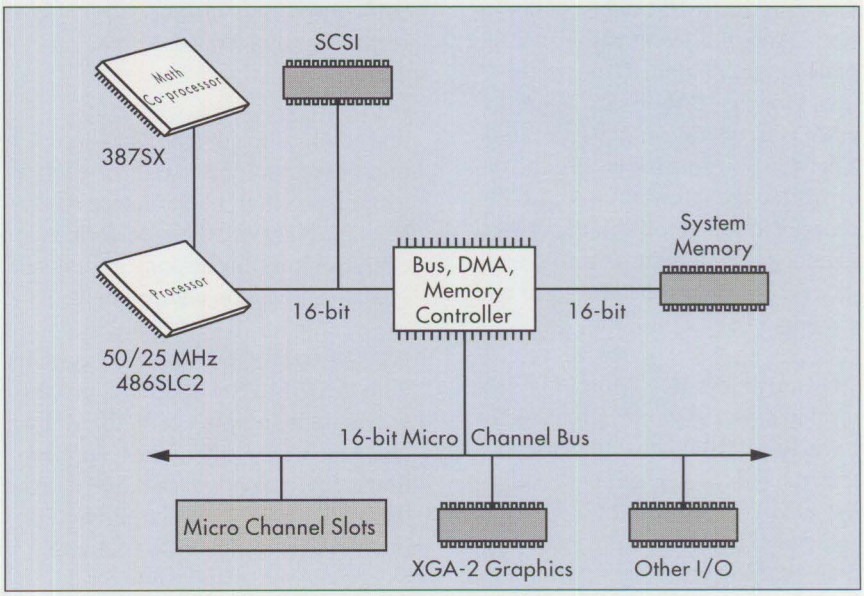


Figure 3. Components of the PS/2 Model 56 486SLC2 System

I/O bus and devices: The 16-bit Micro Channel architecture I/O bus provides a high level of system reliability and data integrity. Micro Channel support for busmasters allows for improved system throughput and greater overall processing power. Streaming data transfers, data parity, and synchronous channel check features are supported by the bus between expansion boards that incorporate these features.

The storage subsystem includes the SCSI controller and its attached devices. The SCSI controller is located on the system board, and does not occupy a slot. It is attached to the local bus, as shown in Figure 3. Designed as a busmaster, it has direct access to system memory on the high-speed local bus. The average seek time of the 104 MB and 212 MB disk drives is 12 ms. Up to seven devices can be attached to the SCSI controller, providing a high degree of flexibility and expansion for hard disk drives, optical drives, and backup tapes.

Processor and memory characteristics: An Intel 486SX processor has a 32-bit path from the processor to memory, while the IBM 486SLC2 processor has a 16-bit data path. Sometimes the 16-bit path can affect performance, although the performance of the IBM 486SLC2 processor is frequently comparable to the Intel 486SX. This is because the on-board cache in the IBM 486SLC2 reduces the need to fetch data from memory, which reduces traffic on the data path between the processor and memory.

I/O characteristics: Most I/O devices used in personal workstations today have 16-bit data paths. Because of this, the lack of a 32-bit data path for the IBM PS/2 56 486SLC2 system is somewhat irrelevant. The most common exceptions are high-performance LAN adapters. Most 32-bit Micro Channel adapters also are designed to function efficiently in systems with 16-bit data paths. Examples of such IBM adapters are the 32-bit SCSI and XGA adapters.

In the server arena, 32-bit devices, such as disk and LAN adapters, are more common. It is with server systems that the absence of the 32-bit data path is most significant. However, compared to 386DX and 486SX systems based on the AT bus, no significant effect is anticipated in I/O-intensive applications, because the AT bus supports only 16-bit transfers. Micro Channel busmasters and the more efficient Micro Channel bus provide some performance benefits compared to AT adapters and the AT bus.

The PS/2 Models 56 and 57 486SLC2 feature two Direct Memory Access (DMA) serial ports and one DMA parallel port. In systems without these features, the processor manages serial and parallel operations, which are interrupt-driven processes (see "Parallel Port Protocols" in this issue). The consequence is that they require significantly more processor cycles than DMA serial and parallel operations. Therefore, systems with these DMA features may have some performance advantages, especially in multitasking environments.

Summary

The combination of the IBM 486SLC2 microprocessor, Micro Channel architecture, and high-performance subsystems provides excellent performance characteristics for desktop systems in typical application environments.

PS/2 Models 56 SX and SLC systems with the IBM PS/2 486SLC2 processor upgrade (running at 40/20 MHz) compete effectively in most application areas against systems based on the Intel 486SX-25. A large 16 KB internal cache and clock-doubling technology are responsible for the 486 level of performance that this processor achieves.

The new PS/2 Models 56 and 57 486SLC2 systems have performance comparable to 486DX-33 systems in most applications. This level of performance is attributable to the IBM 486SLC2 50/25 MHz processor, XGA-2 graphics, and high-performance disk drives.

Walt Adams *is an advisory programmer in the Workstation Performance Analysis department, Entry Systems Technology Laboratory, Boca Raton. He joined IBM in 1983 to design and implement an X.25 device driver for the IBM Series/1™ computer. Walt's current assignment is performance analysis of PS/2 and PS/ValuePoint™ systems. He has a BA in psychology from the University of Virginia and an MS in computer science from the University of Connecticut.*

Dr. Patsy Bowlds *is a senior technical consultant on loan to the IBM European Personal Systems Center in England from the Boca Raton Development Laboratory. She has held numerous positions in IBM, including manager of storage systems, personal systems strategy manager, and PS/2 technical marketing support. Pat is author of* Micro Channel Architecture: Revolution in Personal Computing, *published by Van Nostrand Reinhold. She earned her PhD at the University of Alabama.*

Tikiri Wanduragala *is a senior specialist on the marketing staff of the IBM European Personal Systems Center in Basingstoke, England. His current responsibilities include technical marketing of PS/2 client/server solutions. He also has held positions in compatibility testing, performance analysis, and OS/2 technical marketing.*

Micro Channel Developers Association

Ramiz H. Zakhariya
Micro Channel Developers Association
Redding, California

This article describes the Micro Channel Developers Association, whose membership consists of microcomputer hardware and software developers supporting the Micro Channel.

The Micro Channel Developers Association (MCDA) is an independent, worldwide, non-profit organization established to facilitate the evolution of the Micro Channel as an industry-wide open standard. The association's primary focus is to help developers in building compatible Micro Channel-based computer products, and to explain to users the benefits of the Micro Channel architecture.

Fourteen industry leaders joined forces to form the Micro Channel Developers Association in October 1990. Now, only two years later, the total number of members in the MCDA is 90. Membership in the MCDA is open to all companies and individuals who are either directly or indirectly involved with developing products or services for Micro Channel architecture.

MCDA Services

The association offers its members many essential services:

- Micro Channel architecture seminars – four per year (two in Boston and two in San Jose) plus on-site at user locations, if requested
- MCDA database, including technical specifications, questions and answers, and marketing information
- MCDA newsletter
- Micro Channel Adapter POS ID assignment now the responsibility of the association (IBM and the MCDA are working on the transition logistics. IBM has been offering this service.)
- Product referral
- Product directory
- Technical literature and specifications
- Technical conferences
- Technical hotline
- MCDA bulletin board

MCDA current technical programs include common design tools, common `SETUP` utility, design verification facility, compatibility certification, architecture roadmap, standard Micro Channel Backplane, Subsystem Control Block architecture specification, Micro Channel architecture specification, multimedia extensions, and live insertion capability.

Marketing and promotional programs include positioning strategy, participation at trade shows, sales aid kits, competitive analysis, product directories, and architecture promotion.

Membership Benefits

Membership in the MCDA has many benefits.

- **Leveraged advertising**. As a member, you receive advertising that is published independently by the MCDA. Members value this advertising as equal to, or more than, their annual dues.
- **Early visibility**. You have access to information when it becomes available, even in preliminary form. You no longer have to find out belatedly that specific information exists.
- **Input on your technical requirements**. Through the association and its technical meetings, you can propose and lobby for features that you feel are necessary.
- **Interaction with competitors**. Through the MCDA's working groups, you have the opportunity to meet with peers and competitors. Inevitably, this interaction leads to learning something new about the industry and its environment.
- **Specifications for MCDA offerings**. The MCDA is in the process of defining several specifications and programs, both technical and promotional. As a member, you can join this important effort and use the results to your benefit.
- **Technical assistance**. Through the various programs implemented by the association, technical help is available to members in several forms: technical references, design seminars, workshops, answers to technical questions, electronic conferences, and technical working groups.
- **Common design tools**. The association has completed phase one of a comprehensive design tool set that includes behavioral models, test vectors, and simulators. These tools are available now. It is intended that this tool set will be used by all members to ensure a higher degree of compatibility and a shorter development cycle.

- **Common configuration utility**. The MCDA has completed the specification for its common configuration utility to be used by all developers. This new SETUP utility should be available in early 1993.
- **MCDA database**. This database will serve as a repository for technical questions and answers, design guidelines, specifications, the library of test vectors, and statistical information about members' products and the market. The MCDA database should be online by the end of 1992.
- **Starter Kit**. This kit, sent automatically to new members, consists of:
  - *Micro Channel Architecture Specification*. Micro Channel Developers Association, January 1992.
  - *IBM PS/2 Hardware Interface Technical Reference:*
    - *System-Specific* (S84F-9807) and *Update* (S04G-3280)
    - *Architectures* (S84F-9808) and *Update* (S04G-3282)
    - *Common Interfaces* (S84F-9809) and *Update* (S04G-3281)
    - *BIOS Interface Technical Reference* (S04G-3283)
    - *Architecture Supplement* (SCB) (S85F-1678)
    - *RISC System/6000™ POWERstation™ and POWERserver* (SA23-2643)
  - MCDA Technical Strategy Working Group minutes since January 1991
  - MCDA Marketing Strategy Working Group minutes since April 1992
  - *The Catalog of International Micro Channel Expansion Adapters*
  - "Focus on Micro Channel Architecture," a special supplement to *IBM Personal Systems Technical Solutions* (G325-5018)
  - The current list of MCDA members
  - The MCDA newsletter
  - Bowlds, Patsy A., Dr., *Micro Channel Architecture: Revolution in Personal Computing*. Van Nostrand Reinhold, 1991. ISBN 0-442-00433-8.
  - Heath, Chet and Rosch, Winn L. *The Micro Channel Architecture Handbook*. Prentice-Hall, Brady Books, 1990. ISBN 0-13-583493-7.

  The Starter Kit will soon include specifications of Micro Channel chip sets.
- **Important visibility**. The association promotes Micro Channel architecture, its benefits, and the organizations involved. This promotion is done through international advertisements, product directories, participation at trade shows, and direct mail campaigns.
- **Toll-free hotline**. As a member, you can call the toll-free number (800) GET-MCDA for technical assistance and support between 9:00 A.M. and 5:00 P.M. Pacific time on weekdays.
- **Coming soon: MCDA bulletin board**. All MCDA members will be eligible to use the association's bulletin board to communicate with the MCDA and with other MCDA members. News, technical information, marketing information, and product information can be posted and retrieved.

The MCDA also encourages members to participate in development of the market for Micro Channel architecture products. The Micro Channel architecture market includes chip sets, subsystems, planars, adapters, cabinets, backplanes, and software.

MCDA Dues

Dues for member organizations vary according to their revenue. The current dues structure ranges from $1,000 per year for companies with revenue of up to $10 million, to $3,500 per year for companies whose revenue exceeds $100 million.

MCDA Organization

The MCDA executive board sets the strategic direction for the association. Board members are elected annually. The chairperson is elected every three years. The president is elected by the executive board for a term of mutually agreeable duration. Several committees have been established to set policies and processes for executing the mission of the association. The currently formed committees are Technical, Promotion, and Process. Finally, working groups handle issues, resolve problems, and propose changes.

For More Information

Contact the Micro Channel Developers Association at 2280 N. Bechelli Lane, Suite B, Redding, CA 96002, (916) 222-2262 or (800) GET-MCDA; fax (916) 222-2528.

Ramiz H. Zakhariya *has served as president of the Micro Channel Developers Association since 1991. He is currently on loan to the MCDA from NCR® Corporation. During his 12 years with NCR, he drafted next-generation architecture and system hardware specifications, and designed and developed Large-Scale Integration (LSI) and Very Large-Scale Integration (VLSI) chips. Ramiz has degrees from the University of Southern California in computer architecture and nuclear physics.*

Subscription Form for */AIXtra:* *IBM's Magazine For AIX Professionals*

The IBM RISC System/6000 with AIX platform is one of the fastest growing advanced UNIX workstations in the industry. You can find out why and get the inside story on IBM's AIX systems straight from the source with a subscription to */AIXtra: IBM's Magazine For AIX Professionals.* Detailed technical articles are written by the experts — people who design, develop, and support AIX and related products. */AIXtra* magazine covers AIX systems, software, networking, implementation, and more.

A single issue costs $12.95; you can subscribe now and receive a one-year subscription for only $55.00 **(Canada/ Mexico $80; other countries $100).** Just complete this form and either **fax it to (415) 948-4280 (please include VISA/Mastercard number and expiration date), or mail your check or money order to The TDA Group, Box 1360, Los Altos, CA 94023-1360, or call (415) 948-3140. All orders must be pre-paid. Checks must be in U.S. dollars drawn on a U.S. bank.**

1. Do you currently have AIX installed?
   Yes____ No____
   How many years experience have you had with AIX or UNIX?_____

2. If not, do you plan to install AIX within the next 24 months?
   Yes____ No____

3. Other operating systems in use:
   - ☐ UNIX Sys. V
   - ☐ SCO Unix/Xenix
   - ☐ Ultrix
   - ☐ A/UX
   - ☐ HP/UX
   - ☐ Sun/OS/Solaris
   - ☐ PC/MS-DOS
   - ☐ OS/2
   - ☐ VM or MVS
   - ☐ Windows
   - ☐ Mac

4. The number of employees at your location:
   - ☐ 1-99 ☐ 100-199 ☐ 200-399
   - ☐ 400-999 ☐ 1,000-1,999 ☐ Over 2,000

5. The number of employees company wide:
   - ☐ 1-99 ☐ 100-199 ☐ 200-399
   - ☐ 400-999 ☐ 1,000-1,999 ☐ Over 2,000

6. Applications in use:
   - ☐ CAD/CAE
   - ☐ Education
   - ☐ Science/Engineering/Research
   - ☐ Graphics
   - ☐ Point of sale
   - ☐ CASE/Software Development
   - ☐ Financial Services
   - ☐ Data Base
   - ☐ AI
   - ☐ Numeric Intense Computing
   - ☐ Manufacturing/CIM
   - ☐ Office/Professsional
   - ☐ Communications/Networking
   - ☐ Mapping (GIS)
   - ☐ Other

7. Are you involved in evaluating, recommending, specifying, or approving any of the following? (Check all that apply.)
   - ☐ Workstations
   - ☐ Terminals
   - ☐ X stations
   - ☐ Software development tools
   - ☐ Storage/memory
   - ☐ Printers
   - ☐ Training
   - ☐ Maintenance/service
   - ☐ Business applications
   - ☐ Graphic software
   - ☐ I/O boards
   - ☐ Modems
   - ☐ Comm./Networking
   - ☐ Servers
   - ☐ DBMS
   - ☐ Tape drives
   - ☐ DOS to UNIX tools
   - ☐ UPS

IMPORTANT: PLEASE COMPLETE ALL INFORMATION REQUESTED.
(Please print or type.)

Name _____

Company _____

Address _____

_____

City _____ **State** _____

Country _____ **Zip Code** _____

Business Telephone _____

☐ **Yes, please enter my paid subscription:**

Signature _____

Charge to ☐**VISA** ☐**Mastercard**
Number _____

Expiration Date _____

8. How much will your organization spend on computer related products/services in the next year?
   - ☐ Over $10 million
   - ☐ $1-$10 million
   - ☐ $250,000-$999,000
   - ☐ $100,000-$249,000
   - ☐ $50,000-$99,000
   - ☐ $10,000-$49,000
   - ☐ Under $10,000

9. Your primary job title:
   (Check the one best descriptor.)
   - ☐ Corporate and Financial Management (Pres., Owner, CEO, VP, Marketing Dir., Gen. Mgr., Financial VP, CFO, Controller, Treasurer)
   - ☐ Computer Systems Management (DP/MIS Dir., Network Dir., Communications Mgr., Systems Analyst, Software Developer, CIO, Systems Administrator)
   - ☐ Engineering Management and Staff (VP Engineering, Chief Engr., Technical Director, Systems Integration Mgr.)
   - ☐ Consultant and Educator (Computer/Network Consultant, Computer Technology Educator)
   - ☐ Other (please specify)_____

10. Your company's primary business activity:
    - ☐ Manufacturing (computer hardware)
    - ☐ Manufacturing (non-computer products)
    - ☐ Systems Integrator, VAR, OEM
    - ☐ Software Development
    - ☐ Financial: Banking, Insurance, Real Estate
    - ☐ Retail, Wholesale, Distribution
    - ☐ Utilities, Communications Services, Transportation
    - ☐ Government and/or Military
    - ☐ Computer Services, Data Processing, Service Bureau
    - ☐ Health or Legal Services
    - ☐ Education
    - ☐ Consulting
    - ☐ Agriculture, Mining, Construction, Petroleum, Forestry, Chemical
    - ☐ Architecture/Engineering
    - ☐ R&D, Testing, Evaluation Labs
    - ☐ Other qualified business including Hotels, Publishing, Amusements, and Non-Profit Organizations
    - ☐ Other (please specify)_____

Trackpoint II: The In-Keyboard Pointing Device

Ted Selker and Joseph Rutledge
IBM Corporation
Yorktown Heights, New York

Two of IBM's new ThinkPad notebook computers have keyboards that feature TrackPoint II, a unique, integrated pointing device. This novel design for a pointing device takes advantage of special behavioral motor research that gives the device unprecedented productivity improvements compared to using a mouse. This article describes TrackPoint II, how to use it, and the research behind its design.

Now there is a pointing device that occupies no space on a desk, increases productivity, does not have to be adjusted for left- or right-handedness, and has no moving parts. Called TrackPoint II, it is an innovative keyboard feature of IBM's newest notebook computer systems, the IBM ThinkPad 700 and 700C.

Without requiring users to move their hands away from the normal typing position, TrackPoint II provides smooth control when using popular graphical user interfaces that mix typing with pointing. It eliminates the loss of time and the distraction of reaching for another pointing device. TrackPoint II can be used equally well by either hand. All typing, pointing, selecting, and dragging now can be done on the keyboard. Users no longer need to carry an extra pointing device for their notebook computer systems, because TrackPoint II is built into the keyboard.

TrackPoint II is the first pointing device that outperforms a mouse for integrated typing and selection activities. In laboratory tests, it typically increases speed in tasks similar to text-editing by 25% compared to other pointing devices.

Using TrackPoint II

TrackPoint II, shown in Figure 1, consists of a small red rubber button that fits between a user's hands on the keyboard, and thumb-activated "mouse buttons" molded into the keyboard below the space bar. The red button is situated between the G and H keys, above the B key. It is positioned to be instantly available, yet it does not interfere with a touch typist's fingers. The user's hands can remain in the "home" typing position.

To use TrackPoint II, place an index finger on the red rubber button and press in the direction you want the cursor to move. The amount of pressure determines the speed at which the cursor moves to an icon, character, or even a pixel-sized target. You also can rest your finger lightly on the rubber cap without causing any action to take place.

To make a selection, press the appropriate "mouse button" (below the space bar) with either or both thumbs. It also is easy to press both buttons simultaneously with one thumb.

Dragging is just as easy. Hold the buttons down with one or both thumbs and use an index finger to control the motion. TrackPoint II works equally well for left- or right-handed people because either index finger can reach the rubber button while the thumbs use the mouse buttons.

Our research has shown that the act of taking one's hands off the keyboard, reaching for the mouse, and replacing the hands on the keyboard takes about 1.35 seconds. TrackPoint II eliminates this distraction and saves a substantial amount of time. Using TrackPoint II to make a single selection while typing saves 0.9 second.

You still can use your current mouse pointing device because both TrackPoint II and a mouse can be active. However, you quickly will notice that the TrackPoint II is faster for making a single selection while typing. Many users find that after a week of using TrackPoint II, a mouse becomes unnecessary for most work.

Collaboration

For collaborative work, such as two people working jointly on a writing project, it is helpful if both can point to words on the screen without interfering with each other. A unique feature of TrackPoint II enables two users to take turns controlling the cursor. In this scenario, a mouse is connected to a notebook computer; one user is using the mouse while the other is using the keyboard with TrackPoint II. With this collaborative feature, the first pointing device that moves takes precedence. The person using the cursor retains control until a moment after completing an action. Because the second user is temporarily locked out, that user cannot inadvertently interfere with the first user's cursor action.

Designing the TrackPoint II

Two important innovations have been designed into TrackPoint II: the ability to integrate typing and selection, and – most important – an algo-

rithm that allows quick, precise selections.

Experiments were conducted using secretaries and other typists to test the placement and usefulness of pointing devices within or near a keyboard. We tested joystick pointers that lay in various positions: under the entire keyboard, under specific keys, above and below the keyboard, between the G and H keys, near the number pad, and so on. We found that the space between the G, H, and B keys worked best. Our extensive testing has shown that TrackPoint II does not interfere with normal typing, and that typists do not inadvertently move the cursor.

Still, no one had ever made a joystick-like device that could perform nearly as well as a mouse-like device. A force-sensitive control is appealing whenever space is limited, and the psychology and human factors literature over the last century records repeated efforts to use such a control. However, it always was hard to use – it could be adjusted to be either sluggish or skittish, or even both at once, but there seemed to be no middle ground.

TrackPoint II results from new comprehension of behavioral and motor issues at play while using rate-control devices. For example, if the cursor moves faster than the eye can track, the user loses sight of it; if the unsteadiness of the user's hand affects cursor speed, movement becomes unpredictable. These psycho-motor realities influenced our design of a special algorithm for using a rate-control device to guide a cursor. The algorithm enables the cursor to move across the screen in a tenth of a second when the user pushes TrackPoint II relatively hard, yet it gives TrackPoint II a precise, firm feel when selecting icons, characters, and even pixels. The primary innovation here is the development of a joystick that gives users the feeling of smooth, positive control at all useful speeds.

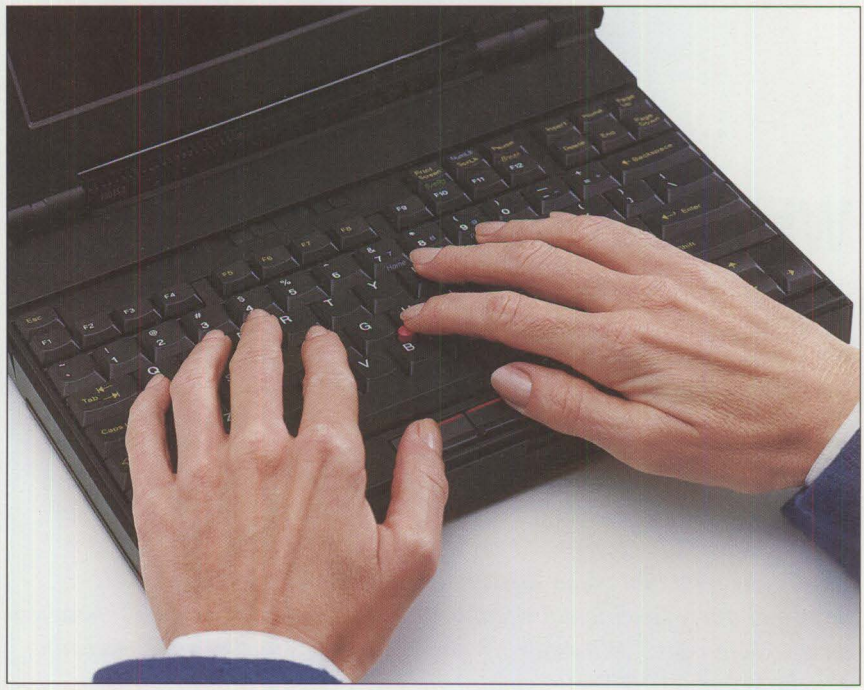


Figure 1. TrackPoint II

Ted Selker *is a research staff member at the IBM T. J. Watson Research Center in Yorktown Heights, New York, where he has been working for the last seven years on new paradigms for using computers. His recent successes include creating COACH, an adaptive interface that dramatically improves user performance, and the design of the TrackPoint II integrated pointing device. Ted previously worked at Xerox® Palo Alto Research Center (PARC) and at Atari's® Sunnyvale Research Center, and he also has taught at Stanford University. Ted holds a PhD from City University of New York.*

Joseph Rutledge *has been a research staff member in the Mathematical Sciences Department at IBM T. J. Watson Research Center since 1958, working mainly on the mathematical side of computer science, but with occasional forays into the "practical" world; TrackPoint II represents the most extensive and successful such excursion. He first met computers as one of the initial programmers of UNIVAC® I in 1950 and did logical design, systems architecture, and early programming language work before returning to school to acquire a PhD in mathematical logic from Cornell University.*

Why OS/2 2.0?

David Reich
IBM Corporation
Boca Raton, Florida

Wayne Caswell
IBM Corporation
Dallas, Texas

Martin McElroy
IBM Corporation
Basingstoke, United Kingdom

This is a reprint of a paper distributed by IBM during Fall 1992. It describes many leading-edge features of OS/2 2.0 and compares these features to Microsoft® Windows 3.1 and Microsoft's forthcoming operating system, Windows NT™. It demonstrates why OS/2 2.0, which was released in March 1992, is the operating system platform of choice and will continue to be after the availability of Windows NT.

Note: The discussion of Windows is based on information that Microsoft Corporation has made publicly available as of October 1, 1992 or information in the public trade press, and is subject to change.

Most people agree that, as an operating system, IBM's OS/2 2.0 is superior to Microsoft's Windows 3.1. To compete with IBM's OS/2, Microsoft has announced another system, Windows NT. Windows NT is not yet available and Microsoft says the first version may ship in late 1992 or in 1993.

When it finally arrives, Windows NT is expected to address some of Windows 3.1's shortcomings. However, based on the preliminary beta release and Microsoft's public comments, Windows NT will only partially close the gap with OS/2 2.0.

For example, the state of the art in user-friendly interfaces today is the object-oriented Graphical User Interface (GUI), such as the Workplace Shell™ in OS/2 2.0. Only recently has Microsoft begun to talk about releasing a similar user-friendly interface – sometime in 1994.

Today, OS/2 2.0 surpasses Windows 3.1 in the following areas:

- Superior crash protection
- Greater number of applications supported
- Superior multitasking
- Object-oriented GUI
- Superior file system
- More memory available for applications

Today, Windows NT is *not* available. In the time frame that Microsoft is expected to complete Windows NT, OS/2 will have moved forward significantly. The following enhancements are planned for OS/2 during the latter part of 1992:

- Additional performance improvements, especially for the minimum hardware configurations
- Support for more displays, printers, and other devices
- Improved graphics engine
- Support for Windows 3.1 applications

When the first version of Windows NT finally arrives, IBM is confident that OS/2 will still surpass it in the following areas:

- Compatibility with DOS and Windows applications
- Greater number of applications supported
- Object-oriented GUI
- Less expensive hardware requirements (memory and disk)

So, users can choose to live with the shortcomings of Windows 3.1 and wait for Windows NT to arrive. However, when they are finished with this wait, they may face a hardware upgrade and a conversion of Windows applications.

Or, users can enjoy the benefits of OS/2 2.0's superior operating environment, avoid the upgrade and the conversion, and still have a superior operating environment in the future.

Why OS/2?

The Best of Both Worlds

In the new PC environment, both personal productivity and line-of-business applications are essential. OS/2 can satisfy both needs. It provides a better DOS than DOS itself, and it runs a wide range of DOS and Windows applications. In addition, OS/2 2.0 is

a superior platform for running in-house mission-critical applications with industrial strength, robust protection, and powerful multitasking. Users do not have to choose between different systems for their different needs – OS/2 can do both.

Freedom of Choice

Today's computing environment can be confusing; the variety of options can be overwhelming. When making choices about hardware and software platforms, it is difficult to follow a path that keeps a wide range of options open. Too often, choices are constrained by compatibility issues or by a limited growth path. OS/2 2.0 aims to simplify the decision by providing a choice: the widest range of applications on a wide range of hardware.

OS/2 2.0 runs DOS, Windows, and OS/2 16-bit and 32-bit applications, the widest range of applications available on an Intel-based platform. In fact, OS/2 2.0 is such a superior environment that even if users run only DOS applications on a 386-based machine, OS/2 2.0 is the best environment in which to run them.

Furthermore, applications running under OS/2 2.0, whether they are DOS-, Windows-, or OS/2-based, provide added value by working together, sharing information, and running from the common Workplace Shell. This not only protects your current investment in DOS, Windows, and OS/2 applications, but adds value by integrating them.

In addition, OS/2 2.0, Extended Services for OS/2, and OS/2 LAN Server are supported on a wide range of IBM-compatible hardware as well as IBM PS/2s. This means the user can run OS/2 2.0 with confidence on machines from vendors like COMPAQ®, Olivetti®, Dell®, Hewlett-Packard®, Toshiba®, and others, and IBM support can be included. In fact, IBM has certified over 260 configurations from 71 hardware vendors, so it is highly likely that PCs equipped with an Intel 386SX or above processor are supported.

A Productive Environment for the User

OS/2 provides an object-oriented user interface, the Workplace Shell, which allows business users to focus on the information they want to work with, not on the application that needs to be loaded. This business-oriented way of working helps users become more productive by concentrating more on what they want to do, and less on how to do it. It also provides a single consistent environment in which multiple applications can be loaded from different sources. Additionally, it is an extremely easy environment to learn, since once users know how to drag a file's icon with the mouse to put it into a folder, they can use the same operation to print it, copy it to another disk, or erase it. In addition, companies can derive the benefits of a standard interface that complies with IBM's Common User Access™ (CUA™) definition for user interface design.

Also, since many applications can be loaded and running at the same time, users can be more productive, especially in work that involves much interruption and switching from one task to another. OS/2's true multitasking means that long-running processes can simply be switched to run in the background, while the user continues with something else – resulting in less "wait time" for the user. At the same time, more can be done with the existing set of applications by allowing them to share information easily through consistent interfaces like the Presentation Manager® clipboard.

A Platform You Can Rely On

When the PC becomes the center of information processing, as it often is in today's environment, then the PC platform must show the stability and reliability of the host environment. Today, DOS and extensions to DOS, like Windows, do not provide the protection that OS/2 2.0 offers. *OS/2 has been designed to protect applications from one another* and delivers today the stable platform required for full multitasking and greater protection from system crashes. It is of little use to have the most fault-tolerant server or host if the client workstations are not fault tolerant. And many users of productivity applications, like word processors and spreadsheets, consider their PCs to be mission critical. For this reason, reliability is a requirement for every PC.

Superior Connectivity

OS/2's strong multitasking and robust protection make it the best operating system available for connectivity applications such as client/server and distributed processing. In addition, OS/2 has Extended Services for OS/2, which provides communications and database functions, and the OS/2 LAN Server, which provides a full client/server environment. This allows networking to be an integral part of the operating system, and provides high functionality at a much more economical cost than buying many separate packages.

OS/2 is not only a superior server platform, but also the most functional and stable client. It provides a consistent platform for both server and client. It can handle multiple concurrent communications protocols – such as NetBIOS, Advanced Program-to-Program Communications (APPC), IPX, and TCP/IP – with ease, and even provides a LAN-independent user interface to mixed vendor networks. In addition, it is enabled for automated LAN-based installation. Most important, OS/2 offers the stability and reliability in a client to match the reliability of the server or host.

The result is that mission-critical applications that depend on communications with various systems can be implemented much more safely in OS/2 than on DOS or its extensions.

The Integrated System

OS/2 allows DOS, Windows, and OS/2 applications to run together, while providing a GUI and the database, communications, and LAN support included in Extended Services for OS/2 and LAN Server. For developers, this means the Application Programming Interfaces (APIs) and services have been designed to work together, eliminating the need for the systems integration of a variety of DOS-based packages, a process that often presents incompatibilities or problems.

The OS/2 function has been designed and tested to work together – IBM has already done the integration work. In addition, the Workplace Shell environment integrates DOS, Windows, and OS/2 applications and allows them to work together, even though they may have been written by different vendors. That's why OS/2 is *the integrating platform* for the 1990s.

32-Bit Power

OS/2 2.0 is a 32-bit system. It gives users the advantages of a 32-bit system: superior application performance and the opportunity to fully use the 386 and 486 hardware that runs OS/2. It provides users with a 32-bit system *now* – eliminating the need to wait for other alternatives with uncertain delivery dates.

The 32-bit API also allows developers to create richer, more sophisticated applications. Applications like multimedia require an advanced 32-bit interface to exploit their full potential and power. Additionally, moving to the OS/2 32-bit API gets developers ready for future developments in OS/2.

Platform for Growth

OS/2 will be the base of new developments for many features that will be requirements for the workstations of the mid-1990s. These include multimedia, object-oriented systems, support for the Distributed Computing Environment (DCE), and portability across different processors. These applications will require a robust, architected, and powerful 32-bit system, and that system is OS/2.

IBM plans to enhance OS/2's capabilities for object-oriented application development in distributed environments by advancing the function provided by the System Object Model (SOM). IBM intends to leverage a subset of Taligent's™ object services and frameworks to benefit OS/2 application development and enable future compatibility with Taligent's environment.

Value for Money

OS/2 2.0 offers a 3-in-1 environment, allowing users to run DOS, Windows, and OS/2 applications, so there is no need to buy DOS or Windows separately. It also includes a series of productivity applications, utilities, and games at no additional cost. OS/2 also provides scalable font support for both Windows and OS/2 applications with Adobe Type Manager®. OS/2 offers all this functionality at a list price that is less than the combined list prices of DOS and Windows 3.1.[1] Upgrading from DOS or Windows makes the cost of moving to OS/2 even less.

A Base for the Future

Today, OS/2 supports the widest choice of existing applications while meeting the needs of current client/server and networked environments. OS/2 also provides a strong base for future technologies and a reliable migration path. OS/2 currently offers what other environments can only promise for the future – so why wait?

Alternatives to OS/2

Windows 3.X

Microsoft Windows 3.0 and 3.1 are good attempts to work around some of the architectural limitations of the ten-year-old, 16-bit, single-tasking architecture of DOS. They offer the user a more attractive interface and provide an environment in which programs can be written to do limited multitasking. The underlying architectural limitations still remain, and it is these limitations that will prevent Windows 3.X from fully satisfying the demands of most users in the 1990s. Let us review these demands.

Reliability. DOS was written to run on the Intel 8086/8088 processors available at the beginning of the 1980s. These processors ran in real mode; that is, any program could address and change any part of memory. Therefore, any program that made a mistake could overwrite itself or the operating system. In any case, the program would fail. This may have irritated the user if it led to lost work, but the impact was likely to be small.

Windows enabled more than one program to run, but sometimes still ran the processor in real mode. In this situation, one failing program could necessitate the shutdown of the whole system. This results in the well-known Unrecoverable Application Error (UAE). In Windows 3.1, Microsoft reduced the frequency of the UAE and renamed the remaining UAEs to General Protection Faults (GPFs). However, *as long as a program runs on today's DOS, the potential for these failures remains*. These failures can be very irritating to users and can represent a real impact to their productivity. For businesses that run mission-critical or high-speed communication applications on PCs, it can be potentially disastrous.

From the beginning, IBM designed OS/2 to be a "protected" operating system. This means the operating system and the hardware cooperate to prevent failing applications from impacting any other part of the system. For the user, that means fewer problems and less inconvenience. For the business, it means lower risk and greater productivity.

Multitasking. Windows 3.X is built on the foundation of a single-tasking operating system – DOS. Therefore, multitasking of Windows applications must be done within the applications themselves. Programmers of Windows applications must explicitly include "yield points" to enable other applications to get a share of the processor time. This is called *cooperative application multitasking* and results in inefficient use of available resources, and unsatisfactory and uneven response to users when multiple programs are running.

IBM designed OS/2 to be a multitasking system by basing multitasking in the operating system, not the applications. For this reason, OS/2 can outperform Windows 3.X in many multitasking situations. In practice, this advantage is felt by the end user in the increased smoothness of response. For example, an OS/2 user can type into a word processor while formatting a diskette.

Application support. OS/2 runs more Windows applications than Windows 3.1 because it enables users to simultaneously run applications written for Windows real-mode (Windows 2.X applications) and Windows 3.X applications. (Windows 3.0 can run these applications, but not simultaneously with Windows 3.X applications.) OS/2 also will run OS/2 applications written for OS/2 2.0 and all previous releases of OS/2. An independent estimate put the customer investment in OS/2 applications at $2 billion, in addition to the $2 billion invested by software vendors.

OS/2 is the first mainstream 32-bit operating system for the Intel hardware architecture. Many software vendors and companies are developing applications that take advantage of the investment made in Intel 80386 and 80486 processor-based machines over the last several years. The second edition of the *OS/2 Application Solutions Directory* published by Graphics Plus, Inc. lists 1,100 32-bit OS/2 applications available or in development as of July 1992. OS/2 has the widest applications portfolio of any operating system on the market.

Networking. The role of the personal computer is changing; fewer business PCs are now stand-alone machines, and highly connected client/server architectures will provide the Information Technology (IT) systems of the 1990s. The original PCs were not designed to manage the demands of networking, which always required compromises for DOS-based PCs. The limited memory available for programs in DOS often meant that certain, larger applications were mutually exclusive with networking. Networking with Windows 3.0 was not always easy because of the various techniques used to circumvent the memory restrictions.

Windows 3.1 has helped ease these difficulties but has not completely eliminated the restrictions. In addition, the implementation of networking programs as Terminate-and-Stay-Resident (TSR) programs (which ran in the real mode of the Intel processor) further compromised the reliability of the system. Networking is fundamentally a multitasking activity and the limited multitasking in Windows was sometimes inadequate to manage high-speed communications tasks running in the background.

Since networks are increasing in size, effective network and systems management is becoming more important. A sophisticated multitasking system is required to ensure that these tasks can be safely performed in the background at any time without the intervention or knowledge of the user. OS/2 was designed to be part of a network; consequently, it is an ideal choice for a client workstation.

User interface. Windows introduced many users to the benefits of a graphical user interface. Research shows that the underlying conceptual model presented by a software system is as important as the actual look of the program. Windows is still harnessed to the same underlying organization as DOS. This necessitates that users understand the structure of the file system, the distinction between program and files, and so on.

The OS/2 user interface (the Workplace Shell) is a second-generation GUI that presents an interface modeled on the real world. Users interact with the system by manipulating objects, such as dragging a file to a

printer. IBM has conducted thousands of hours of usability research to ensure that OS/2 is easy to use, not just easy to learn.

The Workplace Shell also acts as a unifying layer for applications. Regardless of the system for which they were originally designed, they are used in the same way. Information can be shared between them using the same techniques. Printing is easier in OS/2, enabling users to forget about the mechanics of the system and simply accomplish their tasks. OS/2 is designed to work the way users work, not force them to work the way the computer works. Finally, OS/2 removes from many users the responsibility for understanding and controlling such things as extended memory management (provided by add-on products to DOS such as QEMM and enables them to concentrate on their jobs).

32-bit design. For the end user, the internal design of the system is probably not important. However, for the decision maker, the architectural basis of the product is significant because it dictates the range of future possibilities.

Microsoft has announced a 32-bit API for Windows 3.1 (Win32s), but it is important to understand the limitations inherent in this approach. As the full name (Win32 subset) implies, Win32s only implements some of the API calls in the full Win32 API, which Microsoft states is supported in Windows NT. This means that developers may have to make a choice. They can write an application common to Windows 3.1 and Windows NT (which cannot exploit the additional functions in Windows NT), or develop separate applications for Windows 3.1 and Windows NT. In the latter case, the benefits of the Win32s API will be limited to the flat 32-bit memory model (which a Win32s Dynamic Link Library will map back to the native 16-bit segmented memory model of Windows 3.1). The performance implications of this are unknown.

OS/2 implements a complete 32-bit API with advanced features today. The benefits of this increase as developers ship more advanced, high-performance applications for OS/2. The requirements of the 1990s are already here, and OS/2 can satisfy them today.

Windows NT

Microsoft has announced that it will provide a new operating system called Windows NT. It will share the Windows name and provide some compatibility to existing Windows programs. It has been announced for availability at the end of 1992 or early 1993. Currently, only pre-beta code is available, and this discussion is based on the functions present in this code and stated by Microsoft representatives to be in the plan. Keep in mind that *Windows NT is not a currently available product.*

Windows NT will implement several subsystems on a newly written kernel that borrows elements from different operating system models.[2] Microsoft states that important features of Windows NT will be as follows:

- Preemptive multitasking and multithreading
- Protected architecture
- 32-bit system
- Support for DOS and existing (16-bit) Windows applications

IBM agrees that these features are important, which is why they are already available in OS/2 2.0. Other features that Microsoft claims Windows NT will have include the following:

- Improved security API
- Support of Symmetrical Multiprocessing (SMP)
- Portability (easy migration to different hardware architectures)
- Portable Operating System for Computer Environments (POSIX)

IBM agrees that these features are likely to be of increasing importance in the future and intends to add these features to a future version of OS/2. However, it is unclear to what extent these features are required by customers today, or whether they will be more important than other technologies on which IBM is also working. In particular, the first version of Windows NT will not include any object-oriented user interface technology (unlike OS/2, which incorporates and uses the Workplace Shell/SOM as the basis of its object-oriented user interface).

When considering the value of a new operating system, it is better to take a business-oriented viewpoint rather than concentrating on the technology. Users should consider two vital points: the resources required to run an operating system and its compatibility with the existing application portfolio.

Windows NT System Requirements

The recommended minimum configuration for Windows NT will be a fast Intel 386 with at least 8 MB of RAM and 100 MB of disk space.[3] However, *PC Week* has reported "Many observers say that the practical recommendation will probably end up closer to a 12 MB system. Others predict even higher memory requirements."[4] The Gartner Group also has told its customers it believes "a mainstream platform for Windows NT will be a 486DX with 12 MB to 16 MB of RAM (and up) on the workstation."[5]

Since Windows NT is not generally available, it is unclear how much

memory will be required to run a typical networked application.

Windows NT Compatibility

Windows NT will be a break from previous PC operating systems and may not offer full compatibility with existing DOS or Windows applications.

In its July 27, 1992 review of Windows NT, *PC Week* stated, "Rather than provide compatibility for all DOS and Windows applications, Microsoft Corporation officials have stated their intentions to focus support on 'major' DOS and Windows 3.1 applications." Paul Muglia, a director of Windows NT at Microsoft, was also quoted, "We'll look at what are the top 100 Windows applications and the top 100 DOS applications, and focus more on those than on those that haven't sold well."[6]

In addition, the operating system design is processor independent; so if code written for the Intel 16-bit processors is to run on other processors, a software emulation of the underlying hardware may have to be provided. This technology is familiar from the UNIX world. It enables a basic level of compatibility, but has several potential drawbacks:

- **Performance**. The software emulation of hardware processes may cause applications to run slower.
- **Hardware-dependent programs**. These may often not run. In particular, many DOS device drivers may have to be rewritten. This means that fax, scanner, file backup, and even 3270 emulation programs may not run. Many software vendors will undertake the work of rewriting device drivers only if they are assured of a significant marketplace. The hardware requirements of Windows NT are likely to mean that it will not be a mass-market product.
- **Usability of DOS programs may be compromised**. Microsoft has acknowledged that in the first release of Windows NT, DOS programs using VGA (or higher mode) graphics cannot be windowed onto the desktop.[7] This is not a problem for OS/2. Microsoft's plans to support the clipboard and Dynamic Data Exchange (DDE) for these DOS programs also have not been made clear.

Windows programs written for Windows 3.X are 16-bit programs, and Microsoft has stated that Windows NT will support these programs in a single Virtual DOS Machine (VDM).[8] This means that if one program fails, other Windows 16-bit programs may fail – just as in Windows 3.1.

Windows NT Market Positioning

Windows NT may have several compatibility issues that could make it an unacceptable option for many end users. Add to this the projected higher cost of the hardware needed to run NT, and it is clear that Windows NT is unlikely to become the client of choice for most people. Microsoft also has clearly positioned Windows NT, as more suitable for a server or high-end workstation operating system.[9]

Although Windows NT has many of the features that would make it an attractive base as a server operating system, the reality is that changing a network operating system is a difficult and expensive procedure. Most network managers would choose to run with lower function rather than incur the risk and cost of changing server software.

Because nearly three-quarters of the networks in the world use Novell products that will not even run on Windows NT, it could take a long time for Windows NT to gain any significant acceptance. In addition, it is not clear what effect Microsoft's plans to bundle some basic networking functions with Windows NT will have on other networking product vendors' inclinations to support the platform.

OS/2 users will gain little if any benefit from moving to Windows NT because OS/2 already offers the key features of multitasking and application protection. In addition, Microsoft has stated that Windows NT will not run OS/2 32-bit or OS/2 Presentation Manager (PM) programs.

Many RISC-based workstation users are using UNIX because the specialized applications they need are written for UNIX. It is likely to be a large migration job to rewrite a UNIX program for Windows NT. In the absence of a large market acceptance, it is questionable whether software vendors will be willing to make that investment. Some UNIX users have already expressed their unwillingness to move to a new operating system that is inherently single-user when they are used to the flexibility of the multi-user UNIX. Jay Kidd, director of marketing at Silicon Graphics® (the manufacturer of the only RISC-based workstation that Windows NT runs on today), has stated, "UNIX, rather than Windows NT, will continue to be the operating system of choice for those who want the absolutely best performance and are willing to sacrifice compatibility to get it."[10]

In summary, Windows NT is at risk of becoming a high-technology demonstration piece. If it cannot run existing programs and needs more powerful hardware than is widely installed, then it should have a limited market and remain an academic solution to niche needs.

The Windows Client/Server Strategy

Microsoft has a two-operating-system strategy. Today, the company recom-

mends DOS and Windows for the client and OS/2 for the server.[11] When Windows NT is delivered, Microsoft says that customers should migrate their OS/2 servers to Windows NT servers. IBM believes that the reason Microsoft proposes two separate and different operating systems for the client and server roles is because Microsoft does not offer a product that provides the reliability and efficient multitasking for clients with more limited hardware requirements. *IBM proposes one operating system for both these roles: OS/2.* This reduces administration workload and training overhead for support staff while making better use of software developers' skills.

The dominant system design of the 1990s will be client/server. The flexibility, development speed, and cost advantages of this architecture increase the requirements for systems and network management. A reliable client is a must (why pay for fault-tolerant servers if the clients are not fault tolerant?), but true multitasking also is vital to enable effective and non-intrusive management. *OS/2 is an ideal client.* LAN Server with OS/2 on the server provides the highest performance server in the industry.

Windows Myths

Some claims and beliefs about Windows have gained popularity. They often do not stand up to closer examination.

Myth #1: The marketplace has chosen – Windows is the standard. Windows has been an impressive sales success with Microsoft claiming to have shipped 10 million copies. However, independent consultant groups – Creative Strategies and IDC™ – estimate that only 55% or 30% (respectively) of Windows licenses are in use. *Windows* magazine has also questioned Microsoft's number and estimated the number of copies of Windows in actual use at about 4.5 million.[12] Any of these independent estimates reveal 5% or less of an installed base of 100 million PCs are using Windows – far from being a standard.

Myth #2: Everyone is using Windows applications. Many software vendors have invested a lot of money developing Windows applications. As a result, much attention has been focused on these products. However, in 1991, the Windows applications market was smaller than the Macintosh® applications market (according to the Software Publishers Association). In the nine months prior to June 1992, there were never more than five Windows applications in the "Top 20" best-selling applications.[13]

Personal Computer Magazine, in May 1992, said "Companies that have invested a lot of money in developing Windows applications are battling for a small share of what is a small pie."

Users continue to use and buy the tried and trusted DOS applications, making compatibility with DOS applications a key requirement for any personal operating system. OS/2 excels at this, and this DOS compatibility is one of the areas that should be of greatest concern to users considering Windows NT in the future.

Myth #3: Windows is faster and leaner than OS/2. OS/2's design is optimized for multitasking, making OS/2 better than Windows in most multitasking scenarios. What is not well known is that OS/2 also can outperform DOS and Windows when running some DOS applications individually. OS/2 has a superior file system that gives a significant performance advantage to programs that do a lot of I/O, such as database programs. Microsoft has drawn considerable attention to the different minimum hardware requirements of DOS/Windows and OS/2. However, Windows can run in more than one mode. The Windows mode with the smallest hardware requirements offers the fewest benefits to users, such as more limited multitasking of DOS applications.

What Microsoft is Saying About OS/2 2.0

Microsoft has published several documents that compare Windows 3.1 and Windows NT to OS/2 2.0. Some of the titles include:

- *A Guide to Evaluating Microsoft Windows Operating System Version 3.1 for the PC Desktop with Comparisons to OS/2 2.0*
- *Microsoft Windows NT Operating System – A Technical Comparison with OS/2 2.0*
- *Microsoft Windows or OS/2 2.0?*

These documents from Microsoft contain many statements regarding OS/2 that are incorrect or could mislead users. To help IBM's customers make a more informed choice of operating systems, the following are clarifications to some of Microsoft's statements:

- **OS/2 will run on less than 2% of the Windows-capable machines.** Microsoft cites InfoCorp® as their data source. According to Microsoft's data, approximately 200,000 (1.38% of 18 million) machines can run OS/2. Microsoft's information is obviously incorrect, since there were over 1 million copies of OS/2 2.0 shipped in the first 120 days of availability.

  IDC has stated that at least 28% of the installed base of PCs can run OS/2. Almost 50% of machines shipping in 1992 and 66% of machines to be shipped in 1993 are OS/2-capable, signaling a trend in the marketplace. In addition, OS/2 can run on many of today's notebook and laptop computers.

- **OS/2 is not suitable as a network client because of the "relatively few native desktop applications available.**" OS/2, as the integrating platform, runs DOS, Windows, and OS/2 applications. No company has more experience and capability in networking than IBM. IBM believes OS/2 is the industry's best desktop client for connecting to complex enterprise networks. It is an ideal solution for mission-critical networked applications.
- **OS/2 has limited host connectivity based on the number of native communications packages**. That is not correct. The OS/2 Communications Manager has a very comprehensive set of host connectivity options, and current DOS- and Windows-based packages work on OS/2 as well.
- **Windows has more development tools than OS/2**. OS/2 has a full complement of more than 250 development tools, although Windows has more native development tools. Many of today's leading edge tools originated on OS/2, which is why OS/2 is the preferred development environment for many vendors.
- **Microsoft states that OS/2 runs multiple DOS applications by starting a Virtual DOS Machine. This is a feature of the 386 designed to support older real-mode applications, and this feature has been used for some time by a number of DOS extenders**. The reader might infer that this is a limitation or shortcoming in OS/2. This misses the point and could be misleading. Because OS/2 uses the hardware isolation that VDMs provide, OS/2 can offer superior crash protection. *Hardware* protects each application in a VDM from taking down an application or operating system in another VDM. Since Windows does not use this feature, the Windows Unrecoverable Application Errors and General Protection Faults (UAEs by another name) can and sometimes do crash the operating system and other applications.

  OS/2 also provides support for more DOS applications than is planned for Windows NT. Microsoft has confirmed that Windows NT will have limited support of DOS applications because it does not plan to support the Virtual 86 mode of the hardware the same way that OS/2 does. *PC Week* reported that many programs that support fax, scanner, Musical Instrument Digital Interface (MIDI), terminal emulator, and LAN cards (that today run under OS/2 2.0) will not run unmodified on Windows NT. In addition, DOS programs that support VGA or higher graphics will not run in a window on the Windows NT desktop.[14]
- **The new OS/2 Workplace Shell is difficult to use. Having Windows applications running on the OS/2 Desktop will confuse users and drive up support costs**. This argument is very difficult to understand, especially in our industry where new innovations are constantly bringing better products to consumers.

  The Workplace Shell represents a second generation of GUIs and is a major advance over the Windows and previous OS/2 interfaces. These older generation interfaces put a pictorial face on the menus of OS/2 1.X and Windows 2.0. Instead of working with operating system constructs like file managers and program managers, you work with a Desktop with pictures (icons) of familiar things, such as letters, folders, and appointment books. Instead of working with directories, paths, and print commands, you just pick up the picture of the letter and put it on the printer. OS/2 also allows users to preserve the command prompt or menu interface. IBM's OS/2 gives you the choice.

  Microsoft also has recently demonstrated a future (1994) Windows NT user interface, code-named "Cairo," that adds object-oriented functions to Windows NT. This bears a resemblance to the OS/2 Workplace Shell.
- **OS/2 2.0 does not run Windows 3.1 applications, which leads to deficiencies in that it will not use TrueType® fonts, and has limited networking support, performance, and reliability**. Support of Windows 3.1 applications in OS/2 2.0 has been demonstrated at various trade shows and is now in beta test with customers. IBM intends to make the Windows 3.1 application support generally available near the end of 1992.

  With respect to TrueType fonts, OS/2 2.0 offers built-in Adobe Type Manager font technology for both OS/2 and Windows modes. Adobe is widely used in the industry while TrueType is still proprietary. In addition, there are thousands more fonts available for Adobe than TrueType. TrueType support for Windows 3.1 applications also will be included in OS/2 in the near future.

  OS/2 currently provides more networking options than does any generally available version of Windows. OS/2's reliability and performance when performing many simultaneous tasks are hard to match. Several vendors, such as Novell, have networking products available for OS/2 2.0 today, with more coming from other vendors. In addition, OS/2 can run many DOS-based LAN products in its DOS sessions.

With OS/2's entry-level hardware requirements and its superior communications extensions, both from IBM and other vendors, OS/2 is ideally suited for both the client and server ends of communications, thus keeping all systems consistent and homogeneous.

- **The installation of OS/2 2.0 can be difficult**. Installing 15 to 20 diskettes can seem complex at first, but OS/2 does an admirable job of making it easy and of migrating existing applications. The installation process can be done across a LAN or eliminated entirely by choosing OS/2's remote Initial Program Load (IPL) capability. In addition, many new systems are preloaded with OS/2, and a CD-ROM version is planned for availability soon.

- **OS/2 2.0 offers limited reliability when running multiple Windows applications in the same session**. Actually, OS/2 has a big advantage over Windows 3.1 when it comes to reliability. Under Windows, an errant application can disable other applications or even Windows itself. OS/2 provides protection that can prevent a failing application from bringing down the whole system or another system.

  Under OS/2 2.0, if a user runs several Windows applications in the same session and two or more conflict, the user can simply specify them to run in separate sessions to protect one from harming the other. Of course this may use more memory, but the gain is the reliability that Windows 3.1 does not offer.

- **OS/2 2.0 has limited video support in that a WIN-OS2 window will only run in VGA graphics mode**. In the initial shipment of OS/2 2.0, this is true. However, there are Super VGA (SVGA) board makers who have already produced WIN-OS2 window (seamless window) drivers for their SVGA boards; IBM's 32-bit XGA and SVGA high-resolution seamless drivers are also available in the market.

- **Configuring OS/2 2.0 is difficult because users must configure both the OS/2 side and the Windows side of things**. Some users may want to customize the configuration of their Windows applications, but OS/2 is generally self-configuring. Once the user installs fonts and other tools, it runs seamlessly.

- **NT will be better in its support of 16-bit windows applications. NT will run these applications in one address space with parameter validation**. We disagree that this provides better protection. In contrast, it should provide no more protection than the current Windows version and still far less than OS/2 2.0.

  Since the applications will run only in one address space, they can still conflict with each other. The parameter validation in Windows 3.1 simply gives users a little more information on what went wrong. Windows can have difficulty recovering from such a situation and users may still have to reboot their system when a GPF or UAE occurs. There is no advantage in this.

  When a Windows application fails under OS/2, just stop and restart the failed session. There is no reason to reboot the entire system. Additionally, users have the advantage of running the applications in separate sessions to avoid conflicting with another application.

- **OS/2 falls short because it does not have a full 32-bit architecture**. In the current release of OS/2 2.0, the operating system code contains a mixture of 16- and 32-bit code. Because of the native support for DOS and Windows applications, 16-bit code must be present. The APIs provided, however, are full 32-bit implementations. This allows developers to write full 32-bit native applications and have total compatibility with OS/2 2.0 as more of the internal subsystems are migrated to 32-bit. In particular, a 32-bit graphics engine that will offer improvements in performance, function, and stability is already in beta test. IBM's intentions are to deliver this new graphics engine to end-users later in 1992.

- **OS/2 falls short because, as a mixed 16-/32-bit system, it cannot be ported to RISC processors**. This is incorrect. It is part of IBM's strategy to port OS/2 to the RISC platform and maintain compatibility with existing OS/2 32-bit applications. Only sections of OS/2 that are required to maintain compatibility with existing 16-bit DOS and Windows applications will remain 16-bit.

- **OS/2 does not have a desynchronized input model**. OS/2 has a mechanism to interrupt "ill-behaved" applications that might "hog" the message queue and inhibit user input. Most OS/2 applications are written so that this is not a problem.

  With OS/2's modular design, a desynchronized message queue can be implemented as a replacement subsystem and added to the system in the future.

- **OS/2 support for Windows applications is more limited in that it runs modified Windows 3.0, not 3.1, and will not run 32-bit Windows applications. These are shortcomings, given the size of the installed base of Windows.** First, there are no 32-bit Windows

(Win32) applications today. OS/2 can add this support if there is demand for it. As stated earlier, OS/2 has been demonstrated running Windows 3.1 applications. The code is in beta test now and is planned for availability near the end of 1992.

Finally, there is a fairly large installed base of Windows applications, and OS/2 2.0 runs virtually all of those Windows applications today.

- **There are only about 300 graphical applications available for OS/2.** Since OS/2 can run all the OS/2 and the majority of the DOS and Windows applications, most of the 6,500 Windows applications should be added to the list of what OS/2 will run.

  While these applications were not written to take advantage of OS/2's native protected mode, they will run well under OS/2 nonetheless. Windows 3.1 cannot run many of these applications without changes. In addition, Microsoft has published a compatibility list describing more than 30 applications written for Windows 3.0 that will not function properly on Windows 3.1, but run on OS/2 2.0.

  Following Microsoft's logic, Windows NT will be in the same situation as OS/2, in that the 6,500 Windows and thousands of DOS applications were not written for its native mode. Microsoft has also stated recently that it will focus only on support efforts on "major" DOS and Windows 3.1 applications for Windows NT.[15]

- **There are significant advantages to coding for the Win32 subset (Win32s) functions, to have code that runs and is portable up to Windows NT once Windows NT ships.** While this may appear to be a sound technical idea, there are some severe shortcomings in this approach.

  Applications coded only to the Win32s API will not exploit many advanced operating system features, such as multithreading or preemptive multitasking, on either Windows 3.1 or Windows NT. On the other hand, applications coded only to the full Win32 API may not run on Windows 3.1 at all.

  Essentially, the Microsoft strategy forces developers to make a choice: sub-optimize either the Windows 3.1 clients or the Windows NT servers, or maintain separate source libraries for each, significantly increasing development costs.

  OS/2, however, has a single, consistent 32-bit API for developers to build both client and server applications.

- **OS/2's scheduler will not preempt a time slice once it has been started while Windows NT will, leading one to conclude that OS/2 is less efficient for time-critical applications.** OS/2 is ideal for time-critical applications, and indeed, is being used in many sites today to control plant floors, loading docks, and medical equipment. OS/2 also was used at the 1992 Summer Olympic Games to control data and has been used to gather and report real-time data at the Indianapolis 500 car race for several years now.

- **Windows NT will support 2 gigabytes (GB) of address space per application while OS/2 2.0 only supports 512 MB.** OS/2's architectural limit per application is 4 GB; the current implementation is 512 MB. Today, there are very few applications that come anywhere near 512 MB of memory, and very few computers that even have 100 MB of real memory.

  Remember, the virtual memory limit for *any* system is its real (physical) memory plus all free disk space.

- **Windows developers cannot leverage the investments made in their Windows-based programs in OS/2.** In OS/2, Windows developers can gain great benefits and leverage their investments in Windows code in several ways:

  — Users can continue to run their Windows applications under OS/2 while developers work on OS/2 versions. OS/2 2.0 can run the majority of the Windows applications that Windows 3.1 does not.

  — Windows and OS/2 have several things in common. Many of the programming interfaces are similar, and in many cases, the structures and APIs are virtually interchangeable. Users who understand Windows programming will understand OS/2's Presentation Manager. Dealing with multitasking and multiple threads is something the user would have to learn for Windows NT and OS/2 2.0.

  — There are porting tools available today for the initial port from Windows code to OS/2. Many large applications can be ported in an hour or two. Then developers can begin to optimize the code for OS/2's advanced features.

  Once application code runs on OS/2, it will be able to run on future versions of OS/2. IBM has been able to maintain this commitment to protect customer investment in applications since OS/2 Version 1.0. Microsoft has forced developers to upgrade code with virtually every revision of Windows. Microsoft has already published a document on porting

Windows 16-bit applications to the Windows 32-bit APIs.

- **Windows NT can share printers and OS/2 cannot**. OS/2 can share printers with any of several network products available. It appears that Windows NT will have some networking features built into the base system. This can have advantages and disadvantages.

The advantage is that users will not have to purchase extra network products to use the most basic of networking functions.

The disadvantage is that users who do not want network functions are bogged down with the extra disk and RAM required to keep this code around. This may also limit compatibility with other vendors' networking offerings.

Summary Comparison Among OS/2 2.0, Windows 3.1, and Windows NT

Figure 1 compares key operating system features for Windows 3.1, Windows NT, and OS/2 2.0. Some of the entries under Windows NT are marked with an asterisk (*). This is because Windows NT is not generally available, and therefore IBM does not have the current specifications for all items. For the same reason, the data about Windows NT may change at any time.

Windows 3.1 Application Incompatibilities

When a vendor ships new software, minor incompatibilities often accompany the new function. Windows 3.1, for example, has problems running dozens of Windows 3.0 applications, including Microsoft applications. Support for Windows 2.X applications has been removed entirely.

OS/2 2.0 will run Windows 2.0 and 3.0 applications concurrently. It also will run nearly all of the 30+ Windows 3.0 applications that Microsoft warns will not run properly under Windows 3.1 and would require upgrades or fixes.

OS/2 2.0 Offers It All – Today

OS/2 2.0 is a fully preemptive, prioritized, multitasking, multithreaded operating system with a superior object-oriented graphical interface, networking, and host connectivity support, along with compatibility with most other software written for Intel-based PCs and compatibles. Best of all, it is available today.

The prioritized, preemptive multitasking of OS/2 utilizes the processor more efficiently than Windows 3.X. The connectivity support and its entry-level hardware requirements make it an ideal platform for both client and server computing.

OS/2 2.0 provides the following:

- 32-bit virtual memory, allowing applications up to 512 MB per application, limited only by the size of the user's hard disk
- Multitasking support, allowing many applications to run simultaneously with excellent performance
- Multithreading to allow those applications wishing to perform many simultaneous tasks to do so
- An easy-to-use and easy-to-program context-sensitive online help system
- Protection among applications and protection to enhance operating system integrity (Users have the option of running applications in separate sessions, or combining them as resources and as the situation dictates, while the operating system is protected from errant code.)
- Extendable subsystems, allowing programmers to add new system services and create custom, enterprise-wide applications while remaining flexible for the small company or home user
- International language support (currently 17 languages) including bidirectional languages for Hebrew and Arabic
- A state-of-the-art, object-oriented user shell that integrates applications with the shell, providing consistent interfaces across the entire system
- Compatibility (OS/2 will run 16-bit and 32-bit OS/2 applications, most DOS applications, most Windows 3.0 and Windows 2.0 applications, and soon Windows 3.1 applications.)
- Connectivity with various network systems along with host environments

OS/2 2.0's compatibility with applications written for previous versions of OS/2, DOS, and Windows is unsurpassed. Even Windows 3.1 will not run many applications written for Windows 3.0, forcing developers to update their code and users to purchase upgrades. OS/2 will run many of these applications, preserving users' software investments.

OS/2's programming interface has not changed from earlier versions. With any new functions that have been added, only minor changes are needed to source code to recompile on OS/2 2.0, and programs that ran on a previous version of OS/2 will run on OS/2 2.0 unchanged. The only need to recode for any upgrade of OS/2 is to take advantage of new features, again preserving programming investments.

IBM Multimedia Presentation Manager/2 (MMPM/2) has been released to provide multimedia capabilities for OS/2 systems for sound, CD-ROM, and MIDI support as well as advanced graphics.

	Windows 3.1	Windows NT	OS/2 2.0
Available	Today	*	Today
Price	$49 to $149	$400 to $500 (estimate)	$79 to $149
Applications Base			
DOS Applications	30,000+	*	30,000+
Windows Applications	5,000+	*	5,000+
16-Bit OS/2 Applications	0	*	2,500+
32-Bit OS/2 Applications	0	*	600
TOTAL	35,000+	*	38,000+
Hardware			
Processor	286 and higher [1]	386DX (33 MHz) and higher	386SX (16 MHz) and higher
Minimum RAM	2 MB	8 MB	4 MB
Recommended RAM	4 to 6 MB	12 to 16 MB	6 to 8 MB
Minimum Hard Drive (Approximately)	9 MB	40 MB	13 MB
Hard Drive for Full Installation (Approximately)	11+ MB [2]	100 MB [3]	28 MB
Largest Hard Drive	1 GB	17 Billion GB (NTFS)	64 GB (HPFS)
Largest File Size	1 GB	*	2 GB
SCSI Exploitation	No	Yes	Yes
File System Options	FAT only	FAT, HPFS, NTFS	Enhanced FAT or HPFS
Memory			
Virtual Memory Limit	4 x Physical Memory	2 GB per Process	512 MB per Process
Memory Model	Segmented (64 KB)	Flat Memory Objects	Flat Memory Objects
Multitasking			
Multitasking with DOS Applications	Time Slicing	Preemptive Time Slicing	Preemptive Time Slicing
Multitasking with Windows/PM Applications	Co-op	Preemptive	Preemptive
Priority	Static (Set by User)	Dynamic	Dynamic
Dispatchability	Process	Thread	Thread
System Services	Serial	Parallel	Parallel
Reliability/Protection			
Protection Between Applications	Limited	Some [4]	Yes
Kernel Protection	Limited	Yes	Yes
Remains in Protect Mode	No; Access to Real Mode is Possible	Yes	Yes

Figure 1. Comparison of Windows 3.1, Windows NT, and OS/2 2.0 (Continued)

	Windows 3.1	Windows NT	OS/2 2.0
Application Compatibility			
Multiple Concurrent DOS Applications	Yes (in Enhanced Mode Only)	Some [5]	Yes
Windows 2.X Applications	No	No	Yes
Windows 3.0 Applications	Most [6]	Some [5]	Most
Windows 32-Bit Applications	Some	Yes	No (Possible in Future)
Clipboard Support	Windows and DOS Only	Windows and DOS Only	Windows, DOS, and OS/2
DDE Support	Windows Apps Only	Windows Apps Only	Windows and OS/2 Apps
OLE Support	Yes	Yes	Yes
16-Bit OS/2 Applications	No	Partial (Character Mode Only)	Yes
32-Bit OS/2 Applications	No	No (Possible in Future)	Yes
Printing and Fonts			
Print Spooling	Limited [7]	Yes	Yes
Adobe Type Manager Standard	No	No	Yes
Network Printing Support	Some	Yes	Yes [8]
Background Printing Performance	Unpredictable	*	Predictable
National Language Support			
Number of Language Versions	12	*	17
Data Interchange	SO8859 (Different from DOS)	*	CP850 (Consistent Throughout OS/2)
Host Connectivity	Third Party	Third Party	Included in Extended Services for OS/2
Code Page	Single	Unicode	Selectable
Other Factors			
Full 32-Bit APIs	No	Yes	Yes
Concurrent High-Speed Communications	Unreliable	*	Yes
Background Communications	Unreliable	*	Yes
OEM Hardware Support	Yes	Some [9]	Yes
Development Tools	Yes	Yes	Yes
Command Language	.BAT	.BAT, BASIC	.BAT, .CMD, and REXX
Installation Migration for Existing Applications	Limited	*	Yes
User Interface			
CUA Compliance	Graphical Model ('89)	Graphical Model ('89)	Workplace Model ('91)
Icons Representing Non-Loaded Files on Desktop	No	No	Yes

Figure 1. Comparison of Windows 3.1, Windows NT, and OS/2 2.0 (Continued)

	Windows 3.1	Windows NT	OS/2 2.0
Place Icons Anywhere on Desktop	No[10]	No[10]	Yes
Group Windows	Single-Layer Only	Single-Layer Only	Multi-Layer
Customize GUI Look	No	No	Yes (Workplace Shell, Windows 3.X, OS/2 1.X)
Context Menus	No	No	Yes
Object Management	No	No	Yes
Graphical Installation	Yes	Yes	Yes
Intelligent Fonts	Yes (TrueType, 650 Fonts)	Same as Windows 3.1	Yes (Adobe Type Manager for PM and Windows, 1,200 fonts
Long File Names	No	Yes	Yes
Applets	Yes	Yes	Yes
Consistent GUI Logon	No; Requires Network Vendor Utility	Yes	Yes
Interactive Tutorial	Yes	*	Yes
Command Reference	No	*	Yes
Advanced Connectivity [11]			
Client and Server Platform	No	No [12]	Yes
Multiple Concurrent Protocols	Limited	Yes	Yes
SNA LU6.2	Third Party	Third Party	Yes
APPN	Third Party	Third Party	Yes
TCP/IP	Third Party	Third Party	IBM TCP/IP for OS/2
Systems Management	Third Party	LAN Manager (Future)	Various from IBM
SQL Server	MS SQL Server	SQL Server NT (Future)	OS/2 Database Manager (Requires OS/2)
SQL Client	Third Party	Yes	Yes
NFS	Third Party	Third Party	IBM TCP/IP for OS/2

Notes

[1] Although Windows 3.1 will run on a 286, this limits the features available to the user (multitasking DOS applications, demand paging, 32-bit support).

[2] An additional 50% of the remaining partition is used for the swap file. This is the default.

[3] This includes a mandatory 20 MB swap file.

[4] Windows NT will run existing Windows 16-bit applications in a single address space. If one of these applications goes down, all applications in the address space could go down as well.

[5] Windows NT has been shown to have compatibility problems with some classes of DOS and Windows applications. See *PC Week*, July 27, 1992.

[6] Windows 3.1 will not run some Windows 3.0 applications, which will need updates. Compatibility notes are listed in the `APPS.HLP` file. Several Windows 3.0 applications need updated versions to run on Windows 3.1. OS/2 2.0 runs virtually all Windows 3.0 applications, as well as all Windows 2.X applications that Windows 3.1 will no longer support (because no real mode support is provided).

Figure 1. Comparison of Windows 3.1, Windows NT, and OS/2 2.0 (Continued)

Notes (Continued)

[7] Print spooling is provided by Windows 3.1 for Windows applications only, not for DOS applications. OS/2 2.0 provides print spooling for DOS, Windows, and OS/2 applications. OS/2 2.0 has extensive user print management capabilities (40 APIs versus 12 APIs in Windows 3.1) for querying, holding, releasing, and deleting jobs (including a graphical view of job and queue status).

[8] OS/2 has been shown to outperform Windows 3.X with background print operations in multitasking environments.

[9] Early feedback on CompuServe® of the pre-beta Software Development Kit (SDK) indicates that 386 processors with a B0 or B1 stepping are incompatible with Windows NT. Several common BIOS chips also have been found to be incompatible.

[10] In Windows, files exist only in the File Manager, programs only in Program Manager, and so on. There are no icons for printers.

[11] OS/2 2.0's "Yes" answers here are all using Extended Services for OS/2 except where stated. The Windows column refers to Windows-specific programs (programs written explicitly to take advantage of Windows GUI, memory addressability, or time slicing). Although there are many DOS connectivity options, and although they may be usable under Windows, the integration of these complex subsystems and any co-residency of two or more options (for example, TCP/IP and SNA) is completely the responsibility of the user, as a custom integration effort.

Moreover, Windows on DOS has architectural limitations – less memory, less protection, and less multitasking support – which make multiple network connections more difficult to integrate than under OS/2. OS/2's base environment provides tools and system support designed to enable this type of multiconnectivity installation. Also, all extra software required for these functions under OS/2 comes from IBM, and therefore, one can anticipate a greater degree of integration.

[12] The projected system requirements for Windows NT may be too large for many of today's client machines.

Figure 1. Comparison of Windows 3.1, Windows NT, and OS/2 2.0

Many applications have already taken advantage of OS/2's powerful multitasking and multithreaded features in their 16-bit versions. Vendors such as Lotus®, DeScribe®, Aldus®, and Novell have 16-bit OS/2 applications. The 32-bit applications will, in most cases, run even better and faster because of OS/2's new 32-bit flat memory model along with its other features. There are more than 200 32-bit applications available now. More than 1,000 software vendors have committed to delivering 32-bit OS/2 applications in 1992.

OS/2 2.0 offers users and developers alike powerful multitasking features, with limitless possibilities for the future. Best of all, OS/2 2.0 is available on the desktop today.

References

[1] On October 1, 1992, the suggested retail prices of MS DOS 5.0, Windows 3.1, and OS/2 2.0 were $99.95, $149.00, and $149.00 respectively.

[2] "A Grand Tour of Windows NT," *Microsoft Systems Journal*. July/August 1992.

[3] *Microsoft Windows NT – An Overview*. April 1992.

[4] *PC Week*, Windows and OS/2 Supplement. August 17, 1992. Page S1.

[5] Gartner Group. *Personal Computer Research Notes*. (P-230-853). July 31, 1992.

[6] *PC Week*. July 27, 1992. Page 1.

[7] *PC Week*. Windows and OS/2 Supplement. August 17, 1992. Page S9.

[8] *Microsoft Windows NT Operating System – A Comparison with OS/2*.

[9] "Microsoft Operating Systems Directions," presented by Dwayne Walker at COMDEX®, Spring 1992.

[10] *Windows* Magazine. October 1992. Page 20.

[11] *Microsoft Windows Strategy – An Overview*. Page 5.

[12] *Windows*. October 1992. Page 16.

[13] Data from Romtec, Ingram-Micro D®, Software Unlimited, and PC Connection®.

[14] *PC Week*. July 27, 1992. Page 1.

[15] *PC Week*. July 27, 1992. Page 1.

David Reich *is a staff programmer in OS/2 Customer Support in the Personal Systems Programming Laboratory, Boca Raton. He has been involved in OS/2 development for the last five years. David has just completed a book titled* Designing OS/2 Applications, *soon to be released. He has an MS in computer science from the State University of New York at Albany.*

Martin McElroy *works in OS/2 Technical Marketing in IBM's European Personal Systems Center, Basingstoke, UK. His responsibilities include OS/2 product positioning, strategy, and competitive analysis. Martin has been working on OS/2 since he joined IBM in 1990. He has an MA (with honors) from Lincoln College, Oxford.*

Wayne Caswell *is a senior marketing support representative and OS/2 strategist for IBM's southwestern area. The last eleven of his 23 years with IBM have been spent with end-user computing, even before the PC. Wayne has a BA in management technology from American University.*

Did You Know...?

1 Advanced applications such as object-oriented programming, multimedia, and distributed computing require the versatility and reliability of OS/2 2.0.

2 OS/2 2.0 is the first workstation operating system to fully exploit the features of the 386/486 family of processors.

3 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... 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.

**Discounts available for multiple copies.*
***California residents add applicable sales tax.*

☐ Charge to my VISA/MasterCard ☐ Check ☐ Money Order

Name: _____

Company: _____

Address: _____

City/State/Zip: _____

Phone: _____

VISA/MasterCard #: _____ *Expires:* _____/_____

Make checks payable to: The TDA Group, P.O. Box 1360, Los Altos, CA 94023. For faster service, order toll-free by calling (800) 551-2832, or fax this form to (415) 948-4280. All orders must be prepaid.

OS/2 Distributed Systems Management

Jim Alexander
IBM Corporation
Austin, Texas

This article provides insight into IBM's strategies and plans for managing sophisticated OS/2 LAN system environments through industry standard interfaces. The term used to describe this strategy is OS/2 Distributed Systems Management (DSM). The article describes how systems management for OS/2 has evolved and puts future plans into perspective. It also positions OS/2 LAN management within IBM's overall strategy for enterprise-wide systems management.

OS/2 2.0 is the platform for creating powerful systems management applications. The products that implement IBM's distributed systems management strategy are a suite of applications for managing many resources. An open system framework allows vendors and customers to add applications and agents to manage other resources, complementing those supplied by IBM.

These products provide the systems management solution required for LAN-based systems and network management. With other SystemView™ host procedures, they provide the best management solution for the entire enterprise.

In systems management, the word *management* means the ability to monitor a system's resources and to take appropriate actions when necessary. To monitor a system resource, one must be able to query, retrieve, and examine its attributes, such as displaying the configuration of OS/2 2.0 on a workstation. In addition, one must be able to recognize changes in resources, such as OS/2 2.0 memory usage. The monitoring may be periodic or on an exception basis (where the resource notifies the manager of a changed condition or threshold reached). In the memory usage example, the manager may wish to be notified if memory usage exceeds 90% for a server or workstation.

Another aspect of management – taking appropriate actions – may involve changing the resource, adjusting its usage limits, taking it out of use for corrective service, or a host of other actions that depend on the circumstances (such as terminating an OS/2 2.0 task that is using excessive amounts of memory).

In its simplest form, systems management can be defined as the ability to monitor and control information processing resources, including both hardware and software. Monitoring is done by collecting information from resources; control is done by sending information to resources. Such information is commonly called *management information.*

There are several common questions about systems management: What hardware and software are installed? Where? Who uses it? How is it used? How do I identify and fix problems? When do I need to install additional servers or faster processors? How large a support staff do I need? How can I use budget and resources more efficiently?

LAN Systems Management

In 1992, Benchmark Research Ltd. and Client-Server Technologies released these statistics illustrating the size of this task:

- 82% of LANs use three or more protocols
- 44% support multiple network operating systems
- 32% doubled the number of network nodes during the past year
- 62% already consider their network support staff too small

Considering the spectrum of current application areas, systems management tools probably provide one of the highest paybacks for LAN installations. With the growth of LANs, attached personal systems, and the distribution of mission-critical applications to remote LANs, the costs to administer these resources are becoming very high.

The largest paybacks usually occur because users save time. Users, often involved in diagnosing and solving network and system problems, are nearly always affected by system outages.

It is costly to have trained personnel at each remote site to handle such problems. Often there is not sufficient work to justify full-time support, but the work is too technical to be done by users at that site.

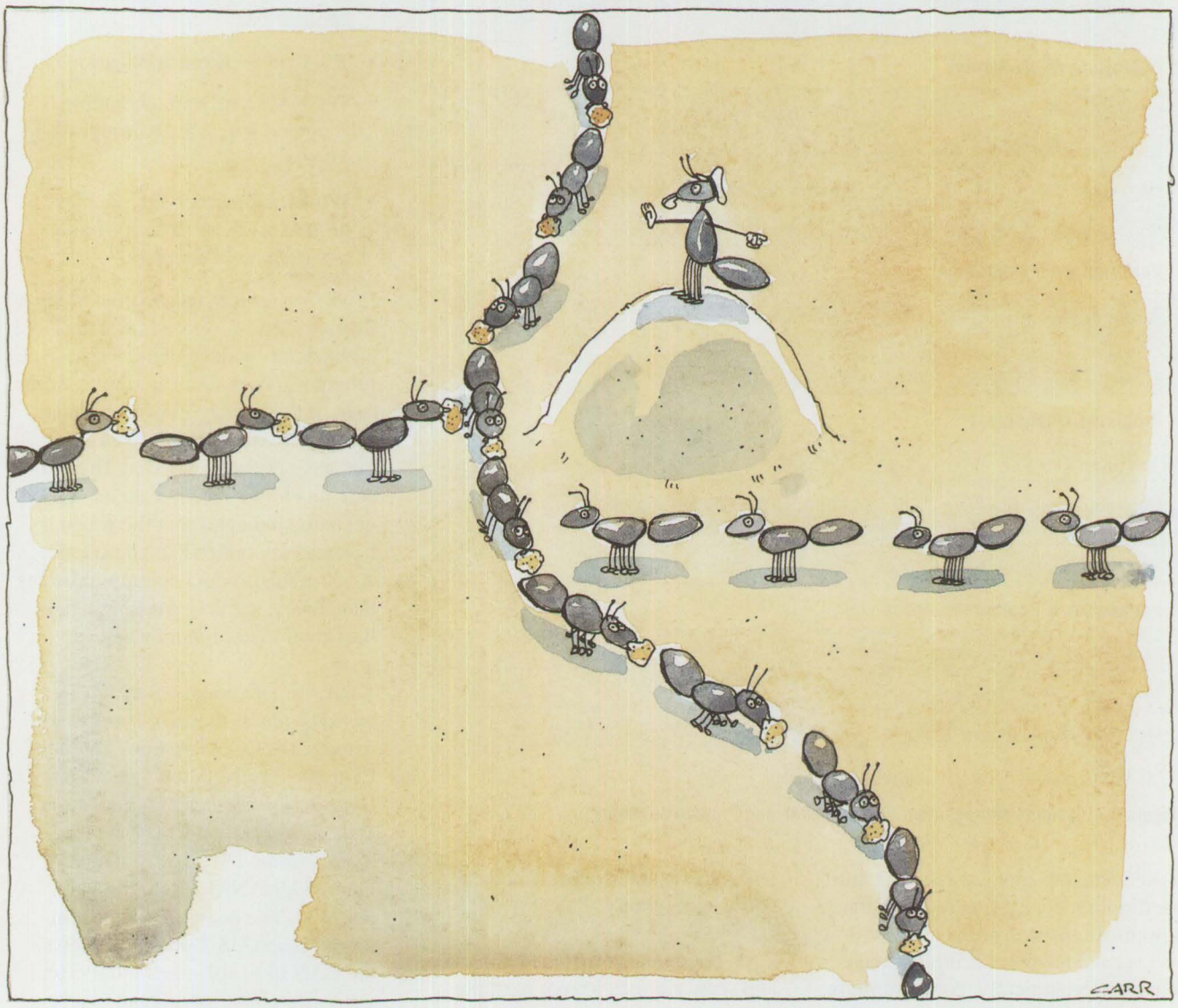
Another approach is to use outside consultants to support the LANs. Since consultants are normally not on site, they add time and expense for travel to the cost of diagnosing the

problem. These consultants may nevertheless be cost-effective when compared to permanent personnel.

Both users and central-site support personnel are often required for diagnosing and repairing problems. Often central site technicians enlist the help of users or consultants in diagnosing problems and applying fixes. For very complex problems, central site personnel may need to travel to remote sites. This adds to the time and cost involved in correcting the problem and takes the technicians away from their normal duties.

Error detection and correction are not the only focus. Maintenance and administration of LANs are also candidates for substantial savings. Maintenance involves adding function to existing systems or applications, upgrading systems and applications to current levels, and adding or moving users on a LAN. Often maintenance tasks must be applied concurrently to all workstations on the LAN, and may also require notifying users about the changes. Adding a user may involve changes to servers and gateways on the LAN, as well as changes to host systems. Maintenance requirements, therefore, also involve users, consultants, or central site support personnel.

No matter what the source, maintenance of LANs can be costly. LAN administration, particularly remote administration, can be a source of justification for systems management applications. LAN administration involves day-to-day operation, monitoring, and tuning. Administrators require access to servers, gateways, user workstations, bridges, routers, and other resources. Monitoring operations requires a constant view of the



Management Element/Requirement	Products
Business Management	
Access Control and Security	OS/2 LAN Server, Database Manager, LAN Network Manager (LAN/NM)
License Management	X[1]
Change Management	
Software Distribution	NetView Distribution Manager/2 (NvDM/2)
Automated Installation	NvDM/2, LAN Enabler
Configuration Management	
Multi-System Configuration	X
Topology Management	LAN/NM, LAN Station Manager (LAN/SM)
Network Asset Management	LAN/NM, LAN Management Utilities/2 (LMU/2)
Operations Management	
LAN-Based Automation	LAN/NM, LMU/2
Remote Command Facility	Extended Services, LMU/2
Help Desk	Distributed Console Access Facility
Backup/Restore/Archive	Sytos Plus® File Backup Utilities
Performance Management	
Resource Monitor	System Performance Monitor/2, LAN/NM, LMU/2
Capacity Planning Aids	Parameter and Tuning Guide
Problem Management	
Alert Support	LAN/NM, LMU/2
Common Dump and Error Log	First Failure Support Technology/2 (FFST/2), OS/2 SNAPDUMP
Remote RAS Services	X
Framework & Common Services	
Managing System Services	X
Managed System Services	X
Discovery/Topology Services	LAN/NM, LAN/SM

[1] X: IBM product currently under development

Figure 1. Systems Management Requirements and Product Solutions

LAN, and any changes in status must be highlighted. Tuning requires the administrator to examine current values and have the ability to make changes dynamically. At a minimum, operation involves the ability to stop and start system hardware and software remotely.

LAN components include hardware and software (often from multiple vendors), business applications created within the enterprise, and the users. All need to be managed. Practicality dictates that a minimum set of resources must be managed, including the following:

- The operating system on each workstation or server, commonly OS/2, DOS, or DOS with Microsoft Windows
- The network operating system, such as OS/2 LAN Server and NetWare
- Applications, including IBM Communications Manager, IBM Database Manager, other vendors' applications, and user-written applications
- The data for these applications
- The LAN hardware, including adapters, wires, hubs, bridges, and routers
- Workstation hardware, such as processors, hard disks, printers, and peripheral adapters

IBM's systems management strategy, SystemView, defines several disciplines for management. Figure 1 provides requirement examples from each of the SystemView disciplines. It also lists specific OS/2 solutions that have been provided by IBM to date to address some of them. A product name indicates that the product has been announced and delivered. An X in the product column indicates that IBM is actively working to add the function to product plans.

There are, of course, many host-based products that also support and manage the OS/2 LAN environment. Examples are NetView®, NetView Distribution Manager, Workstation Data Save Facility/VM, Data Facility Distributed Storage Manager, LANAO, LANRES, and PC Support/400. The combination of host products and OS/2 products allows the LAN to be managed more effec-

tively when a host computer is present in the enterprise.

The Need for a Single Comprehensive Solution

Systems management applications available today address many management requirements for enterprise LANs. They provide varying levels of management for selected resources on the LAN. Their choice of protocols also varies. The following types of offerings are currently available from IBM and others:

Network Managers: These focus on LANs and Wide Area Networks (WANs). IBM NetView/6000, announced in January 1992, is such a product. It facilitates network management in a multivendor Transmission Control Protocol/Internet Protocol (TCP/IP) network. It works with TCP/IP devices that include the Simple Network Management Protocol (SNMP) agents, such as the IBM 6611 Network Processor, and monitors all IP-addressable devices. This product and many others like it use SNMP as a resource management protocol because networking devices from many vendors provide agents that use the SNMP protocol.

Other major vendors have chosen to support the Open Systems Interconnection- (OSI-) endorsed protocol, Common Management Information Protocol (CMIP), as the basis for network management. They chose the International Standards Organization- (ISO®-) standard CMIP because of its richer management protocol. These vendors also support SNMP where required.

Server Managers: These systems manage file, print, or mail servers on LANs. The servers include the NetWare family, IBM LAN Server, and others. Most focus on LANs and do not implement the SNMP protocol for managing resources. A private protocol that exceeds the capabilities of SNMP is often used to better suit the server management environment and to not require the TCP/IP stack as a prerequisite.

Although network managers and server managers are specialized management offerings, they have a significant amount of management infrastructure in common. A single framework to manage the entire enterprise LAN (including networks, servers, and clients) would be the ideal solution – one with a common infrastructure and common services. It would support multiple protocols and APIs for all resources on the enterprise LAN.

OS/2 systems management has thus far been provided by two means – functions integrated directly into the operating system, and by separate, self-contained products that address a specific aspect of managing OS/2 LAN systems.

OS/2 integrated systems management functions help with installation, corrective service, configuration tables, RAS facilities, access control, performance hooks, alert generation and routing, and remote operations.

The self-contained LAN systems management products provided by IBM and other software vendors are extremely useful in addressing portions of the systems management challenge. As a group, however, they also have their shortcomings:

- Unique user interface for each application
- Little or no shared data
- Same data represented in different ways
- Duplication of user-supplied data
- Duplication of function between applications
- Environment- or platform-specific
- Integration left to users
- Little or no interaction between applications

While the systems management strategy for the OS/2 LAN environment includes exploiting some current technologies and products in the short term, the provision of an architected management platform that adheres to industry standards from organizations such as ISO, Institute of Electrical and Electronics Engineers (IEEE®), X/Open™, and the Open Software Foundation® (OSF®) is the primary goal. This goal is achieved through IBM's family of distributed systems management products.

Distributed Systems Management

IBM's strategy for OS/2 Distributed Systems Management is to provide a framework that supports systems management applications, and a set of applications that, in turn, supports LAN resources. This support is in accordance with the SystemView structure that will be used on all of IBM's Systems Application Architecture® (SAA™) platforms. It includes the use of many industry-standard definitions and interfaces, allowing vendors to produce integrated, yet portable, applications on these platforms. IBM's distributed systems management family supports SystemView-selected industry standards as well as other industry standards and technologies.

Why are these industry standards so important in the Distributed Systems Management (DSM) strategy? An open solution provides many benefits critical to DSM's success:

- It allows DSM to manage a broader set of standards-conforming resources.
- It promotes development of systems management applications for DSM by third-party software vendors.

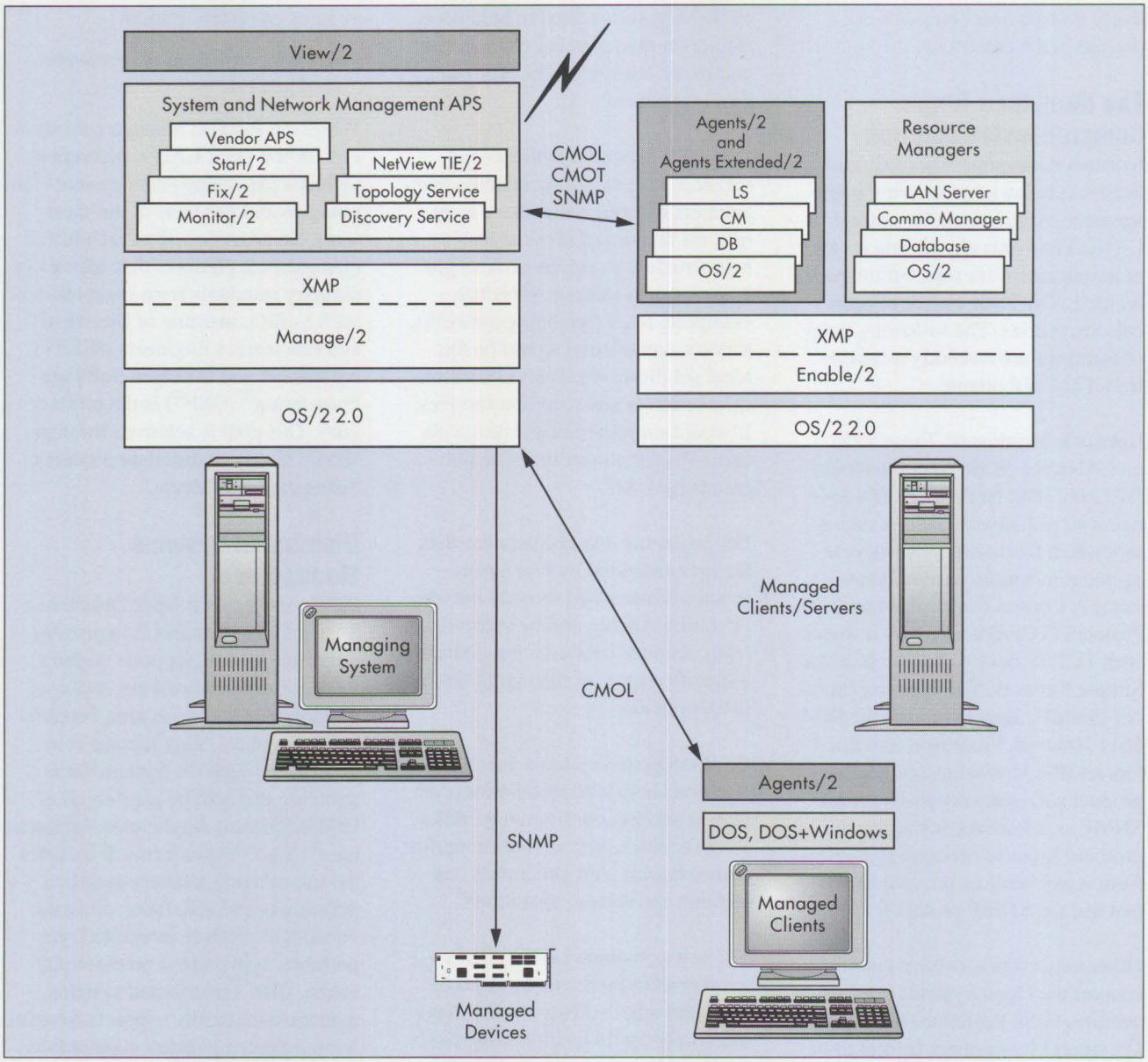


Figure 2. Distributed Systems Management Structure and Resources

- It allows integration and cooperation between applications.
- It allows OS/2 to be managed more easily by other managers.
- Its common platform services allow application developers to focus their energies on developing applications that have added value.

Standards Used in DSM

The primary industry standards incorporated in SystemView and OS/2 DSM are the ISO Open Systems Interconnection specifications. The X.700 family of standards, including Common Management Information Services (CMIS) and CMIP specifications, have been selected from OSI. The X.700 family provides the framework and a comprehensive resource and object definition, especially for system resources. It includes the OSI Guidelines for the Definition of Managed Objects (GDMO) as the way to define the SystemView objects in the data model. These guidelines are used to define the DSM objects. The DSM objects will be registered with SystemView and other applicable standards organizations.

Secondary sources of standards and technology for OS/2 DSM are OSF and X/Open. IBM's workstation development organizations are active participants in both groups. In September 1991, OSF selected the technology for the Distributed Man-

agement Environment (DME). The technology had been requested to provide management for the Distributed Computing Environment (DCE). The DME technology, like the DCE, is intended to work across multiple platforms. IBM and Hewlett-Packard submitted, and the OSF selected, technologies to address the requirement for an open systems management platform for application development (HP® OpenView™) and for object manager support (IBM Data Engine). One technology was submitted collectively by Group Bull, Hewlett-Packard, and IBM to both the OSF and X/Open groups, and it was selected by both groups. OSF calls this technology the Consolidated Management API (CM-API), and X/Open calls it Management Protocol (XMP) API. These are identical and provide a common API for the industry to access systems management services.

The OSF/DME technology, which was designed for workstation platforms that use OSF/1® operating systems, is also useful in other 32-bit operating systems such as OS/2 2.0 and AIX®.

Another important source of industry standards comes from the Internet Architecture Board (IAB). The IAB standards contain the specifications for SNMP and the SNMP agents for managed objects (MIB I and MIB II). IBM's distributed systems management family of products supports these standards transparently through the XMP API. Resource agents conforming to the SNMP specifications can be managed by DSM.

IBM's Distributed Systems Management Family of Products

IBM's distributed systems management family of products, with its CUA-91-compliant, object-oriented, graphical user interface can be used by administrators to define and manage LAN network resources from a central location. Network topologies are created by dragging and dropping icons that represent workstations into a topology drawing area, customizing them with the desired operating system and subsystems, and drawing the needed connections to other nodes (such as a LAN Server Domain Controller). The topology can be stored either in an SQL database or in a plain text file.

The implementation of the DSM strategy on OS/2 requires developing a "managing system," and managed system services and resource agents. Figure 2 shows a high-level view of the OS/2 2.0 systems management platform structure and managed resources supported by the distributed systems management family of products.

The product family includes a set of products that form the strategic framework upon which the systems management functions can be built. These framework products provide the common infrastructure, services, and support elements that:

- Create the "managing system" environment on OS/2 2.0 in which management applications are built
- Create the "managed system" environment in OS/2 2.0, DOS 5.0, and DOS 5.0 with Windows 3.1 that allows resource agents to manage system resources
- Provide the resource agents to manage the operating system resources for the supported systems and the OS/2 subsystem resources for LAN Server, Communications Manager, and Database Manager

Using these framework services simplifies the task of creating systems management applications and resource agents. Developers can focus on delivering valued-added products, not on duplicating these common elements. It is also easier to achieve greater consistency and interoperability between applications and agents. Some common elements of the set of framework products are as follows:

- Common user interface services
- Communications services supporting multiple protocols and transports
- Event and metadata services
- Common programming interfaces
- Discovery and topology services for LANs

The following products form the framework for management applications:

- **Manage/2**: This is a set of common services upon which to build management applications. The open industry standard programming interface, X/Open Management Protocol, is supported. The framework's common services include protocol support for CMIP and SNMP. SNMP devices (such as bridges and routers) can be managed, as can the CMIP-based resource agents provided. Topology and Discovery Services that determine and depict the relationships between the system and network resources are also included.
- **View/2**: This is the graphical user interface component for displaying managed resources and interfacing with distributed systems management application functions that support those resources. The common user interface conforms to SystemView Integration Level 2 and CUA-91. A set of enabling services allows easy extension of the presentation metaphor to include functions provided by systems management applications. These services are made available via the System Object Model (SOM) programming interface of OS/2 2.0.
- **Enable/2**: These are managed system services that are provided for OS/2 2.0. These services are a sub-

set of those provided in the managing system; they allow the systems to be managed by the applications on the managing system.

- **Agents/2**: CMIP-based agents are provided for OS/2 2.0, DOS 5.0, and DOS 5.0 with Windows 3.1 that allow the operating system resources to be managed. Agents allow the managing system to request data about the resource managers, and manage the resource manager by setting and changing values.
- **Agents Extended/2**: CMIP-based agents are also provided for managing the OS/2 subsystem resources contained in the Communications Manager, Database Manager, and LAN Server.

IBM is also developing systems management applications to run within the managing system framework, Manage/2. These include:

- **Monitor/2**: System performance management for disk, Random Access Memory (RAM), and processor monitoring of the OS/2 workstations and servers, based on System Performance Monitor/2 technology
- **Fix/2**: System fault management for reporting hardware and software failures with some automated recovery
- **NetView Tie/2**: NetView gateway service for collecting and transforming OSI performance and fault events for transmission to NetView
- **Start/2**: System configuration management for OS/2 workstations and servers without user involvement (The initial release of this product will precede delivery of Manage/2; therefore, it will run as a stand-alone product.)

The strategy is to provide, on this framework, a complete suite of systems management applications for managing the OS/2 LAN environment.

Monitor/2

Based on the current SPM/2 technology, IBM is developing a SystemView-compliant performance management solution for the OS/2 2.0 environment. This solution will support collecting and analyzing a rich set of metrics for OS/2 2.0 performance-critical resources. In addition to the base operating system, metrics from IBM system extensions such as the LAN Server/Requester, Communications Manager, and Database Manager also will be supported.

IBM plans to initiate a customer evaluation program beginning with the framework products in 1992 and including the applications in early 1993.

Several functions, all written to the X/Open Management Protocol API, will be provided to assist in graphing, recording, reporting, and analyzing performance data collected in an SQL database. These functions will work with OS/2 2.0 resource manager agents that are contained in Agents/2 and Agents Extended/2 to satisfy performance management requirements, such as threshold detection, alert generation, and data summarization.

This performance solution will enable system administrators of an OS/2 LAN environment to monitor system performance, analyze performance trends and problems, and use this solution to help in performance tuning, load balancing, and network growth management. Application developers will be able to better analyze designs, verify performance objectives, and optimize application performance for the OS/2 2.0 environment. This solution will also aid capacity planners in benchmarking and performance modeling exercises.

Fix/2

Systems management tools that help in problem management are a key resource for controlling management costs. Fix/2 assists in managing Information Systems (I/S) networks either locally or remotely, and in providing an increased level of service to the entire I/S user community. The increased level of service results from Fix/2's coordination and automation of error recovery and integration with other DSM functions. Service is also enhanced by managing OS/2, DOS, and Original Equipment Manufacturer (OEM) systems from a single managing workstation on a LAN.

Fix/2 provides a focal point and a control point for stand-alone or host-connected LANs. The Fix/2 function enables a system administrator to automate bypass or recovery procedures. Fix/2 uses event management services provided by Manage/2 to filter and collect OSI events emitted by agents and FFST/2-enabled applications on the network. Fix/2 also supports user-customized filtering of input events and automated invocation of external procedures and management operations.

NetView Tie/2

Management/2 applications and agents use different CMIP protocols from those used by the host-based NetView product family. This gateway function converts LAN-based CMIP protocols to the SNA/MS network management protocols used by NetView. As a result, events that require host involvement can be transmitted to NetView as network alerts for further processing. The OS/2 Communications Manager facilities

will be used for the SNA transport to the NetView host.

Start/2

IBM intends to migrate the configuration management tool Start/2, shipped initially as a stand-alone product in conjunction with Configuration, Installation, and Distribution (CID), to the Manage/2 framework.

In support of the response file architecture, which is at the heart of the CID software distribution strategy, the Start/2 generates, on demand, a validated configuration response file for each product on each workstation. Validation, which is done interactively, takes into account connections between nodes and cross-product parameter relationships.

To further facilitate automated code distribution, Start/2 supports either host- or LAN-based distribution strategies. Support for host-based distribution is in the form of NetView DM/2 Build files. LAN-based distribution is supported through the generation of REXX command files to be used by the Network Transport Services/2 product.

Enhancements for the second release of Start/2, the one integrated with Manage/2, will include support for configuring additional connectivities, further simplified configuration administration, and greater integration with NetView Distribution Manager/2 in support of CID.

For more information, see "CID: Remote OS/2 Configuration, Installation, and Distribution of PC Software" in this issue.

Topology/Discovery Services

These common services are part of the Manage/2 product. The system environment that DSM is intended to manage is a collection of LAN-connected workstations. Therefore, it is useful to build and maintain topological information about objects, their locations, and how they interrelate.

Topology Services enables creating and maintaining topological information necessary for topological views of the DSM-managed objects. These services also provide for displaying topology maps using the System Object Model-based presentation services. Topology information is maintained separately from the user interface services to facilitate sharing the topology information.

Discovery Services can initiate network searches for gathering topological information. Topology Services can use the results from discovery searches to prime the topology information base and to maintain the integrity of its topological information.

Customer and Vendor Involvement

IBM plans to initiate a customer evaluation program for the distributed systems management family of products, beginning with the framework products in 1992 and including the applications in early 1993. Product content and general availability decisions for the applications will be made as a result of this customer evaluation.

While IBM is developing the framework and these applications, other applications will be supplied by leading vendors of systems management products. IBM is actively soliciting specific applications as well as generally encouraging vendors to develop their applications on the DSM framework. IBM is providing documentation detailing what is involved in developing an application (or porting an existing application) to run on this framework. IBM is also providing technical assistance. One such example of a vendor-developed DSM application is the integration of NetWare Services Manager for OS/2 function to allow the full management of Novell networks from Manage/2. Over time, existing IBM applications will be ported to the DSM framework environment as well.

IBM is also encouraging vendors to provide agents for their system resources which would allow DSM to manage them, and is supporting those efforts with technical assistance as well.

The combination of IBM's open systems management framework and a full suite of systems management applications and resource agents provided by IBM and other leading vendors will make DSM the most comprehensive Distributed Systems Management solution available for LAN-based systems.

Jim Alexander *is a senior programmer in IBM's Personal Systems Line of Business in Austin, Texas. His current assignment is system planner for the DSM family of products. Jim joined IBM in 1968 after earning a BS in mathematics from Oklahoma State University.*

CID: Remote OS/2 Configuration, Installation, and Distribution of PC Software

Tom Edel
IBM Corporation
Austin, Texas

The IBM Configuration, Installation, and Distribution (CID) facility provides for unattended or lightly attended installation of PC software products and subsequent maintenance. This support works with PC workstations that have no installed software, as well as designated OS/2 workstations whose current customized operating environment can be migrated to subsequent product releases. This article discusses the need for central administration of PCs and describes IBM's approach.

Distributed computing has made computer technology available to many individuals at various levels within an organization. Installing and configuring hardware and software resources associated with distributed computers has the reputation of being complex and labor-intensive. This notion is fueled by the large number of distributed computers and the variety of their capabilities.

Enterprises recognize the importance of managing the installation and configuration of distributed computing resources. Here, the term *managing* denotes the following:

- Remotely initiating product installations according to a pre-planned procedure (in other words, who gets what when)
- Remotely initiating recovery procedures for failing systems
- Remotely servicing installed products
- Detailed tracking of product installations on remote workstations

In distributed computing environments, there are usually two configuration objectives relating to product installations:

- Standardize hardware and software configurations so that users can perform the same tasks at several workstations. An example of this environment is bank tellers' workstations.
- Personalize hardware and software configurations so that computing capability can vary by workstation, and therefore satisfy specific needs, such as installable program features and tuning parameters. Engineers and programmers need the flexibility afforded by a workstation that is tailored to their needs.

Often, some mix of standardized and personalized computing is appropriate to many organizations. How, then, should the configuration, installation, and distribution of an organization's computing environment be managed?

Configuration

Configuration is a procedure that occurs periodically for new versions of installed software, for changes in the physical characteristics of workstations existing in a domain, and so on. As applied to a particular product, configuration generally means specifying values that control the product's execution characteristics. For example, buffer size and number of network connections are part of the configuration for a network transport product installed on an OS/2 workstation. As applied to a workstation, configuration involves the hardware and software installed at that workstation, including product versions, service levels, and so on.

Many organizations would like to configure software and workstations from a central site so that typical workstation users do not need this specialized knowledge. Managing configuration from a central site would also take into consideration the hardware and software interaction among all workstations within a domain, which is of no concern to the typical computer user. IBM's approach to the configuration problem is to provide a centralized management capability that removes configuration tasks from workstation users.

Installation

Installing software on a workstation usually involves an unfamiliar process that occurs infrequently and varies by the type of product to be installed. Even the installations of similar data communication products – Systems Network Architecture (SNA), NetBIOS, and TCP/IP – differ significantly. A legitimate question to ask is: Why do I need the detailed skills necessary to install products when I only perform this task a few times a year?

One approach to solving this problem is to hire knowledgeable people to physically configure and install each workstation. This approach is usually not practical because there are too many workstations. IBM's approach to installing software on OS/2 workstations is to provide a central man-

agement capability that directs the installation of applications that have been enabled to participate in this environment.

Distribution

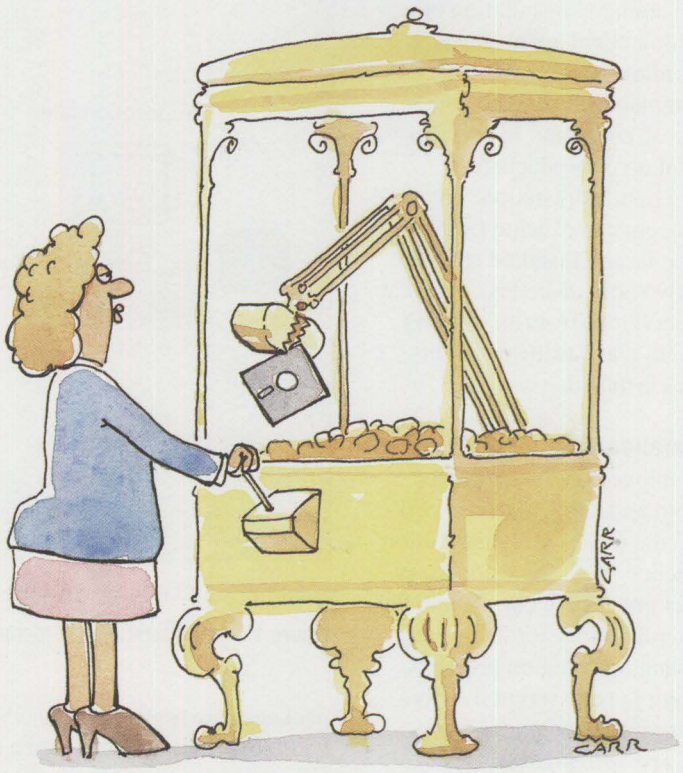
If configuration and installation skills are concentrated among a few individuals, then a distribution component also is required to disseminate the new software according to centrally managed configuration and installation policies. In the past, the term *distribution* meant that diskettes were given to users whose workstations needed new or upgraded software, additional functions for software already installed, or software servicing. Distribution also was done by trained individuals who physically went from workstation to workstation performing the configuration and installation tasks.

For some PC products, partial relief to the distribution problem is possible. This relief is through server workstations that support redirected client requests for access to stored diskette images. However, this solution does not address central management of client configuration and product installations, or the unattended mode of target workstations.

Another aspect of distribution is the need to associate particular products with particular workstations so that new versions, as well as their servicing and recovery, can be managed. When workstations are connected to a central distribution site, it is possible to install particular products on selected workstations at predetermined times. In this case, the workstations do not need the presence of an individual.

IBM's CID Solution

Collectively, the set of systems management capabilities addressed by the remote, unattended installation of PC workstations is known as *Configuration, Installation, and Distribution* (CID).



The term *unattended* means that an individual does not need to be present at a workstation before or during installation of CID-enabled products. CID-enabled products can be installed on weekends or after normal work hours. For a certain class of CID products (such as those that require no rebooting before activation), installations can even occur while the workstation is running other applications, with no disruption to the operating environment.

IBM's solution for distributed systems management addresses workstations that are connected in a Local Area Network (LAN) or Wide Area Network (WAN) environment. It supports two environments: push and administered pull. Figure 1 shows the CID operating environment on a LAN and Figure 2 shows the CID operating environment on a WAN.

Server-Initiated CID Installations (Push): A one-time, attended installation of a CID-enabling client agent is needed. This allows a workstation on a LAN to receive install requests from a code server workstation, which is a repository for CID-enabled product images and other related files. This enabling agent is a continuously running program that waits for install requests sent by the code server. The enabling agent can execute from a system that has no installed software by using two boot diskettes created at the code server, or from an existing OS/2 system that has the necessary CID client enablement procedures installed. The CID process is used to maintain and install new product releases for the CID-enabling agent.

Client-Initiated CID Installations (Pull): An attended installation of a CID-enabling client agent is necessary. This allows a workstation to access command files on the code server. These command files direct the installation of products for a particular client workstation. This enabling agent can execute from a system that has no installed software by using two boot diskettes created at the code server, or from an existing OS/2 system that has the necessary procedures installed.

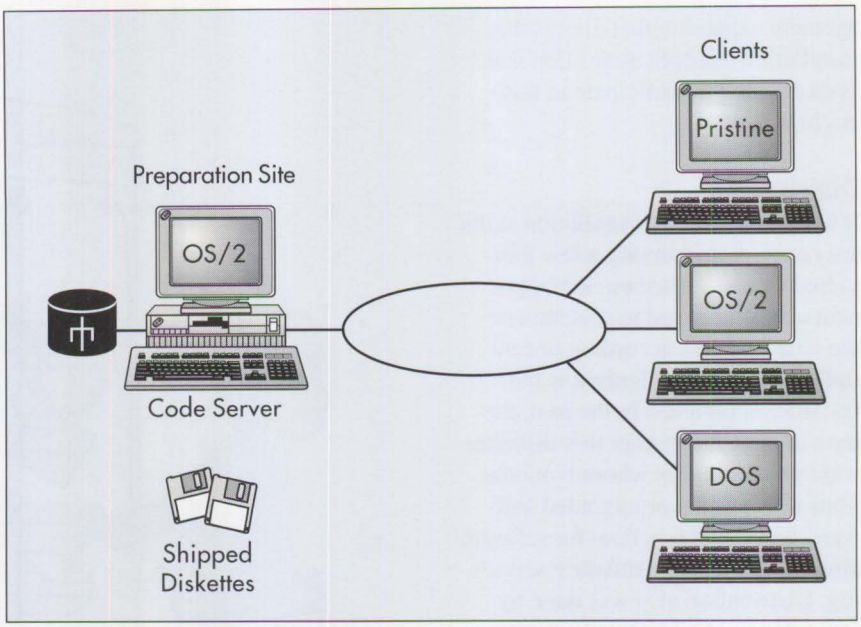
CID Preparation Site Functions

Products shipped on diskettes are copied onto an administrator's workstation. In the OS/2 LAN distribution environment with a single code server, product images are copied into the code server directory for later reference by client installation products. In the multiple code server distribution environment, the image copy function is performed at an OS/2 Preparation Site. The NetView Distribution Manager (DM) subsequently transmits this as installation packages to OS/2 code servers. The installation programs executing at client workstations are unaware of the distribution environment in which they exist.

At the administrator's site, a product configuration is done for each participating CID workstation. This creates a file containing product configuration and installation information that can be unique for each workstation. This file, called a *response file*, replaces the workstation user who normally enters configuration and installation data.

Code Server Distribution

Installation packages are created for the multiple code server distribution environment. These packages may contain product code images, response files, and the install commands that indicate which products to install at particular client workstations. These packages are transmitted to OS/2 code servers and placed into a directory to be accessed by client installation programs. The command that starts a client installation is sent from the NetView DM host administrator to the OS/2 code servers in the WAN environment. The command is invoked at the individual code server sites by OS/2 administrators in a NetView DM/2 LAN environment.



Note: "Pristine" means no installed software.

Figure 1. CID Operating Environment on a LAN

In the single code server distribution environment, product images and response files are placed into a directory for reference by the client installation products. There are two methods that can be used for client installation:

- For each product, the code server administrator enters an install command to indicate which client workstations will receive the planned installation. Client installation programs are initiated when the client installation agent receives an install request from the code server.
- A command file, created by the code server administrator, indicates the sequence of product install commands intended for each client workstation. The initiation of client install programs occurs when the activated client installation agent begins executing install commands (via drive redirection) on the code server.

The command that the code server uses to initiate the installation of client programs also can be a command that prompts the workstation user for the next product install command. This is called *administered pull* because the command file that permits the client prompt function is administered centrally at the code server site.

In the previously described LAN distribution methods, the timing of an installation determines whether a client user must initiate the installation activity. There are advantages to each LAN distribution method, and both capabilities can coexist at the same code server.

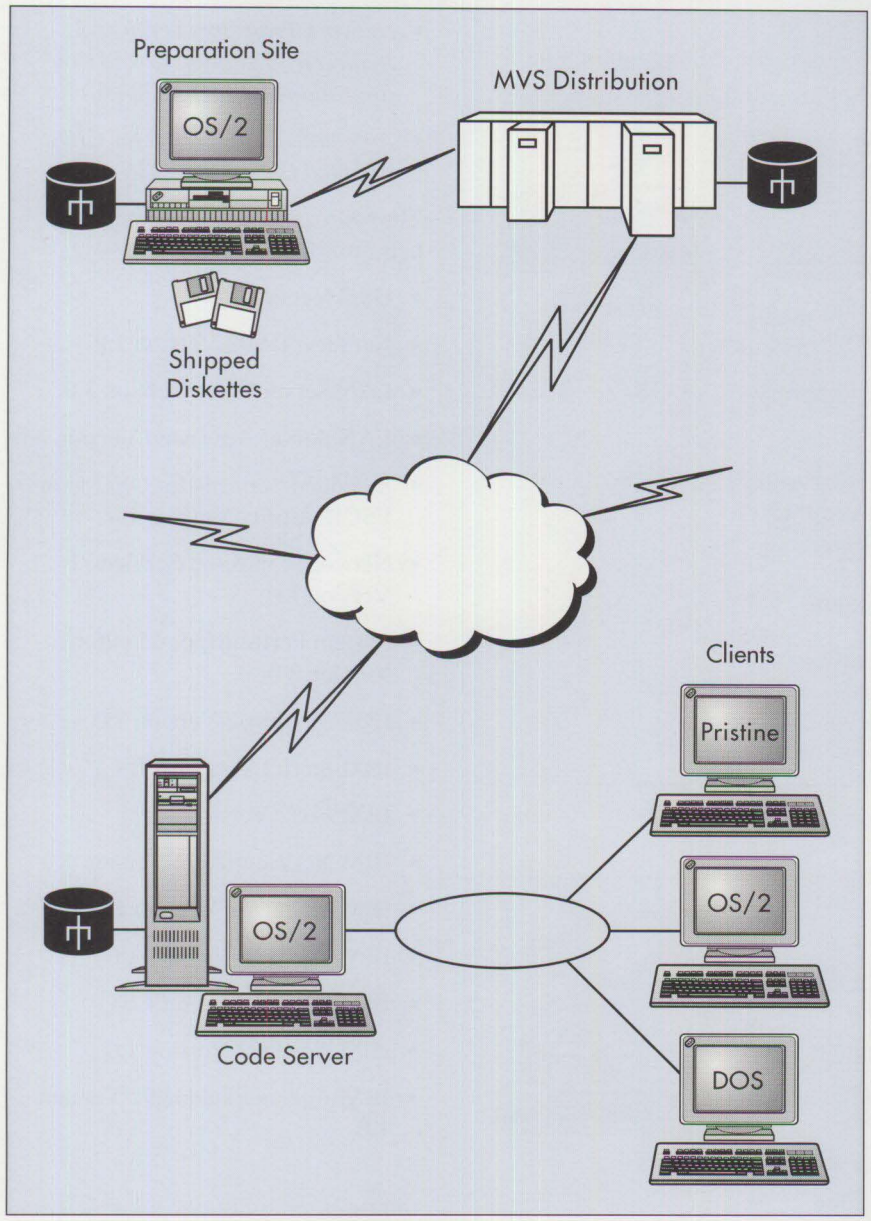


Figure 2. CID Operating Environment on a WAN

Client Installation Procedures

When invoked, the installation program (designated by the administrator) receives the following:

- Input parameters indicating the code server location for the product installation images
- Log files where the installation program records installation progress and errors
- Response files

The primary interface between install programs and the code server is a set of parameters passed to the install program, a set of return codes passed back when installation finishes, and the ability of install programs to access redirected drives.

CID Recovery

During the installation or execution of OS/2 products, it may be necessary to restore a particular product, drive, partition, or system to a prior state. This condition could be the result of an invalid product installation, invasion by a virus, installed fixes, or product versions that do not execute properly.

The OS/2 CID system provides a system-wide recovery procedure (as shown in Figure 3) in which the data for all products within a partition or drive is saved locally or at a backup server. This type of recovery relieves each product from having to provide its own backup and recovery procedure, and removes all interproduct file dependencies. This method also removes the requirement that each product must specify a set of files to be backed up and restored. Hidden files, directory structures, system files, read-only files, files with extended attributes, and so on, are all saved and restored.

CID Migration

It is likely that users will want to retain prior installation and configuration parameters when a software product is upgraded. Some CID products provide this capability by specifying response file parameters, while others use prior settings implicitly. Most CID-enabled products allow the user to specify new, defaulted, and migrated parameters within the installation programs, as illustrated in Figure 4.

CID-Enabled Products

IBM provides a description of the enablement guidelines to programmers who want their products to participate in IBM's CID systems management environment. Software vendors are encouraged to enable their installation programs for CID. Additionally, recognizing that some enterprises want to substitute their own customized distribution or configuration procedures, IBM provides descriptions of the interactions of these CID components.

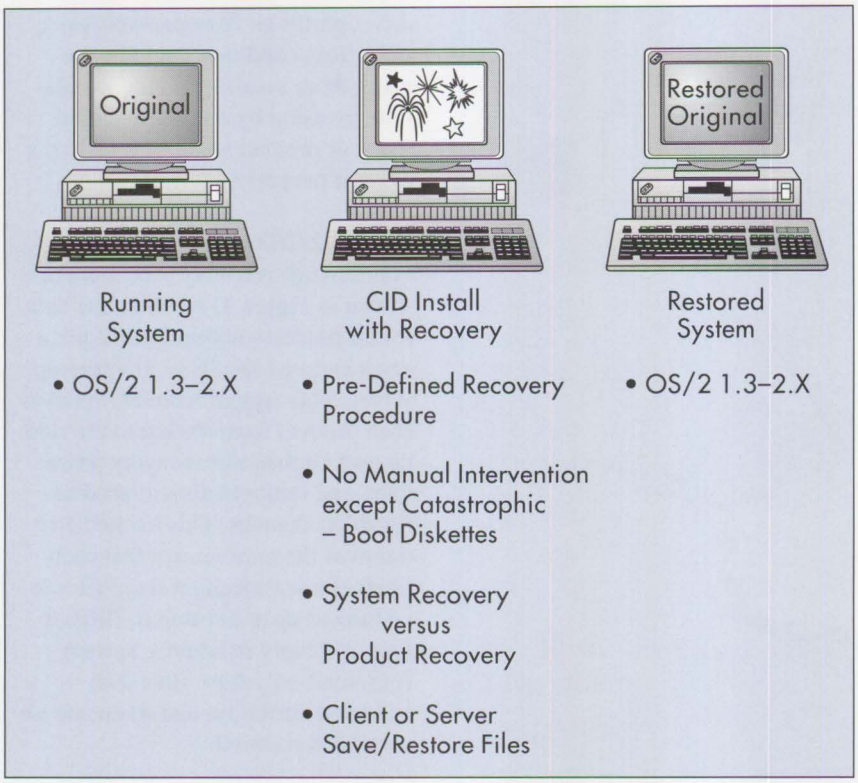


Figure 3. CID Recovery

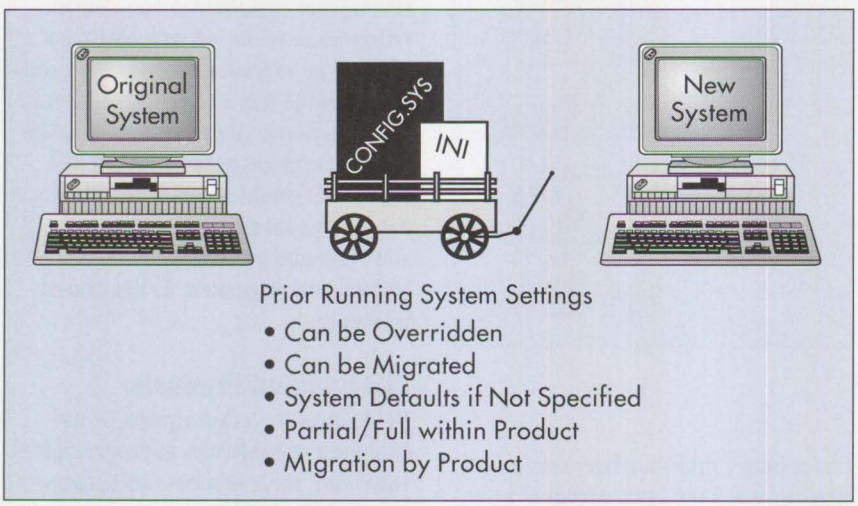


Figure 4. CID Migration

Publications on these topics include the following:

- *Automated Installation for CID-Enabled Extended Services, LAN Server 3.0 and Network Transport Services/2* (GG24-3781)
- *OS/2 Remote Installation and Maintenance* (GG24-3780)
- *Network Transport Services/2: Redirected Installation and Configuration Guide* (F96F8488)
- *Automated Installation for CID-Enabled OS/2 2.0* (GG24-3783)

The following IBM software products are enabled for CID:

- OS/2 Version 2.0
- NetView DM/2 Version 1.0
- LAN Server Entry Version 3.0
- LAN Server Advanced Version 3.0
- Extended Services for OS/2 with ESCID Utility Version 1.0
- Network Transport Services/2 Version 1.0
- System Performance Monitor/2 Version 2.0
- IBM Monitor/2 Version 1.0
- IBM Start/2 Version 1.0
- IBM Fix/2 Version 1.0
- IBM NetView Tie/2 Version 1.0
- IBM Manage/2 Version 1.0
- IBM Enable/2 Version 1.0
- IBM View/2 Version 1.0
- IBM Agents/2 Version 1.0
- IBM Agents Extended/2 Version 1.0

Tom Edel *is a senior programmer in IBM's Personal Systems Line of Business in Austin, Texas. He is currently responsible for Configuration, Installation, and Distribution design for OS/2. He joined IBM in 1961 after earning a BS in industrial engineering at the University of Pittsburgh.*

Start/2: Putting the Configuration into CID

Khalil Emami and Theodore Shrader
IBM Corporation
Austin, Texas

This article introduces IBM Start/2, part of the distributed systems management suite of products announced in late 1992. It explains how Start/2 can be used in configuring workstations from a central location and its role in the remote installation of products. The article also offers a hands-on, step-by-step laboratory session for readers who have recently purchased Start/2 or those who plan to do so.

IBM Start/2 Version 1.0 provides users and network administrators with a friendly, efficient graphical environment for remotely installing OS/2 2.0 and OS/2 subsystems (see "OS/2 Distributed System Management" in this issue). It helps administrators to plan and manage their networks more efficiently. Whether it is creating a new network or migrating workstations from OS/2 1.X to OS/2 2.0, Start/2 is the product of choice. It uses the drag-and-drop capability inherent in the OS/2 Workplace Shell and offers context-sensitive online help. Start/2 can be installed in just a few minutes to launch network administrators into the world of Configuration, Installation, and Distribution (CID).

Configuration for Remote Installation

In the remote installation of OS/2 2.0 and its subsystems, response files substitute for a user's keystrokes. A *response file* is an ASCII file that directs the installation or configuration process of a product on a workstation. It responds to questions such as "Which drive to install on?" and "How many 3270 emulation sessions?"

In the CID process, Start/2 provides the Configuration part, which includes creating response files for CID-enabled products. But Start/2 is more than just a tool for generating response files – it also assists in the remote installation of OS/2 and OS/2 subsystems, such as Extended Services for OS/2 and OS/2 LAN Server.

A product is considered CID-enabled if it can be installed using remote drives, if it processes response files rather than panels for user-entered parameters, and if it provides a return code to the distribution product that indicates the success or failure of the remote installation. (Products not requiring panels are still CID-enabled if they meet the other criteria.) Such a product can be installed with little or no user interaction. The product's installation program retrieves the product's image files from a code server machine, takes keyword values from a response file, and remotely installs the images on the target computer.

The current list of IBM's CID-enabled products appears in the article titled "CID: Remote OS/2 Configuration, Installation, and Distribution of PC Software" in this issue.

Introducing Start/2

Start/2 currently generates response files for Extended Services 1.0, LAN Server 3.0, and Network Transport Services/2 (NTS/2). It also helps in distributing and installing CID-enabled products by producing LAN CID Utility (LCU) command files and NetView Distribution Manager/2 (NetView DM/2) change files.

The LCU, a part of NTS/2, is a CID configuration and installation process. Figure 1 depicts the CID LAN solution. In an enterprise environment or an independent LAN environment, Start/2 can generate change files for OS/2 NetView DM/2. Start/2 can graphically generate the information and the definitions needed to configure and install products on a workstation.

Start/2 makes it possible to configure workstations from a central location. It improves the network administrator's productivity and eliminates the need for users to possess the skills or specialized knowledge of the installed product. A code server that holds the product files can use the response files generated by Start/2 to initiate the installation program. When the LAN CID Utility is used, Start/2 can generate command files that are executed by LCU.

Start/2 Capabilities

Start/2 is an OS/2 2.0 application with a friendly, efficient, graphical environment for planning networks as well as the configuration and installation of software products on networked workstations. Networks, nodes, and connections can be graphically represented easily in Start/2 by drag-and-drop operations on objects. Workstation configuration is done by modifying the notebook settings of each workstation.

The context-sensitive help in Start/2 frees users from referring to the user's guide for help. Start/2 also performs configuration validation and connection verification while the user is building a network topology. Start/2 prevents the user from con-

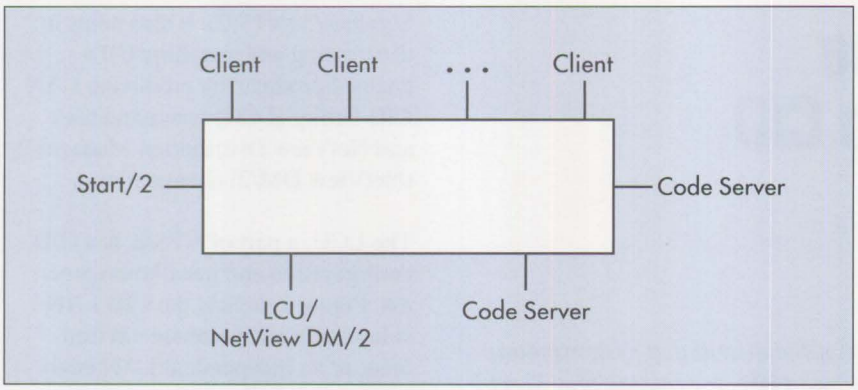


Figure 1. LAN CID

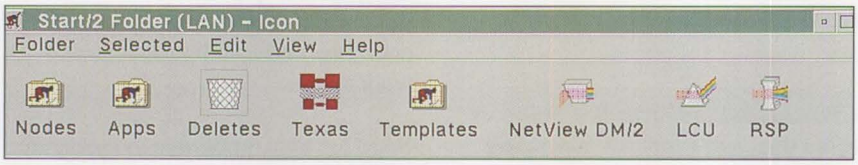


Figure 2. Start/2 Primary Folder

necting a database client workstation to a node without database server capability. Addresses and workstation names are automatically generated. User-defined values are allowed.

Start/2 is designed to offer several connection views, with initial support for the LAN, Database, and 3270 views. This feature increases the user's ability to visually inspect the node connections. After a network is defined or modified, it can be saved in a Database Manager relational database or in ASCII files.

The Start/2 user interface, similar to the OS/2 Workplace Shell, consists of folders, containers, templates, and other objects. At the base of the tree structure of these objects is the Start/2 primary folder, which contains every object needed to create a network and to generate response files. Figure 2 shows the Start/2 primary folder.

Topology

The Topology object is the actual workspace in which a network is designed and represented. A *topology* is a graphical arrangement of node objects, their connections, and attributes. A topology represents a network or part of a network with a distinct characterization (such as LAN servers and requesters). A network consists of one or more network topologies. Start/2 enforces unique node names and addresses across the network. In a network topology, workstations that are installed with DOS, DOS and Microsoft Windows, or OS/2 1.3 also are represented. Response files are generated only for workstations configured with OS/2 2.0.

The contents of the topology window can be displayed in presentation and feature views. The presentation views initially supported are the icon and detail views. Nodes can be displayed as icons with their connection lines or in a table form with attribute values. The feature views initially supported are the LAN, 3270, and database views. When the topology view is set to a particular view, such as the LAN view, only those node objects and connection lines that relate to the LAN view are highlighted; the other nodes are grayed and their connection lines are not shown. Figure 3 shows a topology with five nodes in 3270 view. (See Figures 8 and 9 for different views of the same topology.)

To create a network topology, the user can double-click with the mouse button on the topology icon to open the topology window (the workspace). Next, the user can drag and drop the necessary node objects from the node container, change (if necessary) the attributes in the node notebooks, and draw the connection lines between the nodes. If similar workstation nodes must be built on the network, the "duplicate nodes" feature can be used to copy as many nodes as desired.

Start/2 helps migrate existing nodes with Communications Manager, Database Manager, and LAN Services to OS/2 2.0 base systems. A discovery program collects the necessary data from the workstations, which could have back-level operating systems, and creates import files for each node.

Once the import files have been created, the administrator simply drags and drops the files onto the selected topology. Start/2 builds the graphical display of the topology defined by the import files, fills in each node notebook, and derives the connections for LAN and 3270. This establishes the selected topology. During the process of creating the topology from the import files, the Start/2 program generates a log file to record anomalies. An administrator can use the log file to complete the definition of the workstations whose configurations the Start/2 program was unable to fully derive – such as a database client whose server name is unknown. The administrator can save this topology in an ASCII or SQL database. This process can be repeated for each topology in the enterprise.

After the topology is created, the user or administrator can generate response files by dropping the entire topology onto the transformer object. If indi-

vidual workstation response files are needed, those nodes can be dragged and dropped onto the transformer.

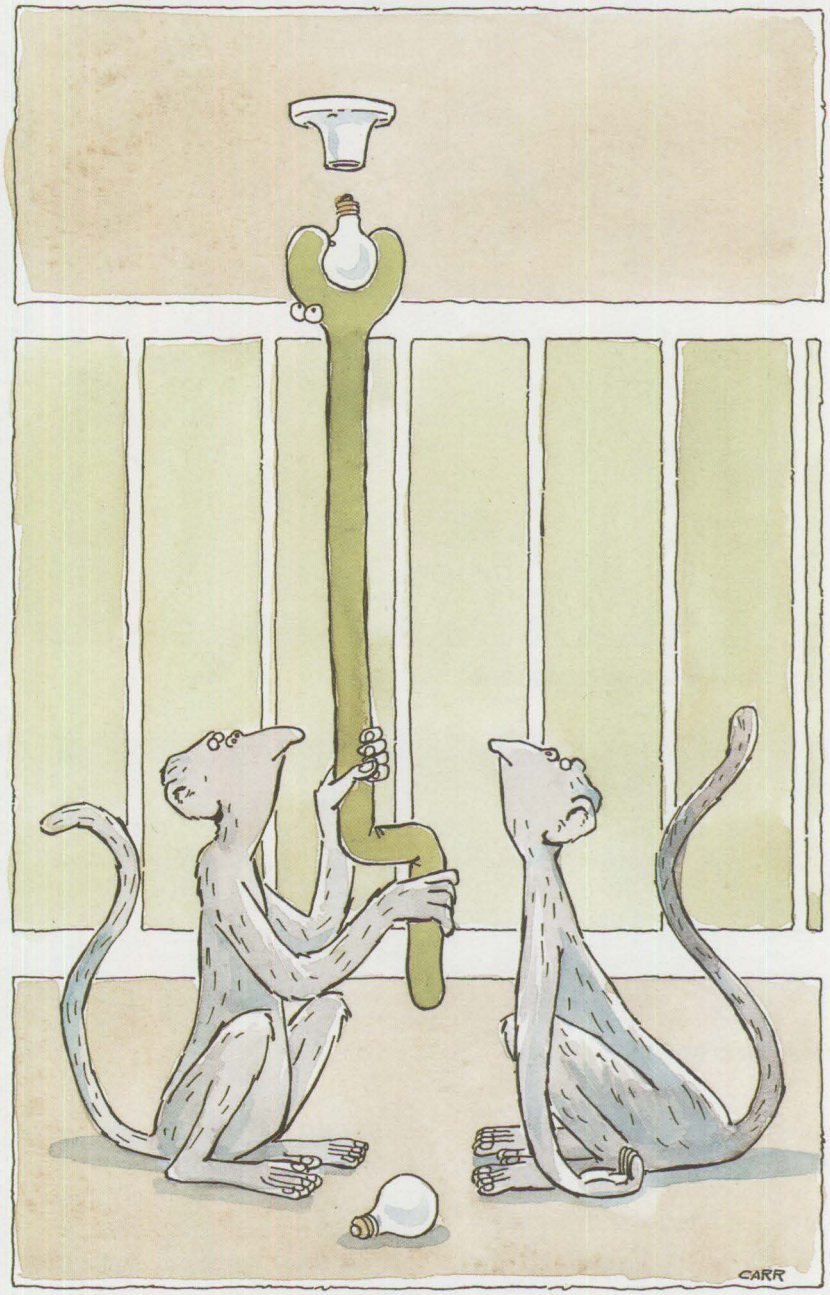
Nodes

The Nodes container consists of a set of canned node objects that can be used to construct a network topology. Depending on the function of a node in a network and the view in which the node is displayed, the node may take on a different shape. For example, a database server node icon, which looks like a cylinder in the database view, has a different shape in the other topology views. If that workstation has 3270 emulation capability, then when the topology view is switched to 3270, the workstation changes into an emulator workstation with a connection line to the host (if the connection exists); otherwise, it becomes grayed with no connecting lines. Figure 4 shows the node objects provided with the product.

Nodes are configured by setting or modifying the values in the node notebook. Figure 5 shows a page in the notebook. Start/2 also offers another way to set the node values. An Attribute Value File can be used to update a set of information about nodes. Similar to an import file, the Attribute Value File can be dropped on the topology to update the nodes with names matching those specified in the file.

Transformer

A Transformer object represents the process by which response files are generated. To start the process, an object is dropped onto a transformer. A single node, a group of nodes, the whole topology, or a group of topologies can be dropped onto the transformer for processing and generating response files. Currently, three types of transformers are supported: RSP, LCU, and NetView DM/2. The RSP transformer generates configuration response files; the LCU transformer generates LCU command files. The



NetView DM/2 transformer generates change files that can be processed by the NetView DM/2 server. Figure 6 shows an example of a response file that was generated for a LAN domain controller node.

Templates

The Templates folder contains object templates. Using the OS/2 Workplace Shell concept, these objects can be used to create other objects with drag-and-drop operations. Inside the Templates folder are other templates, including Transformer, Application, Topology, Folder, and Node.

Applications

The Applications folder contains icons that represent CID-enabled applications for which Start/2 can provide installation commands to the CID distribution product. An administrator can drop an application icon

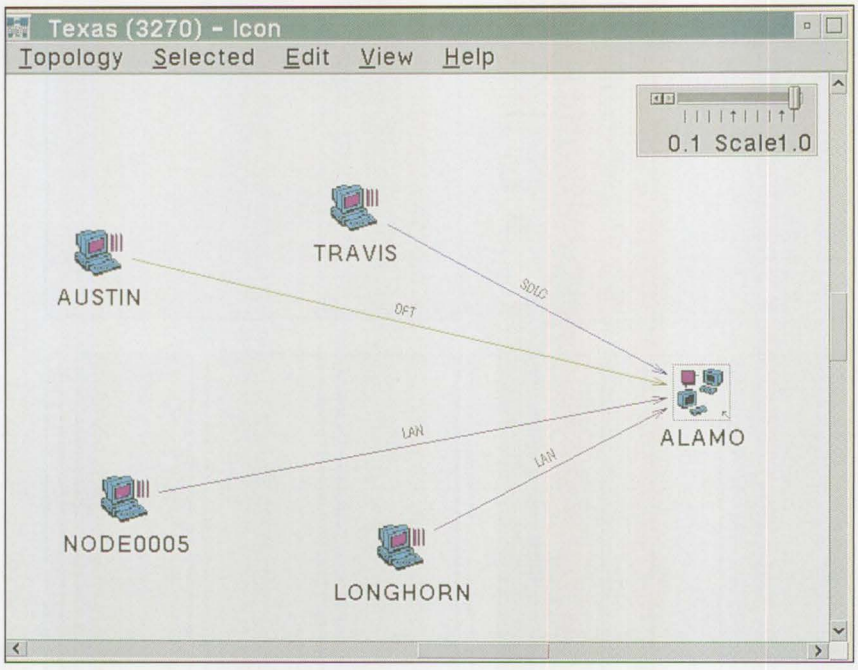


Figure 3. Network Topology in 3270 View

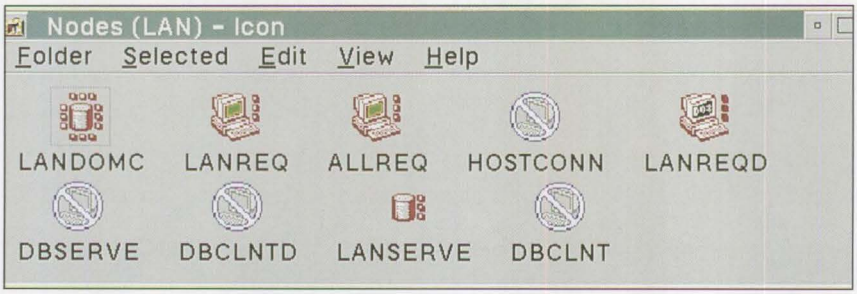


Figure 4. Node Objects

on a node in a topology and cause Start/2 to associate the application with the selected node. When output processing is selected with a transformer, the commands to install the selected application will be added to the distribution command file generated for that node.

Deletes

The Deletes object acts as a wastebasket. Nodes and other objects that are not needed can be discarded by dragging and dropping them onto the Deletes object. If an object is deleted by mistake, it can be recovered by opening the Deletes container and dragging the deleted object from it.

Laboratory Session: Creating a Network

The remainder of this article shows nearly all the Start/2 features in a tutorial. The tutorial can familiarize you with the program at a faster pace. It provides an overview of many of the features available in Start/2. Note that not all steps are used in all cases. Many steps are executed only once for the lifetime of the network.

1. Using the data collector `OCUNODE`, collect information about the existing workstations. These are nodes running back-level operating systems that should be migrated to OS/2 2.0, or nodes already running OS/2 2.0 that should be brought into the same network topology so their settings can be saved. (If the network is constructed anew, skip this step and begin with Step 2.)
2. Invoke Start/2. The command options are set according to the type of database used. For example, `OCUNET /DBASC` brings up Start/2 using an ASCII database.
3. Open the LCU Transformer from the Start/2 primary folder by double-clicking the mouse button on the icon. Figure 7 shows the transformer status window. Note the four boxes on the left. They show the names of nodes and their response file generation status. Messages can be viewed for nodes in the Rejects box, and response files can be viewed or edited for nodes in the Successes box.
4. In the Directory Mapping list box, double-click the left mouse button on `SERVER01`, and set the file path, servers, and aliases. On the Mappings tab (page 1 of 3), the left column specifies the paths that Start/2 can access to store the generated command files, log files, and response files. On the same page, the server and alias settings indicate which servers hold the files, so that every client connected to that server can access the files. Pages 2 and 3, the server and alias settings, specify where the product images or files can be accessed for installation. Close the notebook by clicking on `OK`.
5. Open an empty Topology container from the Start/2 primary folder by double-clicking the mouse button on the icon.
6. Open the topology notebook by selecting `Open as settings` from the Topology pull-down, and change the settings if necessary. A Directory Mapping (in this case, `SERVER01`), node name pattern, and 3270 connection type can be

set to desired values. The name of the topology also can be specified, so click on the General tab of the notebook and change the name of the topology to TEXAS.

7. Open the network notebook by selecting Open as network settings from the Topology pull-down. Type of adapter (Token Ring or Ethernet™), next LAN adapter address, and domain name pattern are specified with this notebook. Click on the Path tab and set the supplemental response files' paths. One or more supplemental response files can be selected. These files are included in the response files generated by Start/2. Later, they are used by the install program of that product.

8. Open the Nodes folder from the Start/2 primary folder by double-clicking the mouse button on the node icon.

9. Drag nodes from the Nodes folder into the topology window in the following order:

   - LANDOMC, the domain controller
   - ALLREQ, the multipurpose requester
   - HOSTCONN, host connections
   - DBSERVE, the database server

   LANDOMC changes the name to NODE0001 when it appears in the topology window. Other nodes follow the same pattern. This pattern could have been set in Step 6.

10. Double-click on the Domain Controller (NODE0001) to open the notebook to change its settings. This opens the notebook to the first page of the General tab. On that page, change the Node name from NODE0001 to AUSTIN. Click on the Adapter tab and select the DFT Adapter check box. Click on the 3270 tab and select the

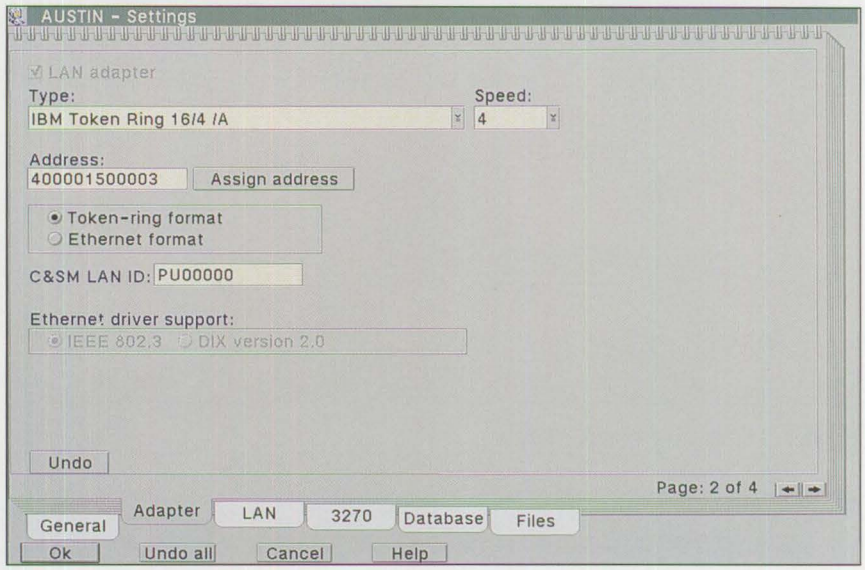


Figure 5. Node Notebook

```
;  8-30-1992  7:10:15 pm
C:\ocudir\RSP\LS30\OCURSP\NET1\AUSTIN.RSP

CONFIGTARGETDRIVE = C
UPDATEIBMLAN =      NETWORKS
                    <
                    NET1  =  *,*,*,32,51,*
                    >

UPDATEIBMLAN =      REQUESTER
                    <
                    COMPUTERNAME  =  NODE0001
                    DOMAIN  =  DOMAIN01
                    >

UPDATEIBMLAN =      SERVER
                    <
                    MAXCONNECTIONS  =  282
                    MAXLOCKS  =  64
                    MAXOPENS  =  250
                    MAXSEARCHES  =  50
                    MAXSHARES  =  16
                    MAXUSERS  =  45
                    NUMBIGBUF  =  20
                    NUMREQBUF  =  90
                    SRVHEURISTICS  =  *******0*****4*****
                    >

ADDIBMLAN =         SERVER
                    <
                    SRVSERVICES  =  NETLOGON,ALERTER,NETRUN
                    >

UPDATEIBMLAN =      LSSERVER
                    <
                    SRVPIPES  =  3
                    >
```

Figure 6. Example of a Response File

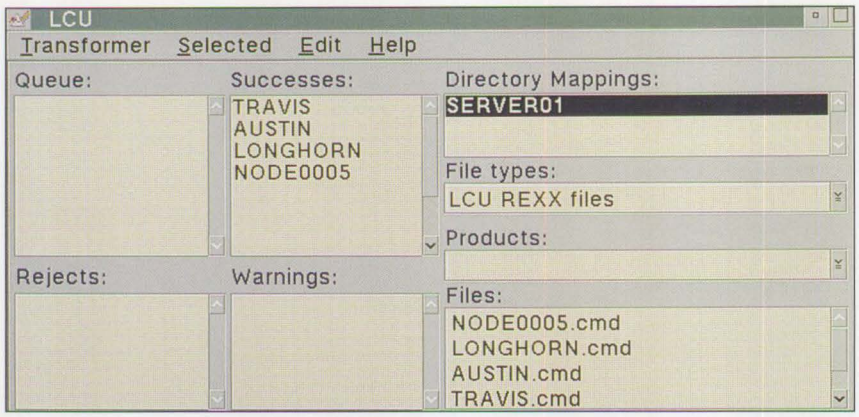


Figure 7. Transformer Status Window

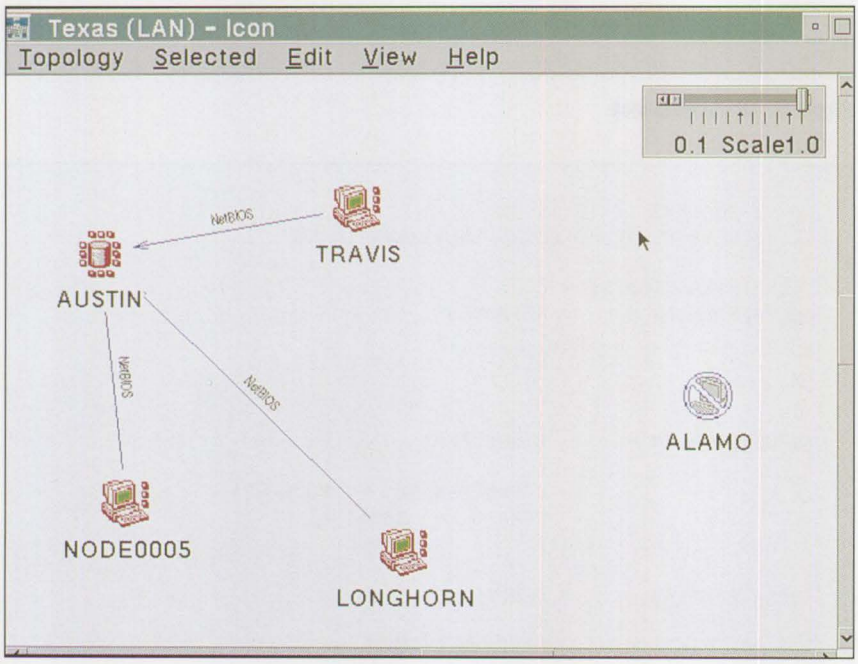


Figure 8. Network Topology in LAN View

Emulator radio button, then page forward to DFT sessions. Click on the drop-down list (under Device) and select Mod 2. Move the cursor to the Session ID field and enter DFT1. Move to the Short ID field, enter A, then click on Auto-start. Click on the Database tab and select OS/2 Database client. Finally, click on the Files tab and select the supplemental response files and the target drive for all products to be installed. Click on OK; this ends the notebook setup for the domain controller AUSTIN.

11. Change the settings on the database server (NODE0004) to configure the node. Follow the procedure stated in Step 10, and make the following modifications:
    - Change the node name from NODE0004 to TRAVIS.
    - In the Adapter tab, on page 1, select SDLC.
    - In the LAN tab, on page 1, select OS/2 requester.
    - In the 3270 tab, on page 1, select emulator.
    - In the 3270 tab, on page 3, configure one 3270 non-DFT host session.
    - Select all supplemental response files.

    Notice that page 1 of the Database tab shows that Database server is selected.
12. Change the name of the host node (NODE0003) to ALAMO.
13. Double-click on NODE0002 to open its notebook. Change the node name to LONGHORN, and select the supplemental response files. (These files are defined by the user.)
14. Copy the node LONGHORN by holding the Ctrl key and dragging the node icon. Notice that the new node name is NODE0005.

    The completed topology is shown in Figures 3, 8, and 9 in 3270, LAN, and Database views respectively.
15. Configure an application as follows:
    - Open the Apps container from the Start/2 primary folder.
    - Double-click on Start/2 icon to open the object's notebook.
    - Complete the application program pages and the application resource requirements, and close the notebook.
    - Drag the Start/2 application object, and drop it on the AUSTIN node. This causes the applications to be installed on node AUSTIN.
16. Draw the connections as follows:
    - In the LAN view, move the mouse arrow to the LONGHORN node, press and hold the mouse button, move the arrow

to AUSTIN, and then release the mouse button. Repeat this for TRAVIS and NODE0005.

- Change to the 3270 view by clicking on the View pull-down on the Topology action bar and selecting 3270 from the menu. Then draw connection lines from nodes LONGHORN and NODE0005 to node ALAMO. Notice that the line tag indicates LAN, the type of connection.
- Open the topology notebook by selecting Open as settings from the Topology pull-down. Click on the 3270 tab, select SDLC, then close it. This allows the default type of the 3270 connection to be changed. Draw a connection line between nodes TRAVIS and ALAMO.
- Open the topology notebook by selecting Open as settings from the Topology pull-down. Click on the 3270 tab, select DFT, then close it. Draw a connection line between nodes AUSTIN and ALAMO. This uses the third type of 3270 connection available in Start/2.
- Change to the database view by clicking on the View pull-down on the Topology action bar, and selecting Database from the menu. Then draw connection lines from nodes AUSTIN, LONGHORN, and NODE0005 to node TRAVIS.

17. At this stage, the import files (from Step 1) can be dropped on the topology. This step is optional, depending on whether Step 1 was executed. The import files created in Step 1 can be dropped on another topology. In this example, no nodes have been migrated.
18. Using the Edit pull-down on the Topology action bar, choose the Select all option to highlight all the nodes. Click on the Selected pull-down and select Mark supplemental; from this menu, select All of the above. This marks the selected nodes for the selected supplemental response files to be included in the response files generated by Start/2.
19. Drag the topology and drop it onto the LCU Transformer. Response files are generated per node, per product. (LCU command files are also generated.) Figure 7 shows details of the transformer status window. These files are stored according to the settings performed in Step 4.

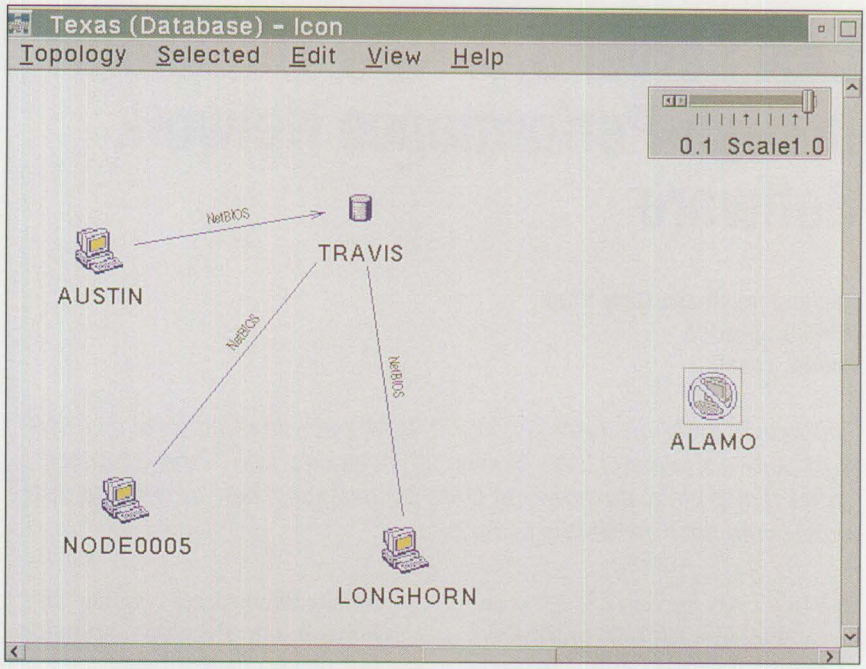


Figure 9. Network Topology in Database View

This concludes the involvement of Start/2 in generating the necessary response files for remote installation of software on the workstations in the network. These response files are accessed by the client workstation while the install program is executing. The command files generated by Start/2 are executed by the LAN CID Utility to automate the remote installation and configuration process.

Khalil Emami *is a senior associate programmer in the Start/2 test and beta support group, IBM Personal Systems Line of Business, Austin, Texas. He has worked in the development of the DCE. Khalil holds a BS in computer engineering and a BA in computer science from the University of Texas, and an MS in computer science from Southwest Texas State University.*

Theodore Shrader *is a senior associate programmer working on the Start/2 program in the IBM Personal Systems Line of Business in Austin, Texas. He has worked on graphical interfaces to the OS/2 Database Manager and on graphical application development. Theodore has a BS in computer science from the University of Texas at El Paso.*

LAN Server 3.0: New Thresholds in High-Performance Network Software

Steven French and Gary Hunt
IBM Corporation
Austin, Texas

This article describes LAN Server 3.0, IBM's new high-performance network software product. LAN Server 3.0 includes a new 32-bit advanced server that exploits the power of OS/2 2.0 and many new usability, performance, and interoperability features.

IBM's LAN Server (LS) 3.0 is an exciting new addition to IBM's family of network software products. Key new features include:

- **A new Advanced Server**. When the Advanced Server was introduced in LAN Server 2.0, it represented the state-of-the-art 32-bit server, running at the same privilege level as the OS/2 1.3 kernel. New 32-bit functions have been added to coincide with LAN Server 3.0's support of OS/2 2.0. These include a cache larger than 16 MB, improved memory virtualization techniques, 32-bit file system Application Programming Interfaces (APIs), and object-oriented implementations.
- **A new Peer Server**. This new feature in LAN Server 3.0 fills the gap between simple client and full server functionality. It allows OS/2 2.0 requesters to perform server functions for one requester at a time. The Peer Server is designed to support casual file and print sharing requirements. The number of drive connections is limited, which helps to reserve processing power for local client applications. In all other respects, however, the Peer Server is just like a full server. There is no limit on the connections that support remote interprocess communications, and the Peer Server provides full support to distributed applications running between peer clients. The Peer Server also supports LAN Server remote administration, and it enforces access control with user-level security and the full LAN Server access control model.
- **Additional DOS support for OS/2 2.0**. This includes new DOS Virtual Device Drivers (VDDs) that support the IEEE 802.2 and IBM DOS LAN Requester (DLR) APIs.

  In LAN Server 2.0, IBM offered a VDD for OS/2 2.0 that supported `INT 5C`, the DOS software interrupt interface into the NetBIOS protocol stack. In LAN Server, virtualization of the OS/2 2.0 DOS box has been extended by providing a VDD for the IEEE 802.2 DOS software interrupt and for the DOS LAN Requester software interrupts, `INT 2AH` and `INT 2FH`. With these new VDDs, the OS/2 2.0 DOS box can be configured as a full-function DLR client without the actual DLR code. This enables LAN-aware applications to run in the DOS box and to access LAN facilities implemented in the DOS 2.0 client. For example, it supports the LAN-enabled features of Microsoft Windows 3.1 in the OS/2 2.0 refresh, without the need for the full DLR client to be running.
- **Remote installation support**. This provides both attended and response file-driven remote installations of both Network Transport Services/2 (NTS/2) and LAN Server.
- **A "thin client."** The reduced disk usage option allows the full-screen interface for LAN Server to be executed from a copy on a server.
- **Network Transport Services/2**. This state-of-the-art network transport software is included at no extra charge, giving OS/2 clients a wide range of network adapters and protocols from which to choose. The improvements in NTS/2 include enhanced multi-adapter support, enabling more than one network card in a single machine to be attached to single or bridged segment networks. Installation of complex network configurations has been simplified and performance has been improved.
- **Key fault-tolerance enhancements**. These include mirroring the boot partition and mirroring without formatting.
- **New DOS LAN Requester functions**. These include Microsoft Windows support for Double-Byte Character Set (DBCS) languages; Windows 3.1 support; improved installation, especially from within OS/2 2.0 DOS sessions; and a new Windows user interface.
- **Performance improvements**. Improvements are made throughout the OS/2 2.0, DOS, and Windows Requester programs, to exploit a hybrid connectionless extension to the IBM NetBIOS software. This dramatically improves the performance of many small network operations.
- **OS/2 2.0 Requester enhancements**. These include better support for multiple adapters in bridged

environments, full support for 32-bit OS/2 file API calls, and a new API into the network Domain Control Database (DCDB).

- **Systems management enhancements**. These include SNAPDUMP and First Failure Support Technology (FFST) for easily capturing error information, System Performance Monitor/2 (SPM/2) for analyzing LAN server performance data, and enhancements to the systems management APIs.

All previous features of LAN Server 2.0 are supported, including local security, fault tolerance, OS/2 remote boot, operator privileges, NetView-compliant generic alerter service, Uninterruptible Power Supply (UPS) support, graphical online help information and publications, easy-to-use LAN-aware Workplace Shell, Original Equipment Manufacturer (OEM) hardware support, and the best performance in the industry (see *LAN Server, Netware and LAN Manager: Performance Benchmark Comparison).*

32-Bit Performance

LAN Server 3.0 was a huge challenge to produce since it came only six months after the release of the previous version. Development and design work began even before LAN Server 2.0 was completed. A large, dedicated development group added tens of thousands of lines of code, and performed thousands of man-hours of testing on hundreds of computers.

The center of attention for Version 3.0 was the new 32-bit, high-performance network server software for OS/2 2.0. Forming the heart of the network, the 32-bit server sets a new standard for performance in the industry. Designed to work with OS/2 2.0, the new server is fully 32-bit. It runs at the operating system level (ring zero) and supports up to 64,000 open files and up to 8,000 active directory searches. With multiple network adapters, and depending on the workload and speed of the server hardware, the new server can support up to 1,000 clients.

In Version 3.0, for the first time, LAN Server supports network server cache sizes greater than 16 MB, enabling the server to retrieve information more rapidly. A key performance enhancement was made to the server and requester software programs, so they could send small frames much faster and with much less overhead. Along with improvements to the file server came improvements to fault tolerance, server trace and performance analysis, usability, and reliability.

Many improvements were made for programmers, including additions for error logging and analysis, hooks for SPM/2 performance analysis, IEEE 802.2 and DLR API support while in OS/2 2.0 Virtual DOS Sessions, improvements to the User Profile Management (UPM) API, and a new set of API calls to access the important Domain Control Database.

For Clients of All Kinds

For OS/2 2.0, DOS, and Windows clients, more changes were made than can be listed in a short article. Some of the most exciting changes are described below.

- LAN Server 3.0 has been shown in independent testing to be the highest performing network software program in the industry for OS/2, DOS, and DOS/Windows clients. A major rewrite of the DOS NetBIOS software has produced huge performance savings. A key enhancement to both OS/2 and DOS redirectors has significantly increased the performance of small read and write operations.

  The DOS enhancement involves combining NetBIOS and IEEE 802.2 into a single, efficient protocol stack. This enhancement was part of the OS/2 transport in LAN

Server 2.0. It is now available to DOS clients in LAN Server 3.0.

The performance enhancement common to both DOS and OS/2 clients is in the protocol stack level where guaranteed message delivery is enforced. NetBIOS has a guaranteed delivery interface, which LAN Server has employed in the past for data reads and writes. The highest level of the LAN Server protocol, Server Message Blocks (SMBs), has always had a protection mechanism to detect whether data reads and writes have been performed successfully. The performance enhancement in LAN Server 3.0 involves moving the responsibility for guaranteed delivery from the NetBIOS level to the SMB level in the protocol stack. Mechanisms also were added to revert to the old, more robust protocol structure if the connection between the computers is not sufficient to allow performance savings.

- LAN Server 2.0 provided much-improved coexistence with NetWare, but key Open Data-Link Interface- (ODI-) to-Network Driver Interface Specification (NDIS) network drivers are further improved in Version 3.0. LAN Server 3.0 can run over NetBIOS for TCP/IP, and has improved its interoperability with Microsoft LAN Manager by supporting the LAN Manager 2.1 network dialect.

  As part of the LAN Server 2.0/NetWare coexistence offerings, NetWare includes an NDIS-to-ODI network driver that allows LAN Server and its protocol stack, NTS/2, to communicate over OEM adapters that supported Novell's ODI specification. With LAN Server 3.0, users now can run Novell's IPX protocol stack on IBM and OEM adapters that support the NDIS specification. Therefore, LAN users have more flexibility in choosing the network adapter that best meets their needs.

  With previous versions of LAN Server, multiple LAN segments could be connected using Medium Access Control- (MAC-) level bridges to support LS clients and servers. Users want to connect LS clients and servers on different LAN segments interconnected by Wide Area Networks (WANs). The protocol requested most often for WAN routability is TCP/IP. Not only can LAN Server communicate using TCP/IP, but since it supports multiple protocol stacks, it can communicate simultaneously with local clients and servers (using native NetBIOS flows) and WAN-connected clients and servers (using TCP/IP encapsulated NetBIOS).

- More "applets" (miscellaneous useful application programs) have been included, and many changes have been made to improve their usability.

- Remote Initial Program Load (RIPL) for both OS/2 and DOS has been enhanced.

- The publications and the exhaustive online references have been greatly improved.

Future LAN Software Development

Future networking software products that we expect to see from the IBM LAN development organization in Austin and other locations include the following:

- An improved, high-performance, multiprocessor version of IBM LAN Server

- Apple® Macintosh client support

- Changes to TCP/IP to improve its coexistence with LAN Server

Long-range plans include support for the Open Software Foundation's (OSF's) Distributed Computing Environment (DCE), and many changes to the requester and server programs.

The LAN Server product will continue to be enhanced, integrating OS/2, DOS, and Windows support, and providing superior performance, interoperability, and usability. IBM intends to keep LAN Server at the forefront of network software.

References

LAN Server, Netware and LAN Manager: Performance Benchmark Comparison, LANQuest Independent Network Testing Laboratory, June 1992 and updated October 1, 1992.

IBM OS/2 LAN Server 3.0 Product Overview and Directions, a white paper written by John Sabol available from IBM marketing representatives.

Steven French *is a staff programmer in IBM's Personal Systems Programming Center in Austin, Texas. He is presently the lead developer in charge of the OS/2 LAN Requester. He holds a BA in computer science and an MS in electrical and computer engineering from Rice University in Austin, Texas.*

Gary Hunt *is an advisory programmer in IBM's Personal Systems Programming Center in Austin, Texas. He is presently the lead designer for the IBM LAN Server product. Gary has worked on the OS/2 LAN Server since its inception in 1987, and managed a programming department during development of Releases 1.2, 1.3, and 2.0. He has worked on the IBM DisplayWrite® word processor and its 3270 emulation capabilities, and on the IBM PC Convertible. He holds a BS and an MS in electrical engineering from the University of Texas in Austin.*

The Future of IBM LAN Network Management

Sallie Matlack
IBM Corporation
Research Triangle Park, North Carolina

This article presents the trends and directions of the IBM LAN network management family of products. It is not about software systems management products; it discusses network management and specifically LAN media management. The article also describes the growing network management product relationships that should evolve over the next few years.

IBM's goal is to provide integrated LAN network management from NetView, NetView/6000, and OS/2 distributed systems management platforms. LAN network management provides management for open multivendor, multiprotocol, and multimedia environments. IBM products for network management (as of September 1992) include the following:

- LAN Network Manager Versions 1.0 and 1.1
- LAN Network Manager Entry
- LAN Station Manager
- S/370 and S/390™ NetView
- AIX NetView/6000

All these products work together to complement and strengthen the others' capabilities.

The PS/2-based Distributed Systems Management (DSM) family will be evolving at the same time as the above products. This suite of products offers an open architecture running primarily on OS/2-based workstations. One application that must be provided on this platform to support complete systems management is the network management facilities found in the products listed above. This article outlines the direction for this implementation.

Figure 1 shows IBM Networking Systems direction – how IBM's products fit together to provide a complete and rich set of network management platforms and applications. At the top of Figure 1 are IBM's three systems platforms. At the bottom are LAN components that must be managed. In the center is IBM's LAN Network Manager (LNM), presented as a "super agent" to the platforms. LAN Network Manager provides LAN element management on all three platforms, offering a choice for system and network management centralization. In Figure 1, some lines emanating from the media elements have arrows. The arrow denotes that it is IBM's intent to provide media management with LAN Network Manager in the near future. Lines without arrows denote that the element will be managed by an application residing on one (or more) of the three platforms.

Current Products

LAN Network Manager Version 1.1 is one step in the evolution of IBM media management products whose increasing functionality will satisfy new requirements. These requirements include the following:

- A graphical interface for views of the network
- A scanned representation of the IBM 8230 Controlled Access Unit that clearly shows the plugged adapters and their active or disabled status
- A command-line interface that enables local (from the LNM console) or remote (from a connected NetView host) automation
- An extended (over the function of LNM 1.0) command interface to enable complete management of the LAN network from NetView
- User-specified local filtering of logged events
- Printing of the event logs
- Enhancements to the management of IBM's 8209 Token-Ring to Ethernet/IEEE 802.3 bridge

Program Temporary Fix (PTF) UR37165 was made available for LAN Network Manager 1.0 to support two network configurations of token rings on the other side of the IBM 6611 Multiprotocol Router. LNM 1.1 addresses the following areas of management:

- A token-ring segment containing an IBM 8230 Controlled Access Unit connected to another ring with LAN Network Manager installed, using the 6611-6611 bridging function
- A token-ring segment connected to a LAN Network Manager segment via an IBM 6611 Multiprotocol Router and the IBM Token-Ring Bridge Program 2.2 (with PTF UR37051)

The second function is supported only if the special bridge PTF noted above was installed on the bridging workstation of the remote segment. It allows LNM to link to the far-side adapter of the bridge. In the first half of 1993, LAN Network Manager Version 1.1 will provide similar support for the IBM 6611 as the LNM 1.0 PTF. IBM's direction is to provide

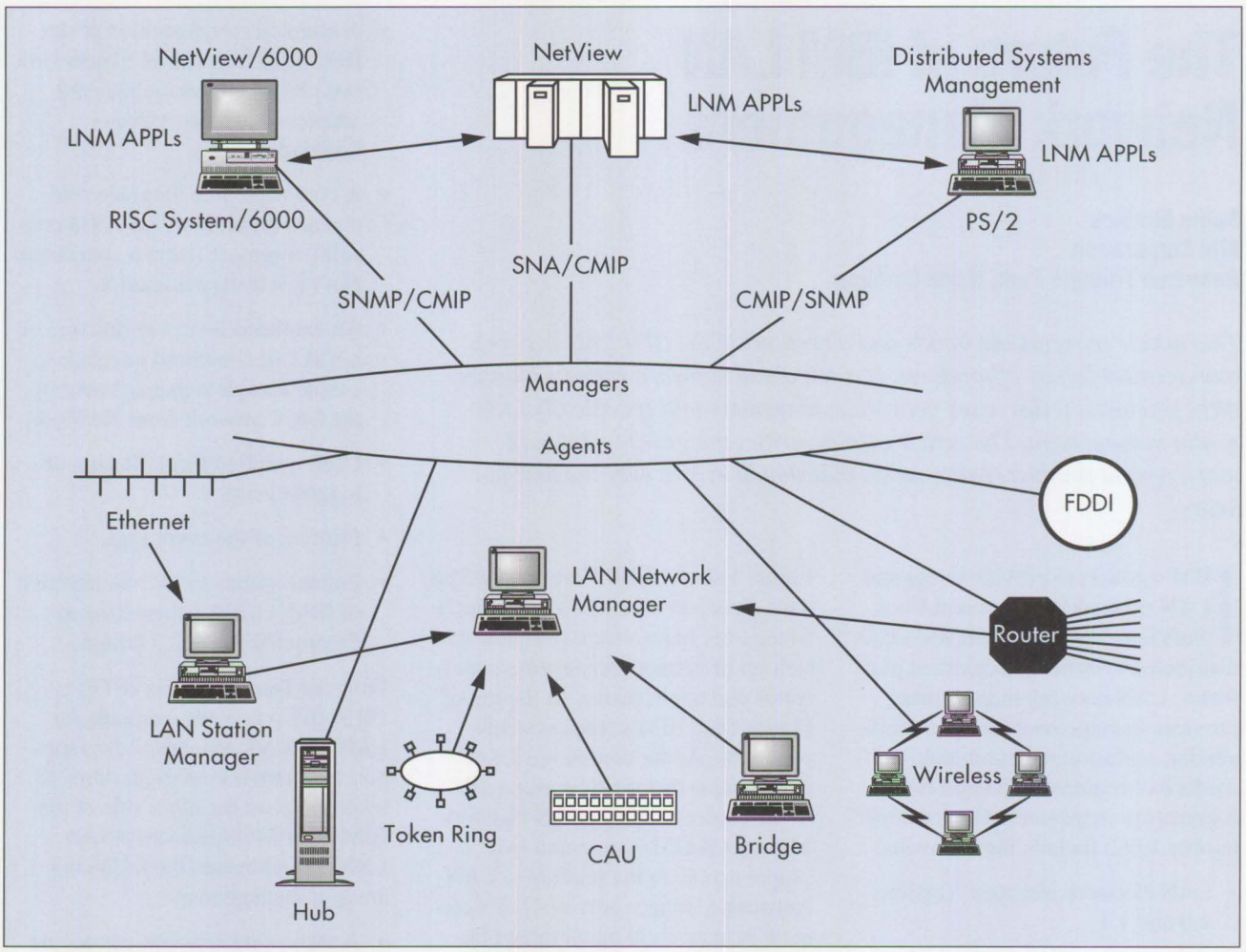


Figure 1. IBM LAN Management Direction

this bridge support within the router, and to allow the router to appear as a virtual segment to LAN Network Manager. This would provide the same capabilities as LNM 1.0 has with the linking of an IBM bridge.

LAN Station Manager (LSM) supports its CMOL (Common Management Information Protocol or CMIP over the Logical Link Control or LLC) layer of the IEEE 802.2 standard protocol interface to LNM 1.1. Vital product data for both DOS and OS/2 workstations can be added to the LNM database. Management Information Base (MIB) attributes sent to LNM from LSM include physical configuration information such as the following.

- Adapter types in the workstation
- Operating system type and version
- Version of the LAN Station Manager code
- Type and size of disk
- Amount of memory installed
- Token-ring primary or secondary adapter indication
- Adapter address (universal or locally administered)
- Functional or group address (for token ring)
- Microcode level of token-ring cards
- Ring utilization of the local token-ring segment

Administrators can add information to the LSM configuration file such as physical location of the workstation; display, printer, and keyboard types and serial numbers; and other user-defined data. The workstation's MIB is sent to LAN Network Manager on request to be stored in the manager's database. This allows host applications to perform asset management and to generate reports for controlling such items as back-level adapter cards or locations of DOS operating stations.

LAN Station Manager, working with the 8230 Controlled Access Unit (CAU) and the LAN Network Manager 1.1 (or LAN Network Manager Entry), provides a new level of LAN

security to central sites. Through NetView or from a controlling LAN Network Manager console, administrators can keep track of the location of LAN stations and control their access to the LAN.

NetView Host as Manager of Managers

NetView is the premier IBM solution for managing mixed Systems Network Architecture (SNA), Open Systems Interconnection (OSI), and Transmission Control Protocol/Internet Protocol (TCP/IP) multivendor networks. It also is a focal point for collecting and correlating information from various distributed network management systems, including LAN Network Manager, AIX NetView/6000, and the OS/2 distributed systems management platform products. Today, a bidirectional flow of alerts and commands between LNM and NetView is supported through an SNA session (SSCP-PU link), giving NetView a central control capability.

LAN Network Manager will continue to provide the current interface to NetView, but in the future it will enrich the exchange of information with CMIP frames packaged within SNA packets, that is, Common Management Information Protocol Over SNA. These frames will provide topology and network status not currently available, so that an application under NetView can display both the physical and logical views of the LAN.

NetView on the RISC System/6000

AIX NetView/6000 extends the NetView family of products by providing powerful management capabilities for devices communicating over TCP/IP supporting the Simple Network Management Protocol (SNMP). The features announced in September 1992 for this product include the following:

- The industry-standard X Management Protocol (XMP) Application Programming Interface (API) for application support based on the Open Software Foundation (OSF) Distributed Management Environment (DME) technology
- The Motif® user interface
- SNMP APIs which, when combined with the user interface, allow seamless integration of vendor and user applications
- Open topology enablement for devices on networks other than TCP/IP to be managed

These enhancements position AIX NetView/6000 as a platform for consolidating multivendor and multiprotocol network management.

Beginning in 1993, IBM will roll out a rich set of network management applications on the AIX NetView/6000 platform. These include the 8250 Hub Management Program (available in December 1992), and a LAN Manager application for the 8240 Fiber-optic Data Distribution Interface (FDDI) Concentrator and adapters. The IBM 6611 Multiprotocol Router also will be managed from NetView/6000. To provide a complete package of LAN management applications on the AIX NetView/6000 platform, IBM plans to offer a consolidated network view of the user's LAN media. This will include the SNMP devices mentioned previously and the token rings (including IBM's bridges and concentrators) as currently managed from the console of IBM's OS/2-based LAN Network Manager (or from NetView). This will be supported through a feature in LAN Network Manager and a new NetView/6000-enabled application that interfaces with this feature.

LNM and Distributed Systems Management

IBM's distributed systems management product family allows administrators to choose the centralized platform for network and systems management. Both the OS/2-based distributed systems management family and AIX NetView/6000 will offer XMP APIs and user interfaces as well as multiprotocol support. IBM Manage/2 Version 1.0 runs on OS/2 2.0-based workstations. IBM has announced a suite of related products including agent applications that can collect systems management information from workstations running OS/2 2.0, DOS 5.0, and DOS 5.0 with Windows 3.1. The logical complement to the operating systems and software management is network management applications that can run on the PS/2 platform to support these same agent workstations.

Initially, LNM must run on the same workstation as Manage/2 to have the appearance of coexistence. There is no interfacing or sharing of resources. The next step is for LAN Network Manager to feature a CMIP proxy that will interface with the OS/2 products to provide topology and network status information. This will give View/2 a consolidated view of both the physical and logical networks.

LAN Network Management Product Direction

IBM's network management strategy is to continue to support either centralized or decentralized network management independent of the operating system platform.

With the current user investment in the OS/2-based LAN Manager and LAN Network Manager products, and with the requirement for a central console management approach to network management, the next logical step in the evolution of LNM is to provide interface support from this product that will move its database information on request, and send its alerts to the user's platform of choice.

LAN Network Manager will offer features that provide LNM management domain interfaces to the distrib-

uted management platforms (NetView, AIX NetView/6000, or the OS/2-based distributed systems management family). There also will be enhancements to the LAN media view at NetView and a NetView/6000-enabled application that can exploit the new interface to the LNM-managed domains of token rings, IBM bridges, and concentrators.

Expanding the Station Management Domain

IBM, Digital Equipment Corporation, and Hewlett-Packard joined the Desktop Management Task Force (DMTF) in July 1992 teaming with the original charter members – Intel, Microsoft, Novell, SunConnect®, and SynOptics® Communications. DMTF's objective is to develop a unified method for gathering information and managing components within PC workstations. The group is focusing on two primary areas of development. The first is to construct an API to facilitate a common method of accessing PCs on the LAN. The second is to define a simple way for software and hardware developers to allow their products to be managed by applications calling the API.

IBM's LAN Station Manager is the first product to use the new Heterogeneous LAN Management (HLM) protocol for exchanging information between agents and managers over the LLC layer of the IEEE 802.2 standard. The source code of the HLM kernel, which is contained within LAN Station Manager, is being marketed by IBM to other vendors. Another IBM implementation of the HLM kernel will be within the agent platform of the IBM PS/2 distributed systems management family. IBM's DOS and DOS+Windows agent schemes will be supplemented to support industry standards as they continue to evolve. IBM also is actively pursuing making HLM/CMOL an integral part of the DMTF referenced implementation. IBM will demonstrate the DMTF in the spring of 1993.

IBM LAN Network Management Direction

September 1992 — IBM's goal is to provide integrated LAN network management from NetView, NetView/6000, and OS/2 distributed systems management platforms.

The next versions of LAN Network Manager will provide the media management functions of LAN Network Manager Version 1.1 on PS/2s and OS/2-based workstations, plus feature an SNMP and a CMIP agent for other management platforms. LAN Station Manager will continue to provide station management through LAN Network Manager.

During the same time, IBM also will introduce a set of applications on the AIX NetView/6000 platform that will take advantage of the support offered by the LNM proxy agent. This will provide a single point for LAN media management and SNMP-managed devices on the NetView/6000 platform, including the IBM 8250 Intelligent Hub and FDDI, including the IBM 8240 FDDI Concentrator.

HLM/CMOL (IEEE 802.1B) allows MIBs to be transported on protocols other than IEEE 802.5 (the 1992 implementation). The network management strategy not discussed so far is the obvious challenge to expand the LAN Network Manager domain of workstation management by developing a LAN Station Manager for other media. The two obvious possibilities are Ethernet and FDDI. IBM intends to carry workstation management first into the IEEE 802.3/Ethernet LAN.

Conclusion

The following summarizes the major points regarding IBM's LAN Network Management:

- There will be three IBM platforms of choice: NetView, AIX NetView/6000, and the PS/2 DSM product family.
- All three platforms will evolve into complete managers of networks.
- NetView will remain the strongest contender for manager of both physical and logical networks – particularly in the Advanced Peer-to-Peer Networking (APPN) arena – with the other two platforms optionally feeding their information to the mainframe host.
- CMIP is still IBM's protocol of choice in managing LANs.
- IBM's LAN Network Manager will be the LAN media manager underneath all three platforms.
- Applications will be provided to bring the controlling, graphically presented network management of LNM up to both the AIX NetView/6000 platform and the OS/2-based DSM platform (although these functions will not roll out in parallel).

Sallie Matlack *is a consulting market support representative in Network Systems and Management, Services and Support, in IBM Research Triangle Park, North Carolina. She joined IBM in 1967 after earning a BS in mathematics from the University of South Carolina. Sallie has spent most of her career with IBM as a development programmer in a marketing headquarters group doing custom development work.*

Understanding and Using the Workplace Shell

Rob Talley
IBM Corporation
Dallas, Texas

This article is a follow-up to the author's article, "The OS/2 Workplace Shell," which appeared in the April 1992 issue of this publication. That article discussed many exciting features of the OS/2 2.0 Workplace Shell. This article provides some suggestions for customizing your desktop to exploit these features.

The OS/2 2.0 Workplace Shell (WPS) is a fourth-generation user interface that is easy to work with and has tremendous capabilities. It allows users to work the way they want to work. Object orientation makes Workplace Shell easy to use. WPS was designed to recognize the importance of everyday work – documents, spreadsheets, and charts in data files on a computer – so that users can create objects that reflect the way they do their jobs.

On the OS/2 2.0 Desktop, items such as programs, files, printers, documents, and letters are called *objects*. These objects appear on the screen as pictures, or *icons*. OS/2 classifies objects into the following types:

- A *data-file* object contains information. Examples include text files, spreadsheet data, memos, letters, documents, image files (drawings), video, and sound.
- A *program* object represents an executable program file. Examples include word processors, text editors, terminal emulators, spreadsheets, and games.
- A *device* object represents a physical system component. Examples include printers, disk drives, and CD-ROM devices.
- A *folder* object is a container for other objects. When one folder contains another folder, the second folder is called a *subfolder*. Experienced OS/2 1.X and Microsoft Windows users can think of a folder as similar to a group, although folders have more functions than groups. A folder also is similar to a directory, and a subfolder is similar to a subdirectory.

Organization of the Workplace Shell Desktop

OS/2 2.0 Workplace Shell's Desktop is where users interact with the operating system. It is a special folder that fills the entire screen and contains all other folders and objects. Let us look at how the Desktop is represented on the computer.

On the Desktop, locate the OS/2 System folder. Place the pointer on it and double-click with mouse button 1 to open the folder. Inside the OS/2 System folder, open the Command Prompts folder in the same manner. Next, open the OS/2 Window icon to bring up a windowed OS/2 command prompt. At the OS/2 command prompt, type `DIR /N` and then press Enter to list files in the root directory of the drive where OS/2 is installed. The `/N` switch shows the size of a file's Extended Attributes (EAs) on a File Allocation Table (FAT) drive. EAs are displayed automatically with a `DIR` of a drive that was formatted with the High-Performance File System (HPFS).

For this article, assume that the default drive, C:, is a FAT drive and that all file and directory names conform to the standard 8.3 naming convention. The term *8.3* means that eight characters is the maximum length for a file name, and three characters is the maximum length for the extension. (If your system uses an HPFS drive as the default drive, you can use and view long file names.)

The Desktop is represented on the physical drive directory `C:\OS!2_2.0_D`. (Note that the directory name includes a three-character extension (`0_D`) that conforms to the 8.3 naming convention for FAT file systems.)

If the drive had been formatted using HPFS, its directory name would be `OS!2 2.0 DESKTOP`. (The / character is changed to `!` on the drive because OS/2 does not allow using / in a file or directory name.) If a long file name contains one or more blank spaces, enclose it in quotes when using commands; for example:

```
CD \"OS!2 2.0 DESKTOP"
```

Inside the `\OS!2_2.0_D` directory (on the FAT system), there are subdirectories that represent all container objects on the Desktop. For example, the Desktop has a system folder called OS/2 System. This folder is represented physically on a FAT drive as subdirectory `\OS!2_2.0_D\OS!2_SYS`. (Since there is no period (`.`) delimiter within the object's name, the subdirectory name is truncated after eight characters.) The OS/2 System folder on the Desktop has other folders inside that are represented as subdirectories within the `\OS!2_SYS` subdirectory.

Figure 1. OS/2 Command Prompt with Directory of Desktop

Looking under the Hood

To better understand what is under the "hood" of the Workplace Shell, first create some new objects. Then look at what is happening in your OS/2 2.0 system.

Using a template is the best way to create new objects in OS/2 2.0. Double-clicking on the Templates icon opens the corresponding folder and reveals several default templates that can be used to create your own objects. Think of each template as a tear-off pad of blank objects. To use one, simply place the pointer over the template you want to use, "grab" it using mouse button 2, and drag it to the location where you want to create a new object.

Data Files

Of all object types, data files are the easiest to explain.

Drag a Data-File template from the Templates folder to the Desktop. (Use mouse button 2 to drag objects.) This action creates an object that resembles a sheet of paper with one corner folded over. By default, this new object's name is Data File.

From the OS/2 command prompt and from within the `\OS!2_2.0_D` directory, use the `DIR /N` command to view the subdirectory structure that is shown in Figure 1. In Figure 1, notice that there is a file (with a size of zero bytes) called `DATA_FIL`. Next, return to the Desktop and name this data file by using direct name-editing feature. To do this, place the mouse pointer over the name of the Data-file object (named Data File because you just created it). Press and hold the Alt key, then click mouse button 1. This changes the name area to an entry field that you can edit. Type `My Data File` as the new name for the data file, then move the mouse pointer from the name field and click mouse button 1.

Return to the OS/2 command prompt and again use the `DIR /N` command. The physical name of the file has changed from `DATA_FIL` to `MY_DATA_`.

Folders

Drag a Folder template from within the Templates folder to the Desktop. This action creates an object that resembles a manila folder. By default, this new object is named Folder. View the `\OS!2_2.0_D` directory again and notice a subdirectory called `\FOLDER`, which is the new one just created.

Rename the folder to My Folder using the name-editing feature discussed previously. Viewing the file structure in the `\OS!2_2.0_D` directory reveals that the `\FOLDER` subdirectory has been renamed to `\MY_FOLDE`. The subdirectory name has been truncated to eight characters.

Program Objects

Program objects are more complicated than folders and data files. Looking inside the Information folder, you will see a `README` data-file object, the Tutorial, Glossary, Command Reference, and REXX Information objects (assuming that you chose to install all these options). However, entering the `DIR` command from within the `C:\OS!2_2.0_D\INFORMAT` subdirectory gives the impression that there is nothing inside the subdirectory. This is because the `README` object is actually a shadow of the `README` file that is located in the root directory of drive C:. The shadow is created within the EAs of the `\INFORMAT` subdirectory. These EAs contain information pointing to the `README` data file. In Figure 1, the `\INFORMAT` subdirectory has 3,988 bytes of EAs.

In FAT file systems, EAs are stored in a hidden system file in the root directory of each FAT partition. The file's name is `EA DATA. SF` (note the blank spaces in the file name and extension). The Workplace Shell also uses EAs to store information about the Desktop setup. These attributes are stored in another hidden system

file, WP ROOT. SF, which is located in the root directory.

The icons created for system objects actually reside in the OS2.INI file. They are created during the installation of OS/2. Pointers to these icons are part of the data stored in EAs of the directories and subdirectories.

"Drop" a Program template onto the Desktop. The Settings view of the object automatically opens, enabling you to type in the program's path, file name, and other parameters. Close the Settings notebook without updating the program information. Note that this new object is called Program.

This new program object will not be found in the \OS!2_2.0_D subdirectory. That is because this object and its settings are located in the EAs of the \OS!2_2.0_D subdirectory.

The EAs do not necessarily grow each time a new object is added to the Desktop, because space for the EAs is allocated as a predetermined number of bytes. As this area is filled, another empty block of bytes is added to the EAs.

Device Objects

All information regarding Device objects, such as the Printer object or Drive object, is stored in the OS2.INI file, OS2SYS.INI, or in

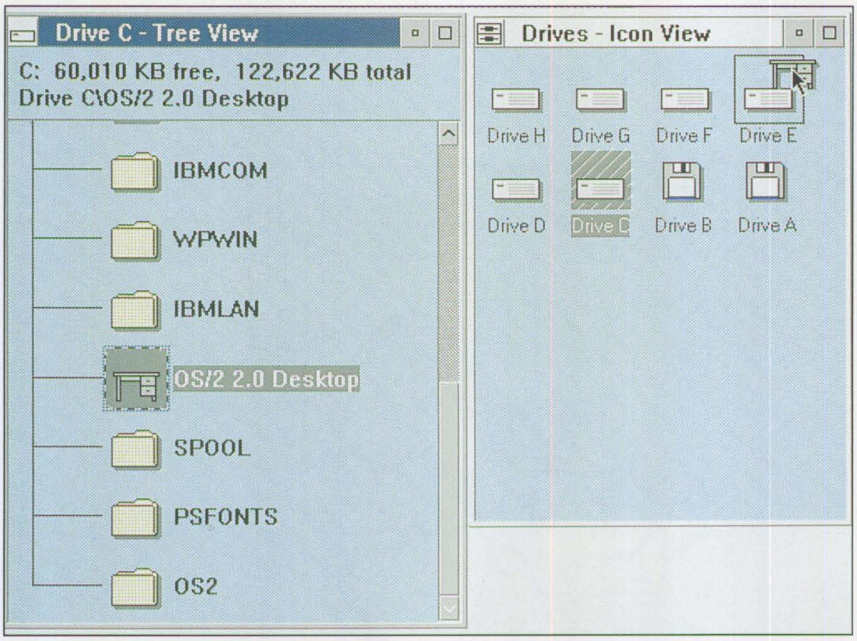


Figure 2. Relocating the OS/2 2.0 Desktop

EAs. For example, the printer object information is stored in `OS2.INI`. Other printer information, such as the port and queue to which the printer is attached, is located in `OS2SYS.INI`.

Notebook settings for Drive objects are stored in EAs that are associated with the `\OS!2_2.0_D\OS!2_SYS\DRIVES` subdirectory. Notebook settings for the Drive C object are stored in `WP ROOT. SF`, a hidden system file located in the root directory of drive C:. The settings for each Drive object are stored in the Drive object's own copy of `WP ROOT. SF` located in the root directory of the drive represented by the Drive object. Icons for Drive objects, however, are stored in the `OS2SYS.INI` file.

Performance Implications

Almost everything created on the Desktop is created physically on the default drive where OS/2 is installed. In a constrained environment, when the default drive is filling up, you may want to avoid placing more data on it. For example, you can create new folders that will contain many data files by dropping the Folder template onto the appropriate Drive object rather than onto the Desktop. Then you can create shadows of your data on the Desktop or in a folder on the Desktop, using techniques presented in the April 1992 article.

Using a Second Drive

Sometimes it makes sense to have all data and objects appear directly on the Desktop, even though the default drive is constrained by its size. In this situation, you can relocate the Desktop to another logical (or physical) drive. To do this, open the Drive object located in the OS/2 System object. Open the default drive by double-clicking on the appropriate Drive object (normally Drive C:). Next, move the open Drive object so that you easily can see both the open default drive window and the open Drive object, as shown in Figure 2. Place the pointer over the Desktop object on the default drive. Press and hold the Alt and Shift keys, and then press and hold mouse button 2. Drag the Desktop object to the Drive object where you want to relocate the Desktop (Drive E: in Figure 2) and release mouse button 2 and the keyboard keys. This action moves the Desktop folder to the new drive. After the process is complete, you can see the Desktop object disappear from the default drive and appear on the second drive. From now on, all items created on the Desktop will actually be created on the second drive.

Remote Initial Program Load

Moving the Desktop to another drive may also improve performance when dealing with Remote Initial Program Load (RIPL) systems that have a local drive on the remote workstation. As objects are manipulated on the Desktop, the WPS updates the `OS2.INI` file to reflect the current location and state of each object on the Desktop. In a RIPL workstation, these updates must be transmitted across the network, which slows down the response time of the WPS.

Suppose a RIPL workstation is originally defined (using the LAN full-screen interface) as `USER1`. To obtain better response time from the WPS on a RIPL workstation, perform the following steps:

1. Using the procedure outlined above, at the workstation, drag the Desktop from Drive C:, which is physically located on the RIPL server, to Drive D: on the workstation. This action affords better response time, but there will still be the overhead of writing small blocks of data to update the `OS2.INI` file on the server.
2. Shut down the RIPL workstation and turn off its power. If the power is not turned off after shutdown, there is still an open session linking the workstation with the RIPL server. The RIPL code within the server continues to behave as though OS/2 is active within the `USER1` subdirectory, so the `OS2.INI` and `OS2SYS.INI`

files on the server remain open. You cannot copy them.

3. At the command prompt on the RIPL server, use the CD command to change the directory to reflect the following path:

```
CD \IBMLAN\RPLUSER\USER1\OS2
```

   Here, USER1 is the name of the RIPL workstation.

4. Copy the OS2INI.20 file to OS2.INI, and copy OS2SYINI.20 to OS2SYS.INI as shown in Figure 3.

   The OS2INI.20 and OS2SYINI.20 files are the initialization files that are used for each individual RIPL computer. By changing the names of these files, the RIPL workstation's CONFIG.SYS can recognize the files.

5. Reboot the RIPL workstation, open an OS/2 window, and create a directory called \MYOS2 on the requester's local hard drive (D:) by entering the following:

```
MD D:\MYOS2
```

6. Copy the OS2.INI and OS2SYS.INI files from the root of drive C to the D:\MYOS2 directory, as follows:

```
COPY C:\OS2*.INI D:\MYOS2
```

7. On the RIPL server, edit files that are located in the \IBMLAN\RPL\FITS and \IBMLAN\RPL\MACHINES\ requestername subdirectories. If the requestername is USER1, do the following:

   - Edit the \IBMLAN\RPL\FITS\ USER1.FIT file. Use a semicolon to comment out lines that point to the location of the OS2.INI and OS2SYS.INI files as shown in Figure 4. For example, the server name could be DALLAS.

```
COPY OS2INI.20 D:\IBMLAN\RPLUSER\USER1\OS2.INI
COPY OS2SYINI.20 D:\IBMLAN\RPLUSER\USER1\OS2SYS.INI
```

Figure 3. Copying Files

```
;C:\OS2\OS2.INI\\servername\WRKFILES\USER1\OS2\OS2INI.20
;C:\OS2\OS2SYS.INI\\servername\WRKFILES\USER1\OS2\OS2SYINI.20
```

Figure 4. Editing the \IBMLAN\RPL\FITS\USER1.FIT File

   - Edit the CONFIG.20 file located in the \IBMLAN\RPL\ MACHINES\USER1 subdirectory. Change the following lines:

     from:

```
SET USER_INI=C:\OS2\OS2.INI
SET SYSTEM_INI=C:\OS2SYS.INI
```

     to:

```
SET USER_INI=D:\MYOS2\OS2.INI
SET SYSTEM_INI=D:\MYOS2\OS2SYS.INI
```

8. Reboot the RIPL workstation. At this point, the workstation is using the OS2.INI and OS2SYS.INI from the RIPL workstation's local hard drive.

Now all updates to the Desktop occur locally and do not require access to the RIPL server. The result is improved user response time from the Workplace Shell.

Backing Up the Desktop

The key to maintaining a backup of the Desktop for disaster recovery situations lies in maintaining a copy of the proper directory structure including all EAs, and keeping copies of the latest CONFIG.SYS, OS2.INI, and OS2SYS.INI files. By not making periodic backups of these crucial files, you risk losing all customization to OS/2 since installation.

Originally, these files were copied into the \OS2\INSTALL subdirectory during the OS/2 installation process. After customizing the Desktop, keep backup copies. The *OS/2 Installation Guide* (84F8464, packaged with OS/2) outlines processes for making backup copies of these (customized) critical files. You can directly back up the files into the \OS2\INSTALL subdirectory (overlaying the previous files) during system boot, or you can periodically copy the backed-up .INI files into the \OS2\INSTALL subdirectory. The risk of not backing up the Desktop periodically is that you may lose recent updates if your .INI files become corrupted.

After copying the customized files into the \OS2\INSTALL subdirectory, you can restore them to their proper locations. To do this, restart the system and then press the Alt and F1 keys simultaneously while the system is rebooting. (For more information, see the README file in the root directory of the default drive.) This process copies the OS2.INI and OS2SYS.INI files from the \OS2\INSTALL subdirectory into the \OS2 directory, copies the CONFIG.SYS file into the root directory of the default drive, and renames the STARTUP.CMD file to STARTUP.BAK.

To protect the Desktop completely, also keep a backup copy of the C:\OS!2_2.0_D directory. Although the BACKUP utility allows backing up this directory, it did not back up empty directories in the initial release of OS/2 2.0. As a result, some objects

on the Desktop were not backed up. BACKUP has been fixed in the first ServicePak (XR06055) for OS/2 2.0. With the ServicePak installed, you can back up the Desktop directory using the following procedure:

1. Boot from the OS/2 installation diskette and insert diskette 1 when prompted.
2. Escape from the installation program when prompted.
3. Enter one of the following commands from the command prompt in a FAT drive:

   ```
   BACKUP OS!2_2.0_D A: /S
   ```

   For an HPFS drive enter the following:

   ```
   BACKUP "OS!2 2.0 DESKTOP" A: VS
   ```

4. Use XCOPY to back up the OS2.INI and OS2SYS.INI files:

   ```
   XCOPY C:\OS2\*.INI A:
   ```

   To restore the Desktop, perform Steps 1 and 2 above, and then enter the following commands in a FAT drive:

   ```
   RESTORE A: C:\OS!2_2.0_D /S
   XCOPY A:\*.INI C:\OS2\*.*
   ```

   Enter the following for an HPFS drive:

   ```
   RESTORE A: C:\S
   XCOPY A:\*.INI C:\OS2\*.*
   ```

If the ServicePak is not installed and the system is installed on a FAT drive, use XCOPY to copy the contents of the OS!2_2.0_D directory to diskette. If the system is installed on an HPFS drive, use XCOPY to back up the Desktop to another HPFS drive. This backup can be done as follows:

1. Boot from the OS/2 installation diskette and insert diskette 1 when prompted.
2. Escape from the installation program when prompted.
3. Enter the following commands from the command prompt:

   ```
   XCOPY C:OS2!2.0_D A:\OS2!2.0_D /E /S
   XCOPY C:\OS2\*.INI A:
   ```

To restore the Desktop, perform Steps 1 and 2 above, then enter the following commands:

```
XCOPY A:\OS2!2.0_D C:\OS2!2.0_D /E /S
XCOPY A:\*.INI C:\OS2\*.*
```

To simplify the above procedure, include the following line as the last line in the CONFIG.SYS file:

```
CALL=C:\OS2\CMD.EXE
```

OS/2 2.0 has pop-up menus for each folder and object, including the OS/2 Desktop itself.

This causes a full-screen command processor to be loaded after OS/2 has started, but before Presentation Manager and the Workplace Shell are loaded. At this point, none of the .INI files or system files are in use, so now complete Step 3 above. After completing your backups, type EXIT at the command prompt. This closes the full-screen command processor and enables the system to continue to boot. Later, bring up your editor and place the characters REM in front of the CALL= statement. The next time you boot the computer, the full-screen command processor will not start.

Customizing Your Startup Options

The Workplace Shell saves all changes to objects. Some changes are saved immediately as you make them and others are saved when the object is closed. This preserves the look of the Desktop and enables you to bring up the system in the same state as when it was previously shut down. This behavior is convenient for many users. However, others prefer to start their OS/2 system with a predetermined status, regardless of which applications were running when they last shut down.

To customize the startup of the system, add the following line to the CONFIG.SYS file:

```
SET
RESTARTOBJECTS=STARTUPFOLDERSONLY
```

(This parameter and others are documented in the \README file installed in the root of the boot drive by the ServicePak.) Place a shadow of any Program objects that you want to start automatically into the Startup folder located in the OS/2 System folder. To create a shadow of an object, first place the pointer over the object, then press and hold mouse button 2. Next, press and hold both the Ctrl and Shift keys while dragging the object to the location where you want the shadow placed. (Note the line connecting the original object to the shadow object that you are dragging.) When you have placed the shadow object where you want it, release mouse button 2 first, then release the Ctrl and Shift keys. You now have created the shadow. The gray (default color) text under the shadow object identifies it as a shadow.

Now your system starts only the programs that you have shadowed in the Startup folder.

Adding Programs to Your Menus

OS/2 2.0 has pop-up menus for each folder and object, including the OS/2

Desktop itself. A pop-up menu is a context menu – it contains the actions and choices available for the specific type of object and reflects the current "state" of the object. You can customize these menus by adding other entries of your own.

Suppose you want to add the Calculator utility to the pop-up menu for the OS/2 System object. (This makes it easy to start the Calculator directly from the Desktop without having to open several folders to find the Calculator program object.) First, open the Settings notebook for the OS/2 System object by placing the pointer over that object and clicking once with mouse button 2. Now press the action arrow on the Open Menu item and select the Settings item. Next, select the Menu tab to bring up the Menu Settings page, as shown in Figure 5.

To create an action item in the OS/2 System object's pop-up menu, select the Create Another pushbutton located on the Available Menus section of the page. This action causes a Menu Settings pop-up window to display. Enter the name (for example, My Utilities) that you want to appear in the pop-up menu.

Notice the Menu Type area with two radio buttons: Cascade menu and Conditional Cascade menu. These buttons enable you to determine the behavior of this menu selection. Before leaving this screen, select which type of menu to create.

The difference between a Cascade menu and a Conditional Cascade menu needs explanation. If you click mouse button 1 on the arrow button to the right of My Utilities in the OS/2 System pop-up menu, the Conditional Cascade menu shown in Figure 6 appears to the right of the pop-up menu. A Conditional Cascade menu always has a default action,

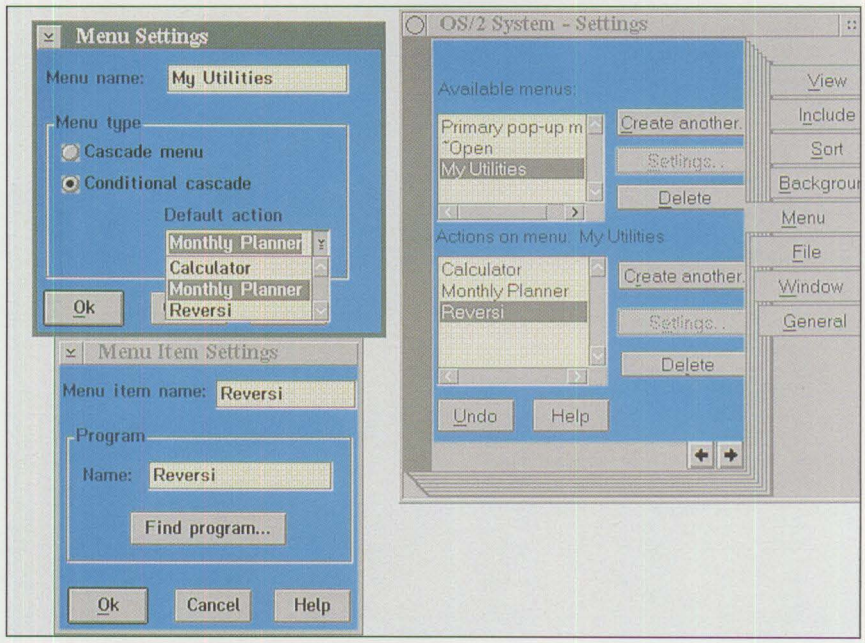


Figure 5. Adding Programs Using the Menu Settings

which is the top choice in the list. You can override this choice by setting the default action from within the Menu Settings pop-up menu shown in Figure 5. In Figure 6, although the top choice is Calculator, Monthly Planner is the default action, so the Monthly Planner choice has a check mark next to it. A Conditional Cascade menu is displayed only if you select (using mouse button 1) an arrow pushbutton next to the title of a selection in the pop-up menu. If you select the title itself, the default action starts immediately.

In contrast, a Cascade menu appears when you select a choice in the pop-up menu that has no corresponding arrow button. Unlike a Conditional Cascade menu, a cascade menu has no preselected or default action. You must specifically select which action to take.

After creating the new menu item for My Utilities, its name appears in the Available Menus list box. Select this item by placing the mouse pointer over the name in the list box and

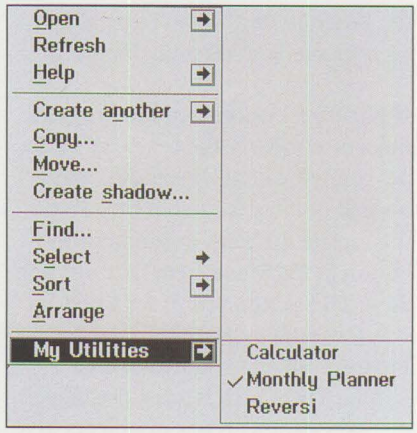


Figure 6. Conditional Cascade Menu

pressing mouse button 1. Now select the Create Another pushbutton located in the `Actions on menu: My Utilities` section of the page. This action causes a Menu Item Settings pop-up window to appear. In this window, notice the two entry fields. Enter the name that you want in the Cascade menu that appears whenever My Utilities is selected from the OS/2 System pop-up menu. The next field represents the actual name of the executable file you are using or

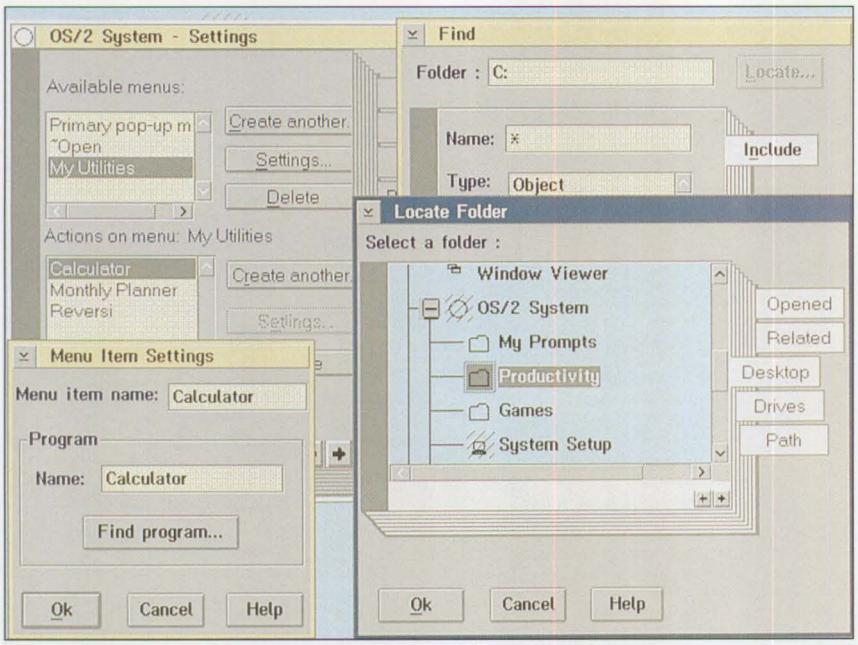


Figure 7. The Locate-Folder Window

the Program object that represents the executable file on the Desktop.

If you have customized the Program objects to modify the behavior of the program, locate the Program object instead of the executable file. To do this, select the Find Program pushbutton in the Menu Item Settings window. This action brings up a pop-up window with a data field called `Folder:` and an entry field called `Name:`. There also is a list box that displays various types of objects to locate. Scroll through this list box and notice that `Program` has been selected already.

Now select the Locate pushbutton. This action displays a Locate Folder window (shown in Figure 7) that allows you to change the folder or drive where the search starts. Since the program object you want is defined as an object on the Desktop, choose the Desktop tab. Now scroll the displayed tree view of the Desktop and select the folder that contains the program object. In the example, Calculator is located within the Productivity folder inside the OS/2 System folder. Now select the OK pushbutton. This action returns you to the Find pop-up window. Here radio buttons let you select whether to search all subfolders or just the folder where the search begins. After you press the Find button, a Find Results window appears, displaying all Program objects in all folders and subfolders on the Desktop.

Locate the Calculator icon by scrolling through the Find Results window. Select the Calculator icon with mouse button 1, then press the OK pushbuttons in both the Find Results window and the Menu Item Settings window. At this point, the menu items have been created, and you can close the OS/2 System Settings notebook. The Calculator utility now can be started from the OS/2 System pop-up menu by selecting the My Utilities button in the menu.

Other applications can be set up similarly. The object itself is passed as a parameter to any executable program that you place in the object's pop-up menu. If you start an application that accepts a command-line parameter, the application may indicate an error when trying to open anything other than a Data-file object. (This may cause problems with the application.) In addition, some applications, such as Solitaire and the OS/2 Enhanced Editor, use their own `.INI` files. When starting one of these applications the same way that Calculator was started in this scenario, a new `.INI` file can be created on the Desktop, or you may receive an error message saying that a file could not be opened. These error messages, however, do not affect the application. Avoid creating the `.INI` file on the Desktop by adding a path to the Working Directory entry field in the Settings notebook for the application that you are adding to the menu. Then the `.INI` file is created in the specified working directory.

Using Data Files

Suppose you want to use WordPerfect® for Windows as your editor for a Data-file object named `FORMLTR`. You have a Program object that starts WordPerfect running on the OS/2 Desktop instead of in a WIN-OS/2 full-screen session. In addition, you want to edit the Data-file object from the OS/2 Desktop by using the Program object.

Use the application to create and save the file in your working subdirectory. You may want to customize headings, footings, page layout, and other aspects of the letter. Now open the appropriate Drive object, and then open the folder that contains the `FORMLTR` file.

Place the mouse pointer on the `FORMLTR` object that you want to edit with WordPerfect and click mouse button 2 to bring up the contextual (pop-up) menu. Open the Data File Settings notebook and select the Menu tab as you did with the OS/2

System menu above (refer to Figure 5). This time, however, select the Open menu in the Available Menus list box. Now, use the Create Another pushbutton under Actions on Menu. Select Open to bring up the Menu Item Settings window. This time, enter `WordPerfect` as the Menu Item Name and then use Find to locate the Program object on the OS/2 2.0 Desktop. This associates the WordPerfect Program object and its settings – not the executable file – with the Menu Item. Now go back to the Menu Settings window and select the pull-down object in the Default Action entry field. Use mouse button 1 to select WordPerfect as the default action. The remaining steps are the same as the steps taken with Calculator above.

The next time you select this Data-file object by double-clicking on its icon, WordPerfect for Windows will be started as an application running on the OS/2 Desktop, and the `FORMLTR` data file will be loaded automatically.

Templates

Suppose that you have the `FORMLTR` object discussed above, and that the `FORMLTR` data file contains a logo that you want to use for your letters. Once assured that the data file contains the correct information, make the `FORMLTR` object into a template to use the contents of the data file again.

Open the Settings notebook for the data file and select the General tab. On the General page is a Template check box. Select this check box and close the Settings notebook. Notice that the `FORMLTR` object now resembles the other templates that are located in the Templates folder on the Desktop. This template can be used to create new data files on the Desktop that automatically launch WordPerfect and load the `FORMLTR` data file.

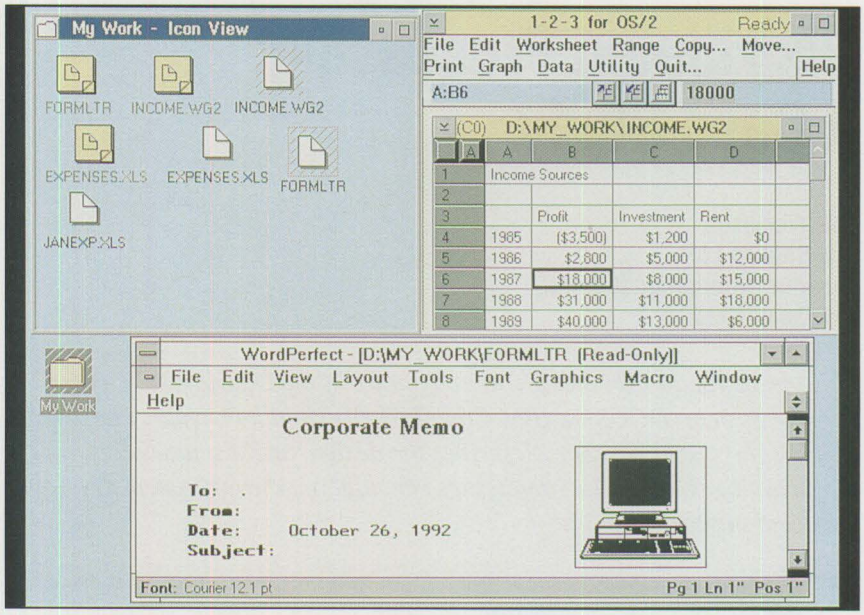


Figure 8. My Work Folder

Putting It All Together

Now use some features documented above to set up a folder that contains your work. To be sure that the folder is not set up on the default drive, open the Drives object first, then the Templates object. Drag the Folder template to the Drive D object and release mouse button 2. This action creates a new folder on drive D:. Open the Drive D object and rename the new Folder object to `My Work` using the procedure outlined above. Since you have just created this folder on drive D:, all objects created within this folder also are created on drive D:. Press and hold the Ctrl and Shift keys, then hold down mouse button 2 to drag a shadow of this folder to the Desktop. Finally, release mouse button 2, then the Ctrl and Shift keys. You now have the convenience of a handy work folder, as well as the ability to control where your data is stored (in this case, on drive D:).

Place shadows of any Program or Data-file objects that you normally use into this folder. Likewise, into your My Work folder, place a shadow of the `FORMLTR` template that you created above (refer to Figure 8). When you are ready to use `FORMLTR`, drag a template into the My Work folder and release the template. The data file actually is created on drive D: in the `D:\MY_WORK` directory. Double-clicking on the `FORMLTR` icon starts WordPerfect.

The Workplace Shell allows users to work the way they want to work. Using the procedures outlined here can enhance the end users' productivity by allowing the Workplace Shell to automatically handle routine functions.

Rob Talley *is a senior market support representative in IBM's Personal Systems Competency Center in Roanoke, Texas. Rob has supported OS/2 since 1987. His key responsibilities include the OS/2 Workplace Shell, and DOS and Microsoft Windows emulation. He received a bachelor's degree in business administration from the University of Texas at Arlington.*

Distributed Processing: A Case Study

Will Youngson
IBM Canada Ltd.
Canadian International Centre
Markham, Ontario

This article focuses on the distributed nature of the Distribution Center Control System (DCCS) at IBM Canada's Distribution Center. The article outlines the system's scope, describes the design, and discusses some of the advantages, obstacles, and issues encountered during system development and implementation.

The words *distributed processing* are very popular, yet have many interpretations. Generally, a *distributed system* is viewed as a number of distinct processing units performing tasks that combine to achieve a common objective or to handle a specific mission. A more precise definition depends on the proximity of the processors and the level of cooperation required between components in the distributed system.

Consider an inventory accounting system comprised of a minicomputer at Company XYZ's head office plus several independent, geographically dispersed minicomputers at XYZ's branch offices. The branch computers process local data and provide local functions. Every month, information is rolled up to the head-office computer for reporting and planning. The information transfer could be in the form of disk, tape, or file transfer over a common carrier. This scenario is often described as a *decentralized* system, but it may also be classed as a *loosely coupled* distributed system.

At the opposite extreme, a *tightly coupled* distributed system consists of two or more processes executing on separate computers and sharing common resources such as memory, direct-access storage, and external devices. These are often called *multiprocessor* systems.

Between these two extremes lies a distributed architecture called a *locally distributed Multiple Processor System* (MPS)[1]. Here, a processor is defined as having a dedicated central processing unit and memory, and is connected to other processors by a high-speed communication medium such as coaxial cable or fiber optics. Multiple-processor distributed systems are gaining in popularity because small, powerful, relatively inexpensive computers are becoming more available. In addition, operating system and software products now exist that assist with development and implementation of an MPS.

IBM Canadian Distribution Center

The multiple processor system was the architecture chosen for the Distribution Center Control System (DCCS) at IBM Canada's Distribution Center (CDC) in Markham, Ontario. It is the hub for the inventory and distribution of products sold by IBM Canada. The CDC, shown in Figure 1, is a highly automated facility with over 145,000 square feet of racking, three quarters of a mile of conveyors, Automated Retrieval Vehicles (ARVs), Automated Guided Vehicles (AGVs), Programmable Logic Controllers (PLCs), Radio-Frequency Terminals (RFs), and bar code equipment.

Several years ago, PC LAN-based systems were used to manage the movement and storage of products within the CDC. In recent years, however, the CDC underwent substantial physical expansion and modification to adapt to new fulfillment processes and increased product flow. Concurrent with this expansion, a more comprehensive system was required to manage the new functions, processes, and physical space. IBM developed a custom system to address this need – the DCCS, which was first released in July 1991.

DCCS System Flow

DCCS manages the movement and storage of products (from both IBM and other vendors) into, around, and out of the CDC.

Inbound Products

As product shipments arrive at the CDC, operations personnel use DCCS to receive each part or pallet, individually or in bulk. Information describing the product is scanned and a license plate (bar-coded label) is generated. DCCS then queries the inventory and racking database and assigns a best-fit location. Depending on which racking area is chosen for storage, the product is then either picked up manually by a forklift driver or automatically by an AGV. It is delivered to the inbound conveyors that are used as staging areas for putting away the product.

Operators who handle products use RF terminals mounted on forklift trucks to view a list of products to be put away into racking. They scan the

[1] Liebowitz, Burt H., *Multiple Processor Systems for Real-Time Applications*. Prentice-Hall, 1985.

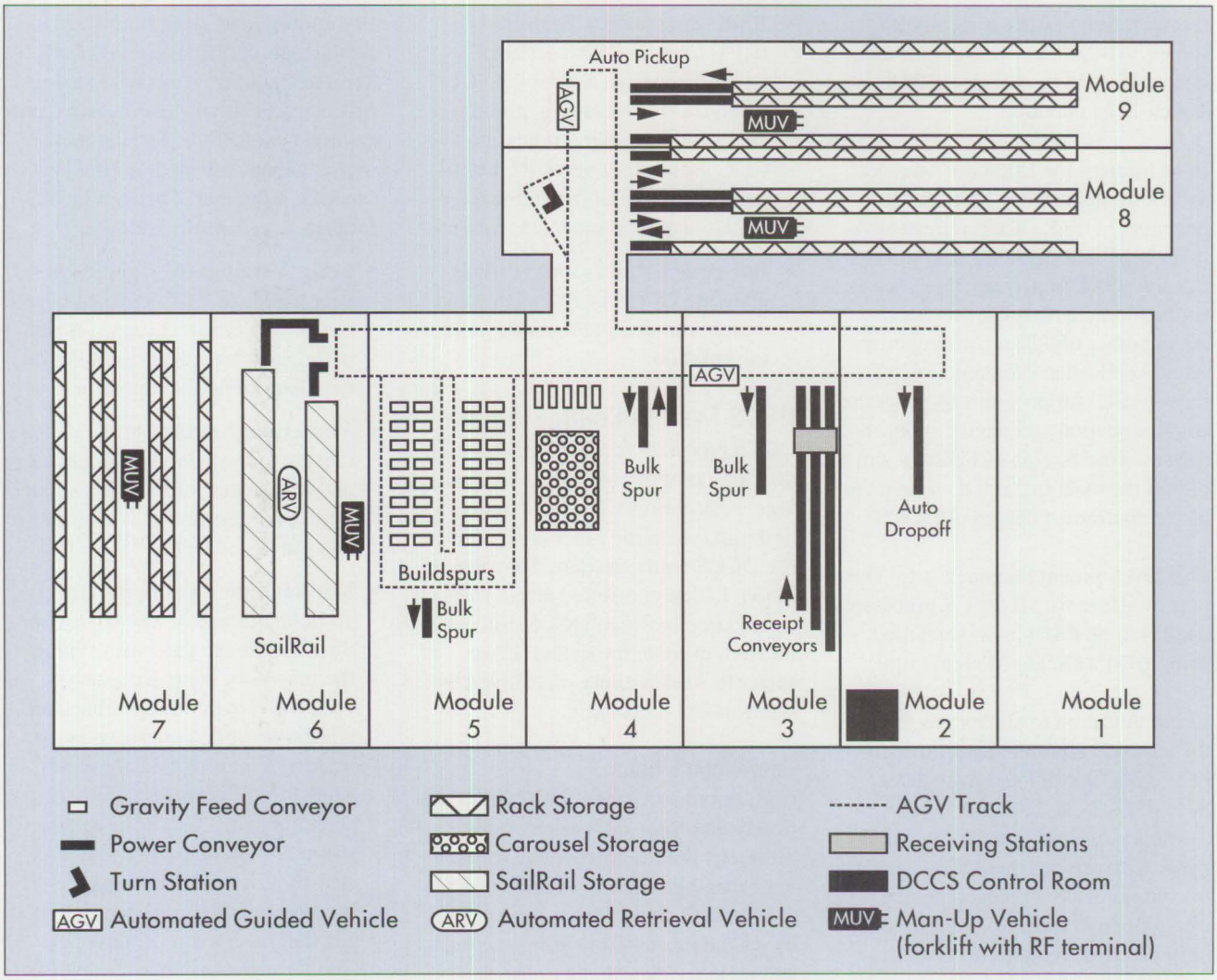


Figure 1. Canadian Distribution Center

DCCS license plate (generated during receipt) and place the product in the location (module) assigned by DCCS.

High-volume products are stored in a special racking system called SailRail™. The CDC's SailRail configuration has two opposing sets of racking. Each set contains about 184 lanes (46 lanes wide by 4 high), and each lane can hold up to 15 pallets of identical product. SailRail lanes are restocked with products from one end by CDC personnel operating forklifts, and are emptied (to fill orders) from the opposite end by an Automated Retrieval Vehicle.

Outbound

Customer orders come in from upstream host systems and are automatically fed into DCCS. DCCS locates the desired parts and generates retrieval requests. Depending on the location of the parts, the pulls may be performed automatically or with operator assistance.

Automated Retrieval Vehicles controlled by DCCS pull pallets from SailRail racking without human intervention. Most products, however, are stored in traditional racking. In this case, operators use RF terminals on forklift trucks to view a list of the retrievals generated. The operator pulls the product from the location indicated by DCCS and confirms the pull on the RF terminal.

In either case, pallets or parts pulled are placed on an outbound conveyor by the ARV or operator. DCCS automatically summons an AGV to transfer the product from the outbound conveyor to a conveyor in the appropriate destination module.

DCCS System Environment

DCCS hardware is based on PS/2 systems that range from 16 MHz Model 70s to 50 MHz Model 95s. All workstations are equipped with the maximum 16 MB of RAM. OS/2 1.3.2 is

the operating system currently installed. All DCCS workstations are interconnected by a 16 MB IBM Token-Ring network.

In addition to the Database Manager and Communications Manager components of OS/2, DCCS relies heavily on IBM's Distributed Automation Edition (DAE) software. DAE is an application enabler that provides, among other things, a comprehensive set of Application Programming Interfaces (APIs) for program-to-program communication and device-independent interfaces. This has greatly simplified the planning and development of the distributed design of DCCS.

The development language is C. The user interface for all DCCS processes displayed on PS/2s was developed using IBM's Dialog Manager tool.

Communication to all devices (except the bar code equipment) is through IBM's Multiport/2 and busmaster ARTIC (A Real-Time Interface Coprocessor) communication cards. Special drivers for the ARTIC cards to communicate with each device were obtained through a combination of in-house development, vendors, and IBM product offerings.

DCCS Devices

There are a variety of devices controlled by DCCS:

- **Automated Retrieval Vehicles (ARVs)** retrieve products from the SailRail racking system.
- **Automated Guided Vehicles (AGVs)** pick up and deliver pallets from one conveyor to another. AGV's follow a series of chemical trails connecting all possible pick-up and drop-off points.
- **Programmable Logic Controllers (PLCs)** monitor and filter state changes picked up by hundreds of optical sensors located on conveyors throughout the CDC.
- **Radio-Frequency Terminals (RFs)** interact with a series of DCCS processes available to CDC personnel for receiving, putting away, and retrieving products. The CDC uses two types of RF terminals: small, portable handheld terminals and truck-mounted terminals.
- **Bar code printers and readers** generate DCCS license plates and scan bar-coded product and pallet information.

DCCS System Configuration

DCCS processes are distributed among 40 PS/2 computers, 25 of which are user workstations. The remaining 15 nodes make up the processing core of the DCCS configuration. They reside in two badge-controlled areas. Communication between DCCS processes located on either the same PS/2 or separate workstations is achieved by interprocess messaging.

User Workstations

CDC personnel access DCCS for day-to-day activities such as receiving products, putting them away, generating orders, and summarizing work in progress. Each activity is represented by a single executable user process, and all activities can be accessed from any PS/2 workstation connected to DCCS. When a user logs onto DCCS, OS/2 groups are created (on the desktop) listing all processes that are accessible based on that operator's security classification. User processes execute locally, and they communicate with other processes on the floor or in the DCCS control room through DCCS messaging.

The DCCS Control Center

The DCCS processing core systems are installed in the DCCS control room. Figure 2 illustrates the physical layout of the hardware and the distribution of the individual processes.

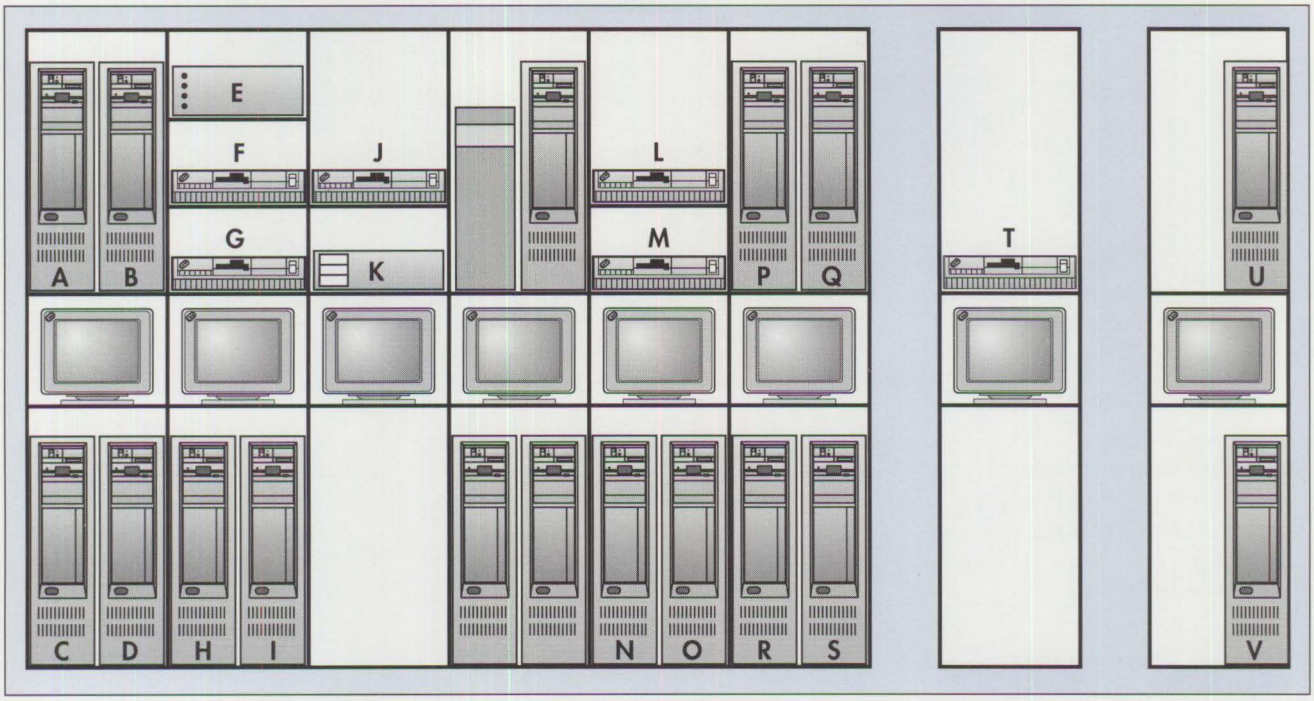
Four high-end processors house the databases required for security, inventory control, and audit trail. Each database server is configured for Remote Data Services (RDS) using Advanced Program-to-Program Communication (APPC). The database engine used in all cases is IBM's Database Manager. The four DCCS databases include the following:

- **Security database**. This controls user access to DCCS function, and contains user information and all parameters necessary to build the menus presented to operators.
- **Main Image database**. This contains inventory and location data, and also stores information about product orders and movement into and out of the CDC.
- **Mirror Image database**. This is an online image of key tables from the main image database. When the inventory control processes update the main image tables, an additional update message is forwarded to a mirror database administration process running on a separate node. This ensures that the mirror database is updated in real-time. The mirror image doubles as an operational reporting database and an emergency main image backup in the event of a non-recoverable database error.
- **Transaction database**. A history of all transactions applied to selected key tables is kept here. It is used primarily for resolution of inventory discrepancies.

Process Separation and Distribution

The distribution of DCCS processes among the remaining nodes in the control room was determined according to role and estimated resource loading. Descriptions of the functions executed by principal DCCS processes follow.

Inventory and work-in-transit tracking. Four processes reside on a client workstation remotely con-



A DCCS OS/2 LAN Server Domain Controller
B DAE Reference Tables
C Inventory and Work-in-Transit Update
D Transaction Database Update
E Radio-Frequency Base Station
F Radio-Frequency Message Router
G Security Database and Security Administrator
H DCCS Message Routing Coordinator Processes
I AGV Routing and Dispatch
J Inventory Database
K Litton ECS AGV Control System
L PLC Control
M PLC Control
N Transaction Database
O Radio-Frequency Screen-Handler Processes (12)
P DAE Tables
Q Mirror Update and Log Update
R Conveyor Control
S Automatic Build Spur Replenishment and Radio Frequency Screen-Handler Processes (12)
T ARV Mission Selection and Control
U DCCS Data/Configuration Backup—Additional Server
V Mirror Database

Figure 2. DCCS Control Room Configuration

nected to the main image database. Together, they manage all updates and queries to inventory and work-in-transit tables (such as queries about the list of putaways and about retrievals still outstanding). Requests for updates and information arrive in the form of DCCS messages that come from processes running on one of the other PS/2s in the control room, or from a user process running on a workstation on the warehouse floor.

Security administrator. This process controls system startup and shutdown, and verifies user logons against the Security database. Messages are generated when operators log on or log off DCCS from a workstation or an RF terminal. When a logon message is received, the security administrator verifies the logon, and then assembles all parameters necessary to build the DCCS menus on the user workstation. Next, for each item to be added to a menu, the security administrator sends a message back to the logon process on the user node.

AGV routing and dispatch. These processes determine the routing of automated guided vehicles for pickup and delivery of products from a source conveyor to a drop-off conveyor. They communicate through a Multiport/2 ARTIC card to a Litton® AGV External Control System™ (ECS) that runs on a PC AT.

PLC monitor. This process communicates with Allen-Bradley® Programmable Logic Controllers (PLCs) for the purpose of monitoring optical sensors located on all pickup and drop-off conveyors. Communication with the PLCs uses a busmaster ARTIC card and a communications protocol program written by IBM specifically for the Allen-Bradley PLCs. Because sensor readings are affected by the movement and placement of pallets on a conveyor, the PLC monitor detects the presence of a pallet and forwards an appropriate message to other DCCS processes, such as conveyor control.

Conveyor control. This process monitors and controls the movement

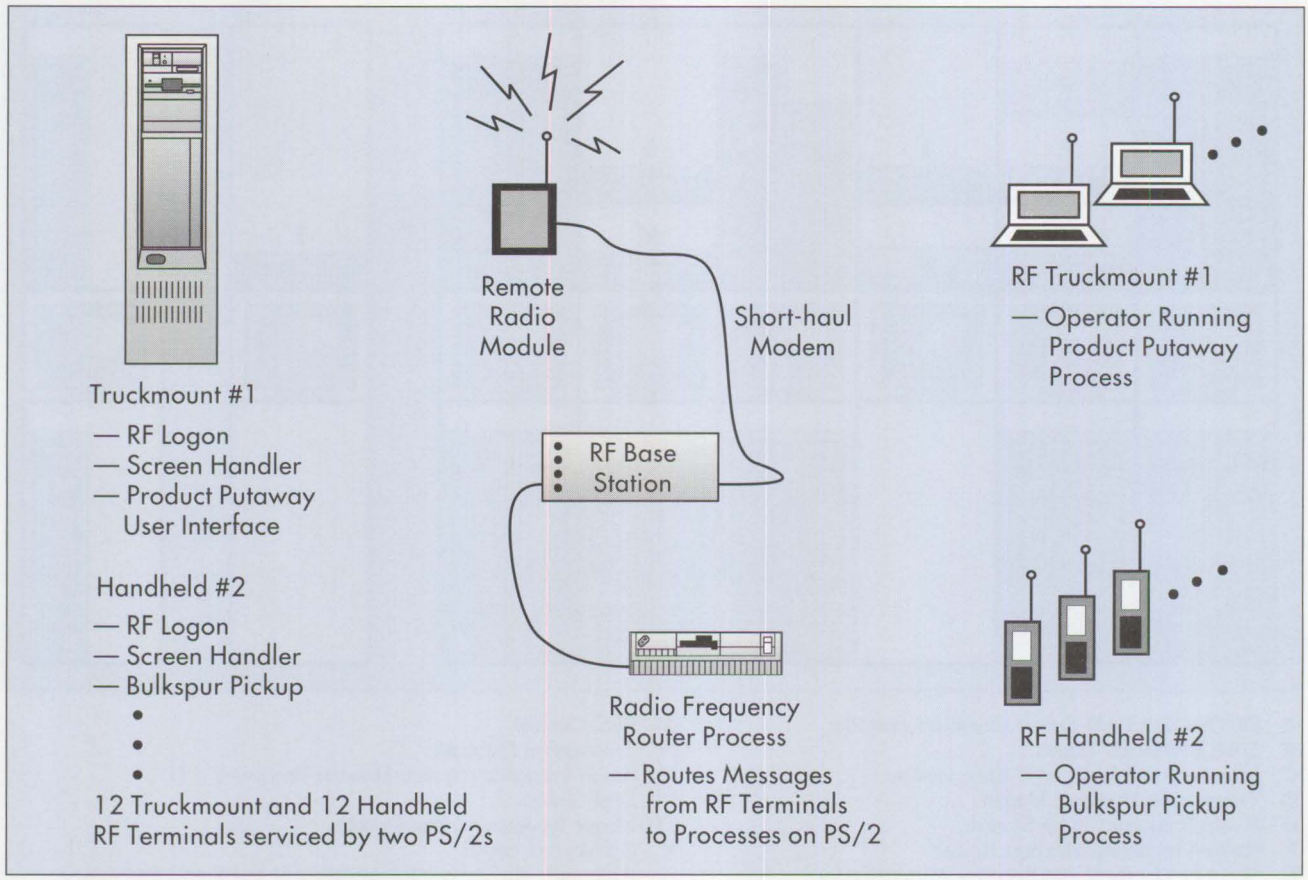


Figure 3. RF Message Routing

of pallets on key pickup and drop-off conveyors in the CDC.

Automatic build spur replenishment. This process controls the automated ordering of products for over 100 build spur conveyors.

RF message router. Many activities performed by CDC personnel require RF terminals. Most DCCS user processes are designed to be accessible from either a PS/2 or an RF terminal. The RF terminals used by DCCS are not intelligent – they can do only simple cursor movement and editing. Screens presented to the user must be downloaded from a PS/2. All data entered by the operator must be passed back to an intelligent workstation for processing.

When an operator turns on an RF terminal, an instance of a logon and screen-handler process assigned to that specific terminal is activated on a PS/2 in the DCCS control room. After logon, if the operator makes a selection from the list of available processes, an instance of the process corresponding to the selection is started on the same PS/2.

Multiple instances – one for each unique RF terminal – of the RF logon and screen-handler processes can run concurrently on a given node. Similarly, multiple instances of user processes, such as product putaway or retrieval confirmation, can run on a given node – one instance for each terminal that is currently executing the process. Figure 3 shows the correlation between RF terminals and user processes running on PS/2s in the DCCS control room.

The RF Message Router process communicates with the RF base station through a MultiPort/2 ARTIC card. It receives messages from all RF terminals and, based on the terminal number, forwards the message to the appropriate RF user process (or screen handler) on the appropriate node.

Currently, DCCS is configured for 24 RF terminals. To reduce CPU loading and to improve performance, the 24 corresponding RF logon, screen handler, and user processes have been distributed between two PS/2s.

ARV mission selection and ARV control. These processes communicate with the ARVs that pull products from SailRail racking. They

determine the next mission based on retrieval priority. Then they direct the ARV to the appropriate rack location and monitor the status of the pull until it has completed successfully.

Recovery log update. This process writes all critical table update transactions to a binary log file for later recovery in the event of a critical failure on the Transaction database, or failures on both the Main and Mirror Image databases.

Mirror image update. This process updates the Mirror Image database.

Transaction DB update. This process adds a new record to the Transaction database for each update performed on key tables in the Main Image database.

Interprocess Communication

DCCS interprocess communication relies heavily on the peer-to-peer communication facility built into the Communication System/2 (CS/2) component of DAE. CS/2 includes a comprehensive set of APIs for sending and receiving messages between resources on one or more PS/2s. CS/2 Application Control Blocks (ACBs) and Message Control Blocks (MCBs) are used extensively by DCCS processes. At least one ACB is required for each process that wants to use CS/2 services. This ACB identifies the owning process, and connects that process to other processes and resources on the CS/2 network. MCBs are message queues that can be accessed by processes connected to the DAE network. In the DCCS implementation, each DCCS process monitors at least one message control block. All communication between DCCS processes is performed by exchanging messages between MCBs.

A common DCCS message structure, shown in Figure 4, was defined for all interprocess messaging.

```
typedef struct
  {
  UCHAR  szReplyTo[ CS2RSNAMSIZ_N ];
  USHORT usMsgType;
  USHORT usMsgCode;
  USHORT usBufSize;
  UCHAR  pchBuffer[1];
  } typDCMSG;
```

Variable	Description
szReplyto	Contains the resource name (MCB) to which the responses to the message should be directed
usMsgType	Identifies the type of structure defined in a common DCCS include file passed in the buffer of the message
usMsgCode	Specifies a particular action that must be performed on or with the message
usBufSize	Holds the length of a variable-length message buffer
pchBuffer	Used to attach buffers to the header, as shown in Figure 5

Figure 4. DCCS Interprocess Message Structure

```
UCHAR pchMsgSpace[ sizeof(typDCMSG) + sizeof(typYOURMSG) - 1];
typDCMSG   *pdcm = (typDCMSG *) pchMsgSpace;
typYOURMSG *pyourmsg = (typYOURMSG *) pdcm->pchBuffer;
                       .
                       .
                       .
pyourmsg->field1 = value1;
pyourmsg->field2 = value2;
```

Figure 5. Attaching Unique Message Structures to DCCS Message Header

Why Distributed Processing was Chosen

A distributed processing architecture was chosen for the DCCS because it satisfied several requirements:

- **Ability to implement tasks as discrete modules**. CDC operations fall into distinct categories. Distributed processing is a good fit for this situation because it allows design and development to approach each area separately. Once the task of laying out the interprocess messaging and control strategy was complete, team members could attack the individual processes responsible for specific tasks, such as AGV routing, conveyor control, and order generation.
- **A flexible, reliable, responsive system that does not slow down or shut down the CDC in the event of a system problem**. Processor and disk failures can be isolated either by temporarily disabling the function (for example, ARV retrievals) affected by the malfunction, or by moving the affected programs to another processor in the system. The distributed nature of the system means that, in most instances, only a small subset of overall system functions may be unavailable if the hardware fails. In addition, dis-

tributing processes over several PS/2s has resulted in reducing processor and storage usage and delivering acceptable performance. As improved computer systems become available, key processors can be upgraded easily.

- **Ability to add modules and function to the system as required.** Since its original installation in June 1991, DCCS has had three new releases. This evolution was necessary from an operational perspective and desirable from a systems perspective. As new functionality is required, nodes and processes can be added with minimal impact to the existing configuration, and little or no impact on daily operations.
- **Need to interface with a variety of automated devices.** A critical component of DCCS is the monitoring and control of a variety of sophisticated automation equipment. Placing the device communication and control processes onto separate PS/2s simplified the design and configuration issues and reduced the constraints inherent with a mix of unique hardware, software, and protocols.

Nothing is Free

The installation of DCCS at the Canadian Distribution Center was a success. Even so, some problems and obstacles were encountered.

Design complexity. The overall complexity of the DCCS system design proved to be proportional to the degree of distribution of processes. The more finely we isolated processes to perform specific tasks, the more complicated became our system control mechanism and messaging strategy. Other coordinating processes had to be introduced to control process start-up and shutdown and message routing. Early in the design phase, it became obvious that a well-defined set of message types and structures must be defined. Early development activities focused on creating a common library of messaging routines to be used by all developers. This helped ensure that processes would be able to communicate with each other when unit testing was complete and integration testing began.

System maintenance. In our case, implementing a locally distributed multiple processor system has translated into more planning, preparation, and work for DCCS system configuration and support staff. Instead of maintaining one or two computers, our staff has to maintain about forty. Installation of new releases of DCCS, OS/2, and DAE are time-consuming tasks. We are currently testing the improved Remote Initial Program Load (RIPL) feature of OS/2 LAN Server 2.0, because we want to RIPL user stations from one or more servers. This would simplify the upgrade process substantially.

Environment configuration. Network configuration and tuning had to satisfy the requirements of OS/2 LAN Server, DAE communications (using NetBIOS), and remote database services (using APPC). Documentation and installation tuning recommendations for each component operating alone are adequate, but when taken together, some adjustments were required.

Error identification and recovery. The advantage of having multiple processors is that if a processing or hardware error occurs on a single node, the DCCS system can provide most system functions. The flip side, however, is that error identification and compensation are more complicated in a distributed system because the source of the error is also distributed. Errors experienced on any node must be reported immediately to a controlling process so that proper action – such as calling for support or shutting down – can be taken.

The incident-reporting component of DCCS handles errors in several ways, depending on the type and severity of the error. All errors encountered by DCCS processes are forwarded to an error-logging process that runs on the local node; errors are displayed on the screen and logged to a file. In addition to local logging, informational and error messages are routed to a central DCCS Analyzer process that interprets the nature of each message and takes appropriate action. If the error is severe, a DCCS support person is paged with an alphanumeric pager, and the pager's display shows the process name and error code. If the error is critical and cannot be corrected by the system, the system is suspended automatically – users are logged off and DCCS processes are held in a frozen state. Intervention by DCCS support personnel is required at this point to correct the error and resume system operations.

A Heartbeat process executing on a separate node monitors the state of critical DCCS processes running on the 15 PS/2s in the DCCS control room. Heartbeat sends messages to each critical process at regular intervals. Targeted processes echo back the Heartbeat message with a time-stamp and sequence number. The Heartbeat process analyzes incoming Heartbeat-echo messages to assess the health of the system. If a PS/2 or process abends, Heartbeat sends a message to the DCCS Analyzer process, instructing it to page DCCS support.

Will Youngson *joined IBM in 1990 as a systems engineer and is currently a technical analyst at the Canadian International Centre for the DCCS project. He holds a BSc in system design engineering from the University of Waterloo in Ontario.*

Parallel Port Protocols

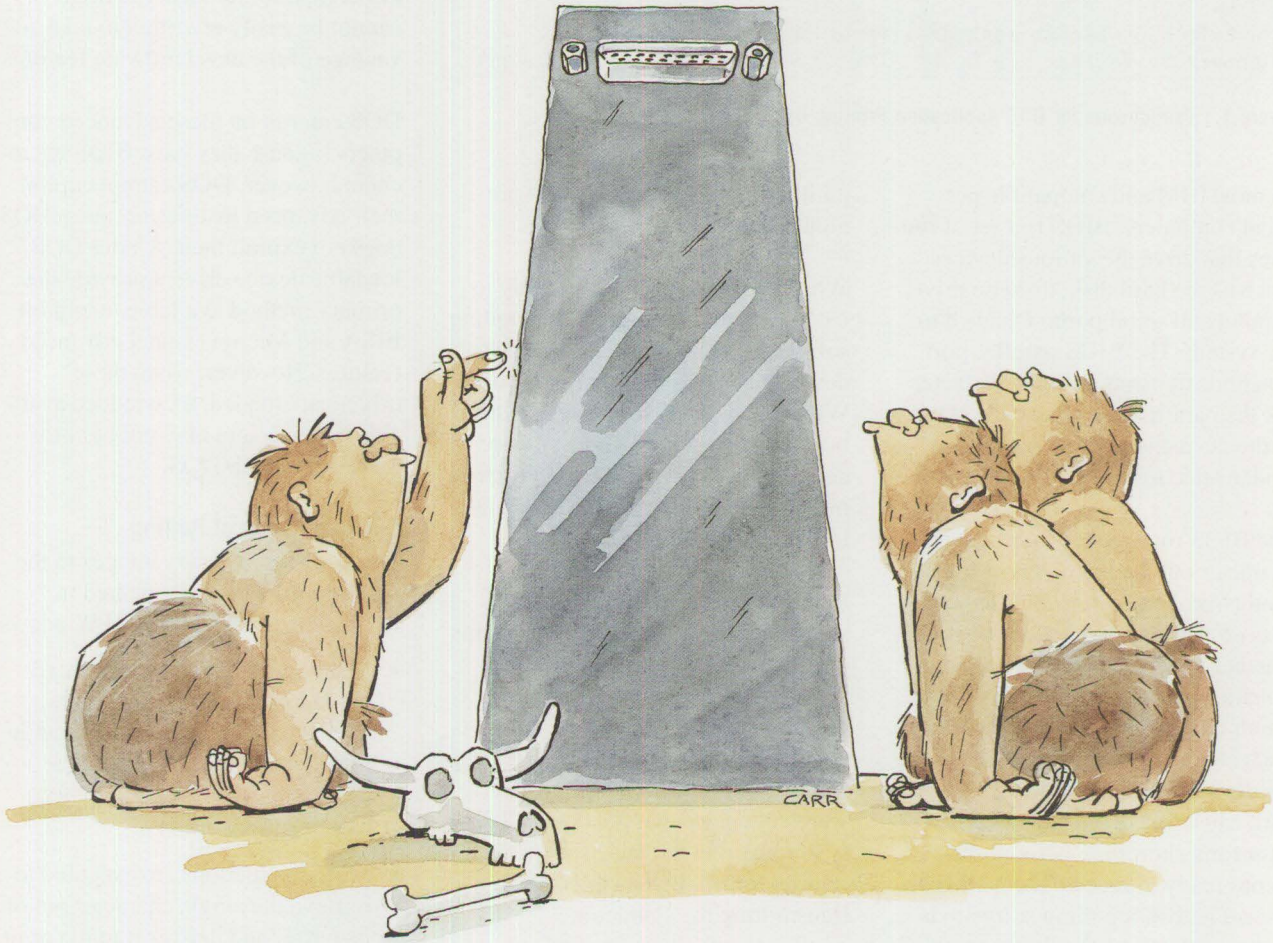
Frank J. Schroeder
IBM Corporation
Boca Raton, Florida

This article discusses the different software methods used to transfer data through a parallel port. Operating systems use these different parallel port protocols so that they can exploit the advanced features of the parallel port hardware. The article describes the protocols used by DOS and OS/2, gives examples, explains the advantages and disadvantages of each protocol, and explains the relationships between the protocols, operating systems, and bus architectures. Certain Micro Channel-based IBM PS/2 systems running OS/2 provide a clear advantage over the AT and EISA bus architectures.

Three parallel port protocols are available in the unidirectional transfer of data through the parallel port: polling, character hardware interrupt-driven (also known as Programmed I/O, or PIO), and block hardware interrupt-driven (also known as Direct Memory Access, or DMA). Each protocol requires hardware with specific capabilities to transfer data through a parallel port.

Parallel Port Transfers Using Polling

DOS is a single-tasking operating system that uses the Basic Input/Output System (BIOS) microcode, which resides in Read-Only Memory (ROM)



DOS Application

```
set abort flag to false
determine buffer address, count, device
repeat
  if buffer is not empty
    get character out of buffer
    call BIOS to print (character, device)
    if error
      set abort flag to true
    endif
  endif
until buffer is empty or abort flag is true
```

BIOS

```
set character-sent flag to false
set timeout counter to large value
repeat
  read parallel port status
  if not busy
    send character to output device
    set character-sent flag to true
  else
    decrement timeout counter
  endif
until character is sent or timeout counter = 0
return to caller
```

Figure 1. Pseudocode for DOS Application Printing Using BIOS

on most IBM and compatible personal computers. BIOS is a set of routines that drive the various devices (such as the hard disk, diskette drive, parallel and serial ports) attached to the system. The BIOS parallel-port functions transmit a single character out the parallel port, check the status of the device, and initialize the device attached to the parallel port.

The BIOS routine that transmits a character out the parallel port uses a protocol called polling. *Polling* consists of instructions to determine whether the device is ready to receive a character and, when ready, to send the character out the parallel port. Each character, as it is received by the device attached to the parallel port, causes the device to become busy for a short time before it becomes ready to receive the next character. The BIOS polling software is necessary because the device itself cannot receive the next character until it first stores or prints the previous character.

While the device is not yet ready to receive the next character, the polling software continuously checks the device's status for a ready condition. When the device status changes from busy to ready, the polling software detects this change of state and transmits the next character out the parallel port to the device.

Polling the device for the ready status continues until either the status changes to ready and the character can be transmitted, or a time-out occurs. If a time-out occurs, it is because the device status has been busy for a period of time. The BIOS software concludes that the device is likely to remain in the busy state, so it returns control to the caller without transmitting the character.

The Microsoft Windows operating environment, built to run on top of the DOS operating system, uses the same polling method of printing that DOS uses.

Figure 1 contains a simple pseudocode algorithm similar to the one implemented in BIOS and used by DOS, Windows, and other DOS applications to transmit a character out the parallel port.

Advanced Hardware Features

The polling protocol implemented in BIOS does not use the advanced features – hardware interrupts and interrupt sharing – found in Extended Industry Standard Architecture- (EISA-) and Micro Channel-based systems. BIOS was originally written for the AT bus (also known as Industry Standard Architecture) design. For compatibility reasons, BIOS cannot be easily rewritten to take advantage of the new hardware features.

DOS can run on these advanced computers because they have BIOS microcode; however, DOS cannot exploit their advanced features because BIOS does not exploit them. Under DOS, loadable device drivers provide the primary method available to exploit EISA and Micro Channel advanced features. However, when these drivers are loaded, they reduce available memory and may not be fully compatible with DOS.

Disadvantages of Polling

There are three disadvantages to the polling protocol implemented in BIOS, and used by DOS and Windows.

- Many more instructions are executed in a loop waiting for the device to become not busy than in transmitting the character. The processor could be doing other more useful work.
- More instructions are executed to retrieve the single character out of its buffer and deliver it to the print routine than are executed in transmitting the character. The ideal

approach is to have the instructions execute only once for the entire print buffer, rather than once for each character transmitted.

- While printing under DOS and Windows, no other tasks can execute, primarily due to DOS's single-tasking nature and the polling method used by its physical transport layer, BIOS. Therefore, a computer's throughput under DOS and Windows is far from optimal. When more activities can execute simultaneously, the computer has greater throughput, which helps increase productivity.

Parallel Port Transfers Using Hardware Interrupts

The OS/2 operating system uses a more efficient method of printing that exploits the advances in EISA and Micro Channel architecture.

Because OS/2 is a multitasking operating system, several applications or tasks can be loaded into memory at one time. OS/2 schedules each task serially with the processor. Each task is assigned to the processor for a short period of time, and then the next task is assigned to the processor. From a user's perspective, each task appears to be running concurrently since each task is making progress, but in reality only one is executing at a time. OS/2 attempts to overlap Input/Output (I/O) with application execution to produce true concurrency – the ability to perform more than one operation at the same time.

OS/2 takes advantage of advanced Micro Channel features, either by having the OS/2 device drivers use Advanced BIOS (ABIOS) routines to access these features, or by having OS/2 device drivers program the hardware directly. The OS/2 device driver is responsible for insulating the application from the specifics of the hardware and for physically transporting data. Support for the ABIOS parallel port includes the following:

- Transmit data out the parallel port
- Return the status of the device attached to the parallel port
- Initialize the device attached to the parallel port
- Cancel the print request
- Return information about ABIOS parallel port support on the particular computer

With a multitasking operating system like OS/2, it is not necessary to wait for a DOS application to finish printing before continuing work with OS/2. The ABIOS routine used by OS/2 to transmit data out the parallel port is more efficient than the polling protocol used by BIOS. That is because the hardware interrupt protocol used by ABIOS permits other tasks to run while the device attached to the parallel port is busy receiving the data being transmitted.

Protocols Implemented in OS/2

The ABIOS routine that transmits data out the parallel port can use either the Programmed I/O (PIO) or the Direct Memory Access (DMA) parallel port protocol. PIO is available on all IBM and most compatible personal computers with AT-bus, EISA, or Micro Channel bus configurations. DMA is available on most PS/2 Micro Channel systems.

Both the PIO and DMA protocols use the hardware interrupt capability of the parallel port. Generally, the hardware interrupt mechanism works as follows: a character of data is transmitted out the parallel port, and the application's thread of execution is blocked, waiting for the transmission to complete. The device attached to the parallel port receives the character and generates the hardware interrupt, which signifies that the device driver can begin transmitting the next character. When the device driver is waiting for the device to generate an interrupt, OS/2 runs other tasks. In contrast, the polling protocol requires the software to loop. This means waiting for the device to return to being ready to receive, and preventing DOS and Windows from running other tasks.

Figure 2 shows a pseudocode algorithm that describes the PIO and DMA hardware interrupt protocols used by OS/2 to transmit characters out the parallel port. It illustrates how PIO and DMA parallel port transmissions work. The OS/2 application sends a buffer of data to the device-driver strategy entry point (entry point 1) to be sent out the parallel port. The device-driver print routine calls ABIOS to transmit the first character using the hardware interrupt protocol. It then starts a timer that can expire if a hardware interrupt is not generated. If the timer expires, the device driver will be called at entry point 3, and the print request will be canceled. This occurs only if the device does not receive the character transmitted and if it does not generate the hardware interrupt. The application's thread of execution is blocked, waiting either for all the data in the buffer to be transmitted or for an error to occur. While the thread is waiting for the transmission to complete, other tasks are scheduled to run. When the hardware interrupt occurs, the other running tasks are preempted so that the interrupt can be serviced. Once it is serviced, the scheduler dispatches the previously running task, which runs while the parallel port device is busy. It is this overlapping of I/O with application execution that produces concurrency and improves system throughput.

Advanced Hardware Features

Hardware interrupt sharing is an advanced hardware and software mechanism that enables multiple devices to be driven by the same hard-

OS/2 Application

```
determine buffer address, count, device
call OS/2 to print (entry point 1)
```

OS/2 Entry Point 1 (strategy routine)

```
register interrupt routine with interrupt manager (entry
point 2)
send first character out the parallel port (call ABIOS)
start time-out timer (entry point 3)
block thread waiting for transmission completion
return to application with return code and count transmitted
```

OS/2 Entry Point 2 (hardware interrupt routine)

```
if parallel port generated interrupt (call ABIOS)
  turn time-out timer off
  if running in PIO mode
    if more characters to send
      turn time-out timer on
      send next character (call ABIOS)
    else
      set count to number of characters transmitted
      set return code to no error
      run blocked thread (entry point 1)
    endif
  else running in DMA mode
    set count to number of characters transmitted
    set return code to error / no error
    run blocked thread (entry point 1)
  endif
  clear hardware interrupt at 8259 controller
  return to interrupt manager as owner of interrupt
else
  return to interrupt manager as not owner of interrupt
endif
```

OS/2 Entry Point 3 (timeout routine)

```
turn time-out timer off
cancel print request (call ABIOS)
set count to number of characters transmitted
set return code to error
run blocked thread (entry point 1)
return to timer services
```

Figure 2. Pseudocode for OS/2 Application Printing Using ABIOS

ware interrupt level. This advancement was added to EISA- and Micro Channel-based systems after computer manufacturers realized that the number of hardware interrupt levels was a limitation in AT bus (ISA) systems. AT bus systems allow only one device to be driven by a hardware interrupt level. OS/2 exploits hardware interrupt sharing by extending this limited resource on computers that support hardware interrupt sharing.

The OS/2 parallel-port device driver supports hardware interrupt sharing on EISA and Micro-Channel buses. When a specific hardware interrupt occurs, the OS/2 interrupt manager calls each device driver that has registered to be notified about a specific hardware interrupt. Device drivers are called in the order in which the hardware interrupt routines are registered. Because the OS/2 parallel-port device driver is a base device driver – a driver that is required for the system to boot – it is called to initialize itself before any loadable device drivers. Therefore, it registers its hardware interrupt routine prior to the interrupt routines of any loadable device drivers. Generally, a device driver registers its hardware interrupt handler within its initialization routine.

The interrupt manager calls the parallel-port device driver to service an interrupt that is generated on the level that the parallel port uses, even though an interrupt may belong to a device that is attached to a different port. The interrupt manager does not know which device owns the interrupt; it only knows which device driver has requested to be notified when a specific interrupt occurs. The parallel-port device driver's hardware interrupt service routine determines that the interrupt was generated by a device that shares the same hardware interrupt level as the parallel port device. This routine returns control to the interrupt manager, specifying that the parallel port device did not generate the interrupt. The interrupt manager calls each device driver in its list until either a device driver claims ownership of the hardware interrupt or the list is exhausted. If no device driver claims the interrupt, the interrupt manager considers the interrupt to be spurious, and will disable the hardware interrupt level.

PIO Versus DMA

A major difference between PIO and DMA is the frequency of interrupts. When ABIOS uses the PIO protocol, the interrupt handler (entry point 2) is called for every character transmitted. If ABIOS uses the DMA protocol, the interrupt handler is called when the entire buffer is transmitted, or when an error occurs at the device (out of paper, offline, or power off).

When PIO is the transmission method and the device has received the character sent, the device then

generates a hardware interrupt. The interrupt acknowledges to the device driver that the previous character was received successfully. It then begins the process of transmitting the next character (entry point 2). As the device driver finishes processing the interrupt, it must end the interrupt processing to the parallel port and to the programmable interrupt controller, claim the hardware interrupt, and return to the interrupt manager. The programmable interrupt controller is responsible for prioritizing hardware interrupt levels, recognizing when a hardware interrupt level is requesting service, disabling and enabling hardware interrupt levels, and alerting the processor about the pending hardware interrupt. In PIO, sending characters and processing hardware interrupts continues, character by character, until all the characters are transmitted and all the hardware interrupts are processed.

When the transmission method is DMA, the DMA controller is programmed to move the characters from the application's buffer in RAM, down the Micro Channel bus to the device attached to the parallel port. The DMA controller is a specialized processor, similar to a busmaster. Its sole purpose is to offload work from the processor by moving data from one location to another (generally from memory to port or vice versa). The DMA controller handles the entire transmission and generates a hardware interrupt that is serviced by the OS/2 device driver when the transmission completes or when an error occurs. The parallel port DMA enables I/O to happen concurrently with other tasks that are executing. It also minimizes processor involvement to the return of completed print requests and the delivery of additional requests to the device driver.

Advantages of Hardware Interrupts

There are several important points about the hardware interrupt protocol and how OS/2 implements this protocol.

- OS/2 enables multiple tasks to run simultaneously, because it does not waste the limited processor cycles on polling a device. Instead, OS/2 uses hardware interrupts to transfer data. During device-busy times, OS/2 can devote the additional processor cycles that would have been required for polling to the execution of other tasks.
- OS/2 encourages the use of buffering; it is designed to pass buffers of data rather than a single byte of data where possible and while maintaining compatibility with existing Application Programming Interfaces (APIs). Improved performance is achieved by spending more time actually transmitting data rather than passing the data to the device driver to be transmitted.
- The OS/2 parallel-port device driver currently uses ABIOS on Micro Channel systems, and automatically switches from PIO to DMA when executing on PS/2 systems that support DMA. The concurrency achieved when using DMA significantly improves system throughput on PS/2 Micro Channel systems compared to other AT and EISA bus systems. Generating one interrupt per buffer rather than one interrupt per character significantly reduces the number of interrupt-time instructions that the processor must process. This, in turn, allows the saved cycles to be allocated to other OS/2 tasks.
- If the parallel-port device driver is called to service a hardware interrupt that was not generated by the device attached to the parallel port, then the parallel-port device driver does not claim the interrupt. On computers that support hardware interrupt sharing, the interrupt manager calls other device drivers that request notification about the hardware interrupt. OS/2 exploits hardware interrupt sharing. Hardware interrupt sharing extends the number of hardware interrupt levels available, thereby helping to alleviate this limited system resource.

Conclusion

Passing blocks of data as parameters instead of a single character at a time, scheduling the processor with other tasks instead of making it wait for slower devices, and offloading output to specialized processors are a few of the many advantages of multitasking and true hardware concurrency.

The OS/2 parallel-port device driver exploits advanced parallel port features, including hardware interrupt usage, hardware interrupt sharing, and DMA. The DOS and Windows parallel-port device drivers do not take advantage of any of these advanced features and, therefore, underutilize advanced computer systems. OS/2 optimizes the personal computer's capabilities and, therefore, your productivity.

Frank J. Schroeder *is a staff programmer in the OS/2 Device Drivers and Multiple Virtual DOS Machines (MVDM) department in IBM Boca Raton. He has worked on several OS/2 components since Version 1.0, and is currently working on performance and functional enhancements to the OS/2 printer subsystem. Frank has previously published articles in* IBM OS/2 Developer *and* IBM Personal Systems Technical Solutions. *He earned a Bachelor of Technology degree in computer science from Rochester Institute of Technology, and an MBA from the University of Miami.*

Developing OS/2 Presentation Manager Applications with Micro Focus COBOL

Christopher Fierros
Dallas, Texas

This article introduces COBOL application developers to OS/2's Graphical User Interface (GUI), the Presentation Manager (PM). Issues specific to the COBOL programming language are discussed. The article shows how to code the most fundamental PM program using common PM Application Programming Interface (API) calls.

OS/2 quickly is becoming the platform of choice for downsizing host applications and developing mission-critical business applications. OS/2's Graphical User Interface (GUI), Presentation Manager (PM), is one of the most powerful GUI development environments on personal computers today. The complexity of programming for PM can be dramatically reduced by developing applications with COBOL.

Advantages of Using COBOL

Using COBOL to develop OS/2 PM applications has several advantages over other programming languages:

- COBOL is easier to learn than most programming languages. It uses a verbose, English-like coding syntax that is easy to read, follow, and understand. COBOL is a structured programming language. Because it is structured, it is easier and less costly to maintain than applications written in non-structured languages.
- COBOL is a popular programming language. Corporations already have thousands of mainframe applications written in COBOL, and many of these were developed on personal computers. Using existing programming skills can reduce application development costs considerably.
- COBOL is much better suited for applications that access data from relational database engines. COBOL is positioned well for the evolving client/server and distributed database architectures.
- Existing COBOL applications can be ported easily to the OS/2 PM environment without having to be entirely rewritten in another language. The application's data processing functions can remain intact, with only minor changes.

These advantages can reduce the cost of developing and maintaining OS/2 PM applications. The ability to extend an existing development environment to include OS/2 PM can reduce costs associated with retraining developers and can protect your existing software investments.

Development Software

The software required to develop COBOL applications for OS/2 PM (henceforth called COBOL/PM applications) includes OS/2, OS/2 Toolkit, and Micro Focus COBOL® Toolset. The Micro Focus 16-bit COBOL compiler has supported PM application development since Version 2.4 of the compiler was released in late 1989. Their 32-bit OS/2 COBOL developer's kit is scheduled for general availability in early 1993.

Figure 1 shows different operating environments available for COBOL/PM development using Micro Focus COBOL Toolset 2.5 or 3.0. Regardless of the operating system and development tool used, be sure to apply the latest OS/2 Corrective Service Diskette (CSD) and Micro Focus update levels.

The OS/2 2.0 Developer's Toolkit includes a greatly improved Dialog Box Editor and Icon Editor, and utilities that were not available previously, such as the `NMAKE` utility.

COBOL/PM Applications

This article presents a simple, straightforward process for developing COBOL/PM applications. The development process begins by creating various files needed to generate a PM executable program. After developing your first program, simply copy and rename the files you need. The COBOL/PM program developed

Operating System	Toolkit
OS/2 1.3	OS/2 Programming Tools and Information Version 1.3
OS/2 2.0	OS/2 Programming Tools and Information Version 2.0
OS/2 2.0	OS/2 2.0 Developer's Toolkit

Figure 1. Software Development Environments

in this article is called PMABOUT. It contains basic components of any COBOL/PM application.

Creating a Build File

The first step in developing a PM application involves creating a *build* file. The build file does exactly what its name implies – it builds source code into an executable program. There are two different implementations for managing the build process.

For the first implementation, simply create a REXX batch file, BUILD.CMD, that passes commands directly to the OS/2 command prompt. This file checks for nonzero return codes from each command submitted to the command interpreter and stops the build process if a nonzero return code is received. (The COBOL compiler returns codes of 0, 4, 8, and 16. A zero return code indicates successful compilation with no messages. A code between 4 and 16 indicates the severity of the messages received during compilation.) The BUILD.CMD file must have the extension .CMD because this is a requirement for REXX files. A REXX build file, BUILD.CMD, is shown in Figure 2.

The NMAKE Utility

The second implementation for managing the build process is a bit more complex. It uses a utility provided with OS/2 2.0 Developer's Toolkit called NMAKE. This utility can reduce development time considerably. The NMAKE utility reads an ASCII text file called a *make* file. The make file specifies the name of a *target* file and the names of *dependent* files on which the NMAKE utility

```
/* REXX BUILD.CMD for PMABOUT */
BUILD:
  'COBOL PMABOUT.CBL; confirm ans85 ganim'
    IF RC \=0 THEN SIGNAL ERROR
  'LINK PMABOUT.OBJ,PMABOUT.EXE,,lcobol+os2286,PMABOUT.DEF;'
    IF RC \=0 THEN SIGNAL ERROR
  'RC -r PMABOUT.RC'
  'RC PMABOUT.RES PMABOUT.EXE'
    IF RC \=0 THEN SIGNAL ERROR
  SIGNAL ENDPROG
ERROR:
  SAY '*** Build ended with error!'
  EXIT
ENDPROG:
  SAY '*** Build ended successfully!'
  EXIT
```

Figure 2. REXX Build File BUILD.CMD

```
All : PMABOUT.EXE

PMABOUT.RES : PMABOUT.RC PMABOUT.H PMABOUT.DLG PMABOUT.ICO
  RC -R PMABOUT.RC

PMABOUT.OBJ : PMABOUT.CBL PMABOUT.CPY
  COBOL PMABOUT.CBL; confirm ans85 ganim

PMABOUT.EXE : PMABOUT.RES PMABOUT.DEF PMABOUT.OBJ
  LINK PMABOUT.OBJ,PMABOUT.EXE,,lcobol+os2286,PMABOUT.DEF;
  RC PMABOUT.RES PMABOUT.EXE
```

Figure 3. NMAKE File PMABOUT.MAK

```
*-----------------------------------------------*
CONFIGURATION SECTION
*-----------------------------------------------*
  Source-Computer.   IBM-PS2.
  Object-Computer.   IBM-PS2.
  Special-Names.     Call-Convention 3 is OS2API.
```

Figure 4. Call Convention

performs a time-stamp comparison. Next, the make file specifies statements to be executed if the time-stamp comparison between the target file and dependent files fails. If the comparison fails – meaning that the time-stamp of the target file is older than the time-stamp of at least one dependent file – then the specified statements are executed.

For example, if you make source code (`.CBL`) changes and save those changes to disk, then the next time you run `NMAKE`, only the COBOL source code is recompiled. After recompiling the `.CBL` source code, the COBOL object module has a more recent time-stamp than the old `.EXE` file. The COBOL object module is relinked, producing a new executable (`.EXE`) module. Throughout this process, the original resource file (`.RES`) was not recompiled. Using the `NMAKE` utility can reduce development time substantially because the files that have not changed are not recompiled.

Additional information about `NMAKE` and other OS/2 2.0 Toolkit utilities is contained in the publication *OS/2 2.0 Developer's Toolkit: Getting Started*, which is provided with the OS/2 2.0 Toolkit.

By default, the make file has no extension; however, it is common practice to give it the same file name as the source code, and the extension `.MAK`. Figure 3 shows an example of the `PMABOUT.MAK` file. In Figure 3, note that the `GANIM` compiler directive creates support files for the Xilerator – the Micro Focus COBOL debugger provided with the Micro Focus COBOL Toolset. To invoke the debugger, type the following on the OS/2 command line:

```
xil /c /p pmabout
```

Source Code

The second step in developing a COBOL/PM application is to create the actual source code, copy book, and module definition file.

The COBOL source code file has an extension of `.CBL` unless it contains SQL code, in which case its extension is `.SQB`. The source code is like any other COBOL program written for OS/2, with a few minor exceptions.

The Call-Convention statement is the first indication that a program is using OS/2 Application Programming Interface (API) routines. This statement is coded under `SPECIAL-NAMES` in the `CONFIGURATION SECTION` of the `ENVIRONMENT DIVISION`. The call convention is defined as 3, which is the Pascal calling convention (see Figure 4). The Pascal calling convention is simply the order in which OS/2 expects to receive the parameters passed from the COBOL program to the API routine. The literal `OS2API` can be any valid COBOL

literal; however, OS2API is the COBOL standard.

Using WORKING-STORAGE, LOCAL-STORAGE, and LINKAGE SECTIONS controls the definition of variables and constants. The WORKING-STORAGE SECTION contains global variable declarations. In the PMABOUT example, all WORKING-STORAGE variables have been placed in a copy book. The copy book file is simply a file (with an extension of .CPY) in which object handles, controls, messages, structures, and other variables are defined. This file is copied into the COBOL source when the program is compiled. Placing all WORKING-STORAGE definitions into a single copy book is a personal preference only; they also could have been placed in various copy book files or the actual source code. Using copy books simplifies development and enables developers who are working on the same application to use standardized variables and constant names. The COPY statement copies the entire copy book into the source code during compilation.

Because PM programs can invoke several instances of a procedure, each instance of a procedure must receive its own copy of certain variables. The LOCAL-STORAGE SECTION ensures that each procedure within the application receives its own copy of variables defined in this section. These variables are destroyed upon exiting the procedure.

The LINKAGE SECTION declares variables that can be passed as parameters to procedures. PM programs define entry points to procedures using the ENTRY statement with the USING clause. The parameters defined after the USING clause are declared in the LINKAGE SECTION of the application. PM now can pass parameters to procedures, similar to calling a COBOL subroutine or an OS/2 Dynamic Link Library (DLL).

```
Perform until EndFlagYes
    Call OS2API '__WinGetMsg'
          using  by value hab
                 by reference QMSG
                 by value ULongNull
                 by value UShortNull
                 by value UShortNull
          returning ReturnData
    If (QMSG-MSGID = WM-QUIT)
        Set EndFlagYes to true
    Else
        Call OS2API '__WinDispatchMsg'
              using  by value hab
                     by reference QMSG
              returning ReturnData
    End-If
End-perform
```

Figure 5. Main Message Routine

The PROCEDURE DIVISION contains the most noticeable changes of the four COBOL divisions. This division is where all API calls are coded. COBOL/PM programs follow the structured programming approach of initialization, process, and termination. The only difference is that the main processing routine serves only one purpose: to receive messages from PM and to dispatch these messages to the appropriate procedures. These procedures receive, evaluate, and either act on these messages (by performing some type of processing) or ignore them (by passing them back to PM to process or discard).

Initialization Routine

As PM applications become more sophisticated, so does the initialization routine. However, some functions must be taken care of, regardless of the complexity of the application. The initialization routine performs the following:

- Initializes PM facilities with the WinInitialize API
- Creates the application's message queue with WinCreateMsgQueue
- Sets the program's entry point to the main window procedure (Set ENTRY...)
- Registers the program's main window and associates it with the main procedure by making a call to WinRegisterClass
- Creates the program's main window with the WinCreateStdWindow API

Main Message Routine

The main message routine receives messages from the application's message queue through the WinGetMsg API call. Then these messages are dispatched to the appropriate window procedure with the WinDispatchMsg call. The main message routine continues to receive and dispatch messages until the application receives a WM-QUIT message.

Figure 5 shows the main message routine. Notice a double underscore (__) preceding each API call. This double underscore informs the COBOL compiler that the call is to an external routine. If all program calls are to external routines, omit double underscores and use the /LITLINK compiler directive.

Termination Routine

The termination routine destroys the main window with the WinDestroyWindow API call, destroys the message queue with the

```
ENTRY 'MainWndProc' using by value hwnd
                          by value Msg
                          by value MsgParm1
                          by value MsgParm2.
```

Figure 6. ENTRY Statement

```
NAME            PMABOUT WINDOWAPI
DATA            MULTIPLE
HEAPSIZE        16384
STACKSIZE       40960
PROTMODE
```

Figure 7. Module Definition File PMABOUT.DEF

`WinDestroyMsgQueue` API call, and terminates the use of PM services with the `WinTerminate` API call.

Main Window Procedure

The main window procedure is the first procedure in a COBOL/PM program and is usually associated with the application's main window. Most window procedures receive messages, evaluate the messages, and either act on the messages or return them to PM for default processing. The entry point to a procedure is coded with the `ENTRY` statement, as shown in Figure 6. The literal `MainWndProc` can be any literal, as long as it is defined in working storage as a procedure pointer and is declared in the module definition file under the `EXPORTS` statement.

The window procedure is associated with a window when the call to `WinRegisterClass` is made. A dialog procedure is associated with a dialog box when a call to `WinDlgBox` or `WinLoadDlg` is made. The only difference between a window procedure and a dialog procedure is that the former is associated with a window and the latter with a dialog box. The `EVALUATE` statement determines which message has been received, and the `WHEN` clause specifies which actions to take. If no actions are taken by the application, the message is dispatched back to PM for default processing.

Module Definition File

The module definition file (Figure 7) passes information about the program to the OS/2 Linker. It is an ASCII text file with file extension `.DEF`. This file is used most commonly to define heapsize and stacksize and to define data entry points within a DLL.

Creating Resources

The final step in developing a PM application involves creating all resources that the application will use – the windows, dialogs, icons, pointers, and other objects.

Resource File

First create the resource (`.RES`) file. As its name implies, this file is where all the application's resources are defined. The resource file contains definitions of windows, dialogs, icons, pointers, and other resources. The OS/2 resource compiler, provided with the OS/2 1.3 or OS/2 2.0 Toolkit, is used to create a resource file. The resource compiler produces a binary resource file with extension `.RES`.

Resource Definition File

The resource definition (`.RC`) file is an ASCII text file. This file can contain actual resource file definitions or point to other files that contain the resource definitions. For example, icons and pointers are binary files, so they should not be placed directly into the resource definition file. The resource definition file points to these binary files with the appropriate statements. During the compilation of the resource definition file, these binary files are retrieved by the resource compiler. The dialog definition file and resource header files are ASCII files that can be placed directly in the resource definition (`.RC`) file; however, most programmers keep these files separate for simplicity's sake. Figure 8 shows an example of a resource definition file. For more information about resource definition files, see the appropriate OS/2 programming guide for the toolkit that you are using.

Resource Header File

The resource header (`.H`) file is an ASCII text file that is used to assign values to user-specified objects (such as windows, dialogs, icons, pointers, pushbuttons, and list boxes). These objects are defined in the resource definition and dialog definition files. These values uniquely define each object in the application program. The resource header file can be edited using any ASCII text editor or the Dialog Editor utility. Figure 9 shows a resource header file. For more information about resource header files, see the appropriate OS/2 programming guide for the OS/2 toolkit that you are using.

Dialog Definition File

The dialog definition (`.DLG`) file is an ASCII text file that contains all dialog box definitions. You can create this file with any ASCII text editor; however, it usually is created and modified by using the Dialog Editor. Figure 10 shows a simple dialog definition file with a single `About` dialog defined.

Icon File

The icon (.ICO) file is a binary file that is created with the Icon Editor tool. The icon file contains a graphical representation of the program and is used to represent the application on the OS/2 Desktop or in a folder. The application's icon can be associated with the main window simply by giving the icon the same identifier as the main window (see the ICON statement in Figure 8). A pointer contains a graphical representation of a pointing device. The standard OS/2 pointing arrow can be changed to any other symbol (for example, a pencil or finger) to help users decide which action to take.

After all files mentioned in the previous sections are created, the iterating process of coding, compiling, and testing can begin. Start by invoking the make file created earlier. If a REXX build file was created, then simply type BUILD from the directory that contains all program files. If the NMAKE.EXE utility is being used, then begin by typing NMAKE PMABOUT.MAK from the OS/2 command line.

Summary of Files

Figure 11 summarizes typical files that a programmer works with when developing COBOL/PM applications. Figure 12 shows the overall relationships among input files, processes, and output files involved in creating the executable file PMABOUT.EXE.

COBOL/PM Tips

The *OS/2 1.3 Programming Tools and Information* refers to programming with the IBM COBOL/2 bindings. These bindings have been dropped in the OS/2 2.0 Developer's Toolkit and should not be used to develop any new COBOL applications. Statements that say COBOL is a non-reentrant programming language that is unable to support PM development simply are outdated.

```
#include "OS2.H"
#include "PMABOUT.H"

ICON WND_MainWnd PMABOUT.ICO

MENU WND_MainWnd PRELOAD
BEGIN
  SUBMENU  "~File",                    AB_File
    BEGIN
      MENUITEM "~New",                 AB_New,,     MIA_DISABLED
      MENUITEM "~Open...",             AB_Open,,    MIA_DISABLED
      MENUITEM SEPARATOR
      MENUITEM "~Save",                AB_Save,,    MIA_DISABLED
      MENUITEM "Save ~as...",          AB_Saveas,,  MIA_DISABLED
      MENUITEM SEPARATOR
      MENUITEM "~Exit",                AB_Exit
   END
  SUBMENU  "~Help",                    AB_Help
    BEGIN
      MENUITEM "~Help for help...",    AB_Hhelp,,   MIA_DISABLED
      MENUITEM "~Extended Help...",    AB_Xhelp,,   MIA_DISABLED
      MENUITEM "~Keys help...",        AB_Khelp,,   MIA_DISABLED
      MENUITEM "Help ~index...",       AB_Ihelp,,   MIA_DISABLED
      MENUITEM SEPARATOR
      MENUITEM "~About",               AB_About
    END
END

RCINCLUDE PMABOUT.DLG
```

Figure 8. Resource Definition File PMABOUT.RC

```
  #define WND_MainWnd          100
  #define AB_File              110
  #define   AB_New             111
  #define   AB_Open            112
  #define   AB_Save            113
  #define   AB_Saveas          114
  #define   AB_Exit            115
  #define AB_Help              190
  #define   AB_Hhelp           191
  #define   AB_Xhelp           192
  #define   AB_Khelp           193
  #define   AB_Ihelp           194
  #define   AB_About           195
  #define DLG_About            900
```

Figure 9. Resource Header File PMABOUT.H

The resource header file contains define statements for windows, dialogs, and controls in the resource definition and dialog definition files. To refer to these objects within COBOL source code, redefine them in WORKING-STORAGE as Pic 9(4) Comp-5 items. Be sure to use the same numeric value defined in the resource header file, or the message evaluation logic will be incorrect. Additionally, if an underscore (_) is used in the resource definition file statements (shown in Figure 9), change it to a dash (-) when you define these data items in WORKING-

```
DLGINCLUDE 1 "PMABOUT.H"

DLGTEMPLATE DLG_About LOADONCALL MOVEABLE DISCARDABLE
BEGIN
    DIALOG  "About", DLG_About, 21, 5, 228, 86, FS_NOBYTEALIGN |
            WS_VISIBLE, FCF_SYSMENU | FCF_TITLEBAR
    BEGIN
        CTEXT           "PMABOUT Version 1.0", 1902, 67, 66, 100, 8,
                        DT_VCENTER
        CTEXT           "by Chris Fierros (OS2AAC06 @ DALVM41B)", 1903, 0,
                        52, 228, 8, DT_VCENTER
        CTEXT           "IBM OS/2 Application Assistance Center", 1904, 0,
                        38, 228, 8, DT_VCENTER
        CTEXT           "A COBOL/PM Application", 1905, 0, 24, 228, 8,
                        DT_VCENTER
        DEFPUSHBUTTON   "OK", DID_OK, 77, 6, 69, 13
    END
END
```

Figure 10. Dialog Definition File PMABOUT.DLG

File Name	Function
BUILD.CMD	REXX build file (See Figure 2)
PMABOUT.MAK	NMAKE make file (See Figure 3)
PMABOUT.CBL	Source code
PMABOUT.CPY	Source code copy book
PMABOUT.DEF	Module definition file (See Figure 7)
PMABOUT.RC	Resource definition file (See Figure 8)
PMABOUT.H	Resource header file (See Figure 9)
PMABOUT.DLG	Dialog definition file (See Figure 10)
PMABOUT.ICO	Icon file
PMABOUT.RES	Binary resource file
PMABOUT.EXE	Executable file

Figure 11. Typical Files Used for COBOL/PM Application PMABOUT

STORAGE. For example, in Figure 9, the Action Bar pull-down is defined as AB_Help. This definition needs to be changed to AB-Help when this Action Bar pull-down is defined in WORKING-STORAGE.

Unlike the COBOL compiler, the resource compiler used to create the application's resource (.RES) file is case-sensitive. This means that Dlg_AboutBox and DLG_AboutBox are not the same. You will receive compiler errors about undefined variables. The COBOL compiler does not distinguish between upper- and lowercase. Coding COBOL in mixed-case is a personal preference, but it enhances readability. If you have ever received an E-mail note typed in all uppercase letters, you will appreciate reading source code that contains mixed-case letters.

The OS/2 2.0 Developer's Toolkit can be used under OS/2 2.0 to develop 16-bit PM applications that can run under both OS/2 1.3 and 2.0. You will need to link with OS2286.LIB. Remember, the online documentation that comes with the OS/2 2.0 Toolkit has been modified to reflect 32-bit API calls. This means that most of the API parameters have been changed from two-byte, Pic 9(4) Comp-5 definitions to four-byte, Pic 9(9) Comp-5 definitions.

Micro Focus COBOL Toolset has a utility called H2CPY that converts the PM header files into COBOL format. The utility converts #define statements into 78-level COBOL elementary items. These 78-level elementary items will not contain a PICTURE clause. The SIZE clause can be used when passing these items as parameters to PM API calls. For example, BY VALUE ITEM1 SIZE 2 passes the parameter ITEM1 as two bytes. A default byte size also can be specified with the LITVAL-SIZE compiler directive. For 16-bit applications, use LITVAL-SIZE"2" and only use the SIZE clause to pass a four-byte parameter. The conversion to 32-bit applications is simplified by changing to LITVAL-SIZE"4" to default to a four-byte parameter.

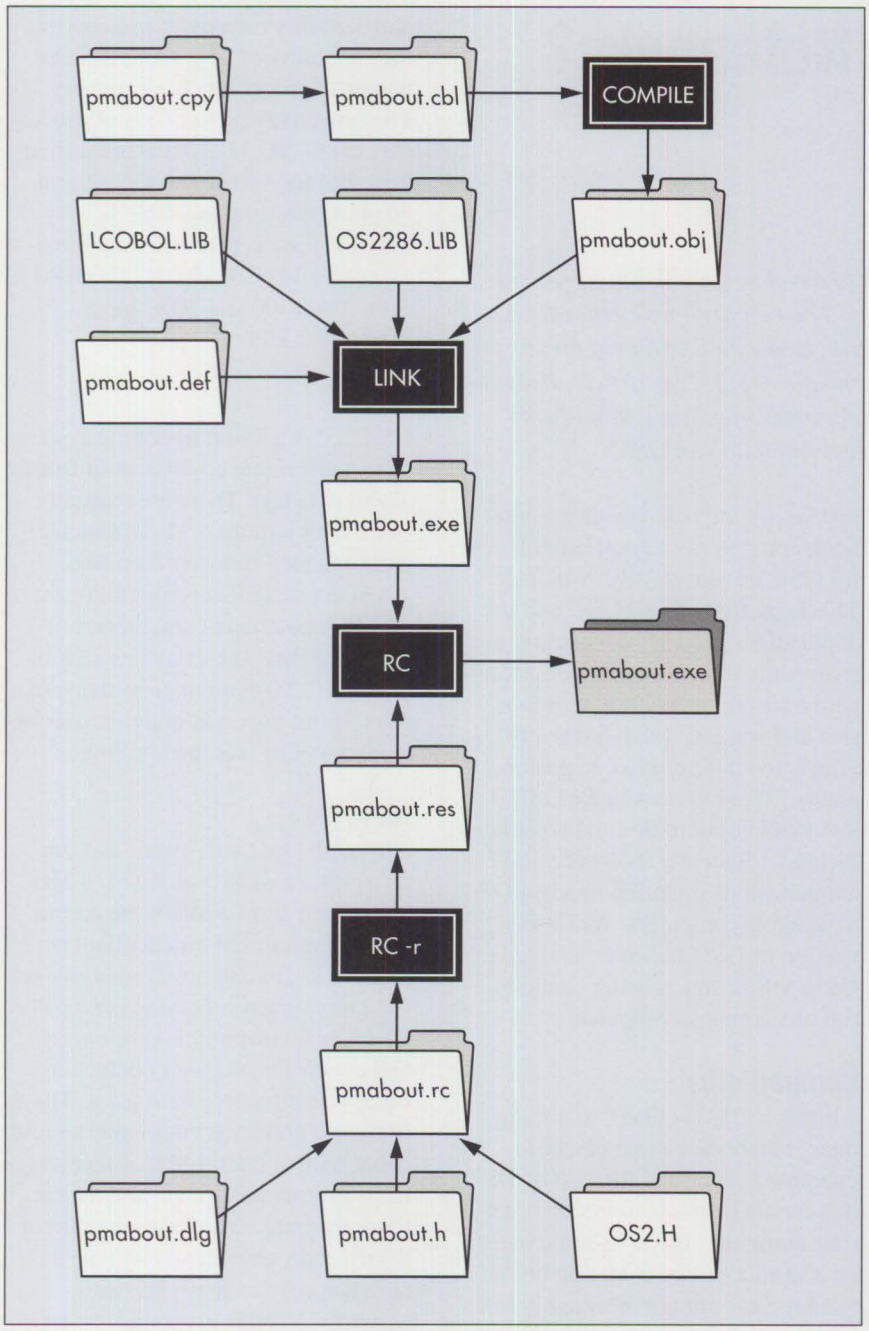


Figure 12. Building the Executable File PMABOUT.EXE

When using the Micro Focus COBOL/PM debugger, Xilerator, set break points and zoom through the application to these break points. Stepping through the application backs up the application's message queue.

PM API calls expect all text strings to be passed as ASCIIZ strings, meaning that they are null-terminated. The null terminator indicates the end of the character string. To add a null terminator, use `& x'00'` at the end of the string. To remove null terminators, use the `INSPECT` statement with the `REPLACING` option. In COBOL, the null terminator `x'00'` is the same as `Low-Values`.

Softcopy of PMABOUT Code

The sample code in this article can be downloaded in its entirety from the IBM National Solution Center Bulletin Board System, (404) 835-6600 (modem settings N-8-1). The source code is archived in the file `PMABOUT.ZIP`.

References

Documentation about both the COBOL programming language and OS/2 PM is provided with the corresponding software packages. In addition, *The COBOL Presentation Manager Programming Guide* (G362-0010) by David M. Dill is an excellent guide to COBOL/PM programming. This book provides both in-depth technical information and how-to sample code. The book also is sold in computer bookstores.

Christopher Fierros *is an OS/2 application specialist in the IBM OS/2 Application Assistance Center (OS2AAC) in Dallas, Texas. His primary focus is to provide expertise to IBM customers who are developing OS/2 applications using Micro Focus COBOL, the OS/2 PM graphical user interface, and the OS/2 Database Manager relational database engine. He is working toward a Bachelor of Business Administration degree in Business Computer Information Systems at the University of North Texas.*

OS/2: How About Notebooks?

Fabian Pascal
micro-paSQaL
Palo Alto, California

This article is part of ongoing research to develop specifications for selecting an OS/2-capable notebook computer. The research will generate a table of features to which OS/2 is sensitive, and which will be a means for comparing notebook products as platforms for OS/2. This article discusses the major factors about notebook computers that users should consider when selecting a notebook system on which to run OS/2.

Because OS/2 truly multitasks DOS and Windows applications, it is a natural platform for notebook computers. Notebook users are showing significant interest in running all of their applications under OS/2, rather than separately under their respective operating environments.

Generally, any computer that is larger than 8.5 x 11 inches, thicker than two inches, and heavier than six pounds is not truly a notebook computer. In fact, everything else being constant, any computer that exceeds these minimal requirements, still delivers good performance, has a good keyboard, and comes at an affordable price should be a serious candidate for OS/2. When considering a notebook computer's physical attributes, keep in mind the size and weight of the AC adapter and battery. These necessary items tend to be bulky and heavy, which can defeat the purpose of a true notebook, particularly for travelers who also are likely to carry a spare battery.

Despite technological advances that keep squeezing more power into constantly shrinking machines while reducing prices, and despite the endless proliferation of new products, finding an affordable notebook computer that runs OS/2 comfortably is far from easy. Many specifications that would indicate whether a notebook computer is a good candidate for OS/2 are not readily available. This is partly because OS/2 and its applications are very demanding in terms of hardware and power. More hardware and power require more size and weight, which adversely affects the ratio of price to performance. These factors make OS/2 a less viable option for notebook manufacturers. Recently, however, IBM announced its intention to adapt OS/2 to portable computers. When this version of OS/2 becomes available, OS/2 will be much better suited to run on notebook computers.

Compatibility

Although OS/2 is likely to run on many notebook computers, IBM guarantees to refund the cost of OS/2 if it cannot be made to run on a specific computer. Buyers should insist on a similar policy from notebook vendors, and should thoroughly test OS/2 and applications on a notebook computer before purchasing it.

Processor

The minimum processor to consider is a 80386 33 MHz. Until recently, 486 machines were rare, large, heavy, prone to overheating, and very expensive. But this situation is changing, so now I recommend the 486DX, 33 MHz version for those who can afford it. Even with a 33 MHz 486DX, and certainly for slower processors, the capability of upgrading to more powerful processors is a necessity. The power-saving versions of processors (SLC, SL, DXL) that are currently scarce for 486 processors should be preferred for reasonable battery life, and a socket for a math coprocessor should be available. The IBM ThinkPads 700 and 700C have a 486SLC/25 CPU.

Bus

OS/2 2.0 is a 32-bit operating system, so its performance will benefit from a 32-bit data bus. There are currently two 32-bit options – Micro Channel and Extended Industry Standard Architecture (EISA) – but there are no EISA notebook computers, but the ThinkPad 700 series has a 32-bit bus. OS/2 2.0 can run on systems that have 16-bit buses, although its performance will be less than optimal.

Memory

Although OS/2 will install and run on as little as 4 MB of RAM, 8 MB is required and 12 MB is the recommended minimum for excellent performance. The memory requirement is a crucial factor, because practically all notebook memory is proprietary and expensive, and can double an otherwise attractive base price. The preferred notebook computers should come with at least 4 MB of memory, have a reasonable price, and accept more than the common maximum of 8 MB. They also should be able to add memory gradually in small amounts. Notebook computers that offer only large add-on memory modules, or force replacing the initial memory module with a larger one, should be avoided. The ThinkPads come with 4 MB, and allow two additional modules of 2, 4, or 8 MB, up to the addressable maximum of 16 MB.

Vendors do not usually publish memory speed. Although 80 ns is common, 70 ns and even 60 ns, although rarer, are preferred for OS/2.

Disk Drives

In most notebook computers, the size of hard disks has hovered around 60 to 80 MB. OS/2 and its applications will quickly fill that space. It is best to have at least 120 MB. Now 200 MB drives are starting to appear. File compression would be a good improvement to OS/2 because hard disks with a large capacity and a small factor (a size of 2.5 inches or less) are expensive. Because OS/2 swaps to disk – especially when a computer has a relatively small amount of memory – 15 ms average access time is the minimum acceptable performance for hard disks. The ThinkPad drive is removable; the optional 120 MB disk drive (standard with the 700C) has an access time of 17 ms.

Frequent backups of hard disks are good policy, but diskettes are not a viable proposition. Access to an external backup device – a hard disk, a removable tape drive, or cartridge drive – should be available. Built-in Small Computer Systems Interface (SCSI) technology enables access to seven compliant devices, but most notebooks come with IDE drives, and small-factor SCSI drives are not available. A docking station such as the one used by the ThinkPads, or an adapter that allows connection to SCSI devices through the parallel port should be considered.

Data transfer rates are not published by vendors, but the higher, the better.

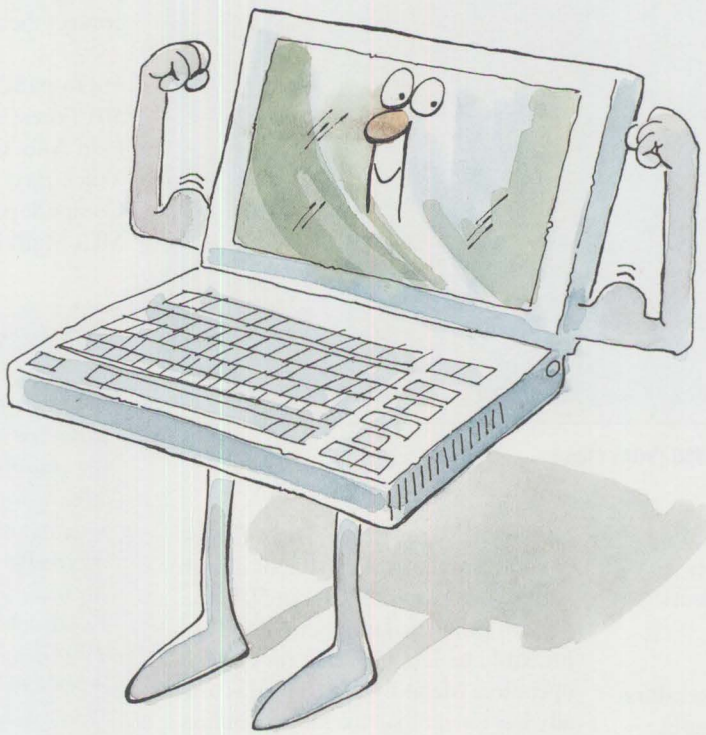
Video

A resolution of 1024 x 768 is the minimum resolution that really exploits the Workplace Shell. This resolution requires IBM's XGA/2, XGA, or 8514/A video adapters, or OS/2-specific Super VGA drivers from other vendors' computers. But these drivers are not yet available. The 10-inch maximum display of notebooks makes the usefulness of resolutions higher than 640 x 480 (VGA mode) questionable (the ThinkPad 700C has a 10.4" screen).

Figure 1 shows the OS/2 desktop in 1024 x 768, 8514/A mode. Figure 2 shows the same screen in VGA mode, where a large portion of the desktop is no longer visible.

A color display on a notebook computer is preferable. However, color displays are expensive and demand a lot of power, so most users will have monochrome displays. The quality of the display, the maximum number of shades (16, 32, 64), and performance will vary considerably across notebook products. There is little information about the amount of video RAM, which is important for performance. At least 1 MB is recommended.

Most notebooks offer access to an external display, and some of these displays have slightly higher resolution than VGA, usually 800 x 600. This resolution is less than ideal for OS/2 and also requires a Super VGA driver that is rarely available. If the notebook is used for presentations, the capability to display simultane-



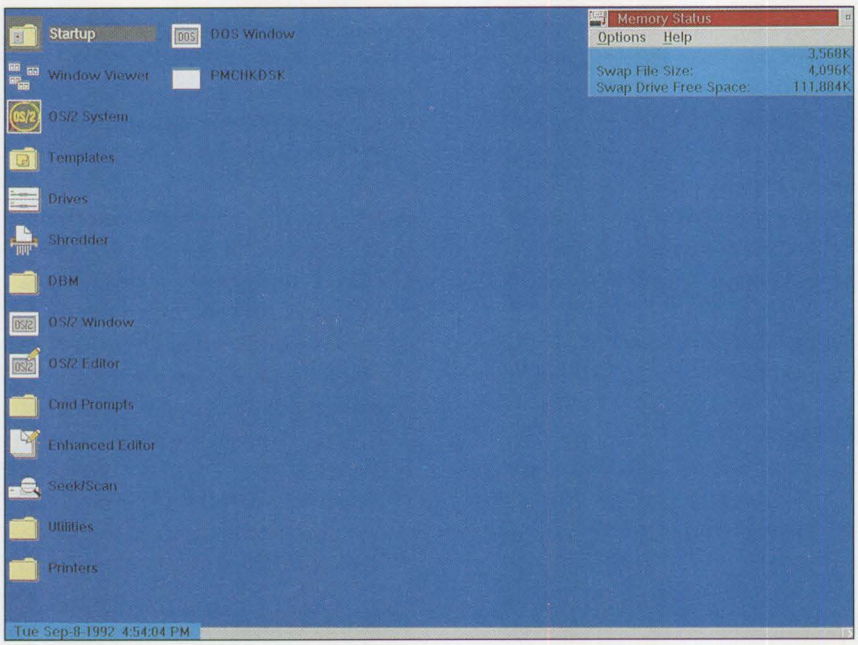


Figure 1. OS/2 Desktop in 1024 x 768 (8514/A) Mode

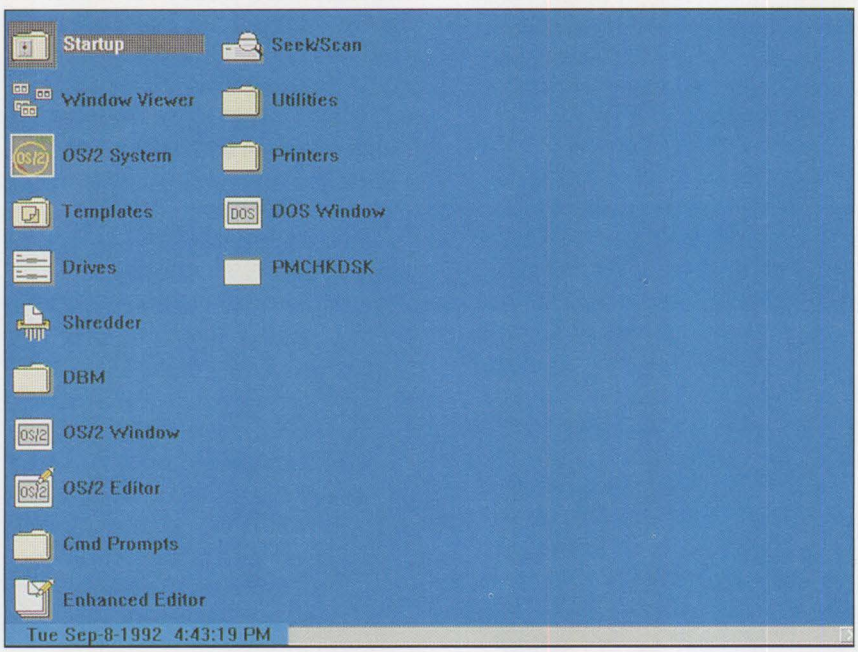


Figure 2. OS/2 Desktop in 640 x 480 (VGA) Mode

ously on the notebook's built-in screen and an external screen is important (the ThinkPads have it).

Power

The battery life published by vendors is for DOS operation and is useful only for very rough comparisons between computers. As a rule of thumb, battery life is actually about 50% to 60% of published numbers. It will probably be less than that for OS/2 operation. Metal-hydride batteries usually last longer than the more common nickel-cadmium ones, but not all vendors offer them (the ThinkPads do). A spare battery is not an option for travelers, it is a necessity – but few vendors include one in their product packages. The recharge time should be as short as possible, but vendors do not always publish that figure.

Pointing Device

Built-in trackball devices are most convenient with traveling notebooks, but they vary in type and quality, and not all users are comfortable with them. Mice are not as easy to travel with, but they are more popular and they also can serve a desktop. Vendors should offer both options and let the user choose which one to use and when. The ThinkPads have a new type of pointing device, the TrackPoint, embedded in the keyboard, that may resolve the dilemma. Notebook computers that have neither a built-in device nor a second serial port make the simultaneous use of a mouse and modem impossible.

Vendors and users interested in supporting this research are invited to contact the author.

micro-paSQaL
501 Forest Avenue, #703
Palo Alto, CA 94301
Voice/Fax: (415) 325-0646
CompuServe: 73677,3306
MCI Mail: FPascal

Fabian Pascal *is an independent consultant, teacher, and writer specializing in relational database management and SQL, and in office and portable computing. As an industry analyst, he teaches seminars, advises, and has published extensively about these subjects. Author of* SQL and Relational Basics *(M&T Books, 1990), he is currently working on a book to be published by John Wiley & Sons in 1993.*

Loadable ABIOS

Richard Bealkowski
IBM Corporation
Boca Raton, Florida

IBM is changing the way it distributes the Advanced Basic Input/Output System (ABIOS) on PS/2s. New IBM PS/2s, such as the IBM PS/2 Model 9557, support the ABIOS that is available on diskette or factory-loaded on the machine's hard disk. This article introduces the concept of a loadable ABIOS, explains why the change was made, and discusses its impact on users.

Applications and the operating system go through Basic Input/Output System (BIOS) routines to access devices on a personal computer. In most cases, only the operating system interfaces directly with the BIOS. Most applications call the operating system to perform these functions, although a few special-purpose applications bypass the operating system interface and call the BIOS directly.

When OS/2 was introduced, IBM began supplying an ABIOS containing functions previously not available in the PC BIOS. To maintain compatibility with other programs, the original PC BIOS – renamed Compatibility Basic Input/Output System (CBIOS) – is also included with the PS/2. Until recently, ABIOS was part of the resident PS/2 *firmware*. Firmware refers to programs that are typically stored in a nonvolatile memory device, such as Read-Only Memory (ROM). PS/2 firmware includes a Power-On Self Test (POST) in addition to the CBIOS and ABIOS.

Over time, all IBM PS/2s that support ABIOS will use a diskette-based ABIOS instead of a firmware-based ABIOS, making it easier to maintain and update.

Once loaded, ABIOS is compatible with resident ABIOS. Software that uses ABIOS directly, however, may require changes in the initial installation procedure. DOS is not affected because it does not use ABIOS, but DOS applications can access it. The installation routines of OS/2 2.0 do not support a loadable ABIOS, but future releases will. In the meantime, PS/2s with a loadable ABIOS are pre-loaded with OS/2.

ABIOS has a device-level Application Programming Interface (API). ABIOS can operate in real mode, in protected mode, or in a bimodal environment using both real and protected modes. ABIOS is described thoroughly in the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* (S04G-3283).

Resident ABIOS

Resident ABIOS comes in three main types: system firmware, adapter firmware, and RAM extensions. Resident ABIOS occupies a substantial amount of memory in the adapter (feature) space and in the system firmware space. The adapter address space is the C000h (hex) and D000h segments. The system firmware address space is the E000h and F000h segments. Adapter address space and system firmware address space provide space for firmware, device buffers, and Expanded Memory Specification (EMS) buffers. The available space within both the system firmware and adapter address spaces has become very limited. Some systems may not have any available space. To solve this problem, the resident ABIOS has been packaged as a separate program. Adapter ABIOSs also can be packaged as separate programs.

Initialization of the resident ABIOS occurs in three phases. Phase one is the initialization of the system ABIOS. Phase two is the initialization of the adapter ABIOS. Phase three is the initialization of ABIOS RAM extensions such as patches. Two CBIOS calls, `INT 15h AH=04` ("Build System Parameters Table") and `INT 15h AH=05` ("Build Initialization Table") are needed to initialize ABIOS. Once an initialization call is invoked, the ABIOS coordinates the initialization not only of itself, but of the adapter and RAM extension code as well.

Loadable ABIOS

The separate ABIOS programs must first be loaded into system RAM by the operating system or special-purpose application before they can be accessed. These now-separate ABIOS programs are *loadable ABIOS*.

Adapters that store their ABIOS in 16-bit ROM may perform better when the ABIOS is then run from 32-bit system RAM.

The loadable ABIOS architecture is based on the existing RAM-Extension Structure described in the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference*. For example, an ABIOS patch adheres to the RAM Extension Structure architecture. Before software, such as an operating system, initializes ABIOS through the CBIOS calls (`INT 15h AH=04` and `INT 15h AH=05`), it must load all RAM extensions into memory and set the DS register to point to the beginning of the RAM extensions. Even if no RAM extensions (patches) are

required, DS must still be properly set and must point to a valid, zero-length ABIOS RAM extension. No changes are required to the runtime portion of software that adheres to the RAM Extension Structure section of the ABIOS architecture.

IBM's loadable ABIOS is shipped with the PS/2 reference programs. Initially, the reference programs can reside on the Reference Partition of the hard disk or on the Reference Diskette. If the diskette version is not available, it must be created because the hard disk version is not available during normal runtime.

The Backup feature of the Reference Partition is used to create the Reference Diskette. This diskette-based version of the reference programs contains the loadable ABIOS that is used during installation of the operating system or special-purpose application. For systems that support only a preloaded operating system, the user may never encounter the requirement for a Reference Diskette. Systems that are user-installed and require ABIOS, however, will require the Reference Diskette.

Module Header

Every loadable ABIOS module is required to have a valid RAM extension header, which is an extension of the existing RAM extension header architecture. Figure 1 shows the fields of the RAM extension header. The last four entries have been added since the initial release of resident ABIOS.

The new fields in the RAM extension header are as follows:

0Eh Extension Header Length (in bytes). This word value indicates the number of bytes in the extended header. A value of zero indicates there is no extension to the header.

Field	Offset
Signature = AA55h (word value)	+00h
Length in 512-Byte Blocks	+02h
Model Byte	+03h
Submodel Byte	+04h
ROM Revision Level	+05h
Device ID	+06h
Number of Initialization Table Entries	+08h
Build Initialization Table Entry Point	+09h
Secondary Device ID	+0Ch
Revision	+0Dh
Extension Header Length (in bytes)	+0Eh
Support Determination Routine Offset	+10h
Length (in bytes) of Extension Without Fill	+12h
Initialization Routine Offset	+14h

Figure 1. ABIOS RAM Extension Header

10h Support Determination Routine Offset. This routine returns a zero value when this RAM extension does not apply to this computer system. When it does apply, this routine returns the actual ABIOS module length in bytes. This routine should be called during the loading of RAM extensions. Only RAM extensions that return a nonzero value should be allowed to remain in memory.

12h Length (in bytes) of this RAM extension without any fill. This is the actual length of the RAM extension, and does not include any pad that may have been added to raise the module size to the nearest 512-byte boundary.

14h Initialization Routine Offset. This routine is called by the CBIOS `INT 15h AH=04` and `INT 15h AH=05` code, and is the ABIOS initialization program.

Wild Card Values

The model, submodel, and revision level values stored in the ABIOS RAM extension header can be specific values or wild card values. A wild card entry is specified by the value zero. Any or all of the model, submodel, and revision fields can be set to the wild card value. When comparing the computer system's model, submodel, and revision fields with those found in the ABIOS module header, a wild card value indicates an automatic match of that field.

Support Determination Routine

This real-mode-only routine should be called only if the extended header length includes the support determination routine entry, and the entry is nonzero. Figure 2 shows the parameters of the support determination routine.

The entry parameters of model, submodel, and revision level should be those of the current system. These values can be obtained from CBIOS through the `INT 15h AH=C0h` call.

The support determination routine finds out if particular ABIOS RAM extensions apply to the computer system. Typically, the RAM extension header contains wild card entries for model, submodel, and revision level values. The support determination routine should be called immediately after the RAM extension has been loaded into memory. If the support determination routine returns a length of zero, then the RAM extension does not apply to the current system. If the support determination routine returns a nonzero length, the RAM extension does apply to the current system.

The size (length) of an ABIOS RAM extension must be an even multiple of 512. The ABIOS initialization program uses the length value (in 512-byte blocks) at offset +02h in the RAM extension header to calculate the beginning address of the next RAM extension header.

When the last RAM extension loaded does not apply to the system, the RAM extension should be invalidated or otherwise cleared from memory. This prevents the ABIOS initialization routine from locating and initializing a RAM extension that is not applicable to the system. One method to invalidate a RAM extension is to zero out the signature in the RAM extension header.

Compaction

The support determination routine is normally required only during initialization. It may be possible to compact an ABIOS RAM extension before the next ABIOS RAM extension is loaded. After the support determination routine is called, the ABIOS RAM extension header can be updated as shown in Figure 3.

When altering an entry in the extended portion of the RAM extension header (beyond +0Eh), the extension header length (+0Eh) must be checked.

Input	
Parameter	**Description**
(AL)	System Model Byte
(AH)	System Submodel Byte
(BL)	System ROM Revision Level
Output[1]	
Parameter	**Description**
(AX)	Length of Extension in Bytes (actual)
(BX)	Length of Extension in Bytes (512-byte multiple)
(CL)	Count of 512-byte Blocks
Flags	Undefined

[1]All other registers preserved

Figure 2. Support Determination Routine Parameters

```
Header.02h = CL     ; new 512-byte block count
Header.12h = AX     ; new length without fill
Header.10h = 0      ; support determination routine disabled
```

Figure 3. Updating the ABIOS RAM Extension Header

The check must be performed to ensure that the header includes the target field.

Implementation

To use ABIOS, the software should first perform a *presence detect* to determine what type (if any) of ABIOS support is provided. ABIOS presence detect consists of several individual tests performed in a predetermined sequence. No individual test is definitive. The test sequence must be performed to determine whether ABIOS is supported, and if it is, what type of ABIOS is present. The software must support incorporating the loadable ABIOS files as part of its installation process. The runtime operation of the software must be consistent with the established ABIOS architecture.

Build System Parameters Table

Build System Parameters Table is the CBIOS `INT 15h AH=04` function call – one of the two CBIOS calls used to initialize ABIOS. It has been discovered that certain software may issue this call as a presence test for ABIOS, only to perform the call later during actual initialization. If this call is made before loadable ABIOS is loaded, this call will not succeed, thus giving the impression that ABIOS is not supported.

ABIOS Attribute Field

This field is returned as part of the Return System Configuration Parameters function, `INT 15h AH=C0h`. All Micro Channel-based PS/2s support the Return System Configuration Parameters function, but not all support the bit settings of the ABIOS Attribute Field. The register pair ES:BX points to a table in memory that has the structure shown in Figure 4. The ABIOS Attribute Field is defined as bits 5-3 of Byte_4 in Figure 4.

Offset	Field	Size	Description
0	Length	Word Table Length	
2	Model	Byte	System Model
3	Submodel	Byte	System Submodel
4	Level	Byte	Firmware Revision
5	Byte_1	Byte	
6	Byte_2	Byte	
7	Byte_3	Byte	
8	Byte_4	Byte	ABIOS Information

Figure 4. System Configuration Table

The length field must be checked to determine if the table includes Byte_4. If Byte_4 is a valid table entry, then the contents of bits 5-3 of Byte_4 can be used to determine the status shown in Figure 5.

Loadable ABIOS Signature

The ABIOS signature byte is a loadable ABIOS presence test method that supports retrofitting loadable ABIOS into existing resident ABIOS systems. For example, it may be desirable to install a new disk controller into an existing resident ABIOS system; however, the new disk controller requires loadable ABIOS. To indicate this to software such as an operating system, an indicator is stored in system NVRAM location 202h. If a loadable ABIOS is required, this byte is set to A1h. If loadable ABIOS is not required, the byte is set to something other than A1h (preferably 00h). An Adapter Description Program (ADP) that ships with a system or an option that requires loadable ABIOS must program this byte accordingly. Note that both systems and options (adapters) must support the loadable ABIOS signature byte.

Loadable ABIOS Signature Interface

A CBIOS function call, `INT 15h AH=A0h`, can be used to obtain the ABIOS signature byte. If this call returns successfully, it provides the ABIOS signature byte. If this call does not complete successfully, NVRAM must be accessed directly. This call can also be used in systems where NVRAM does not exist, or in systems where NVRAM location 202h is unavailable, because the call masks the actual means used to store the ABIOS signature byte.

Bits 5-3[1]	Description
000	ABIOS Attribute information not available
001	ABIOS not supported
010	Resident ABIOS
011	Loadable ABIOS

[1]Values of 100 through 111 are reserved.

Figure 5. ABIOS Type Information

The interface specification for the ABIOS signature byte interface is shown in Figure 6.

Detection Algorithm

Figure 7 shows how to determine the level of ABIOS support present in a system. In Figure 7, *No ABIOS* indicates that the system does not support

Call		
(AL) = 00h		Read Loadable ABIOS Signature
Return		
CF = 0		Operation successfully completed
	(AH) = 00h	Operation successfully completed
	(BL)	Loadable ABIOS Signature value
	= 00h	Loadable ABIOS prompting not required
	= A1h	Loadable ABIOS prompting required
CF = 1		Operation failed
	(AH) = 02h	Unable to read Loadable ABIOS Signature
Call		
(AL) = 01h		Write Loadable ABIOS Signature
	(BL)	Loadable ABIOS Signature to write
	= 00h	Loadable ABIOS prompting not required
	= A1h	Loadable ABIOS prompting required
Return		
CF = 0		Operation successfully completed
	(AH) = 00h	Operation successfully completed
CF = 1		Operation failed
	(AH) = 02h	Unable to write Loadable ABIOS Signature

Figure 6. ABIOS Signature Byte Interface

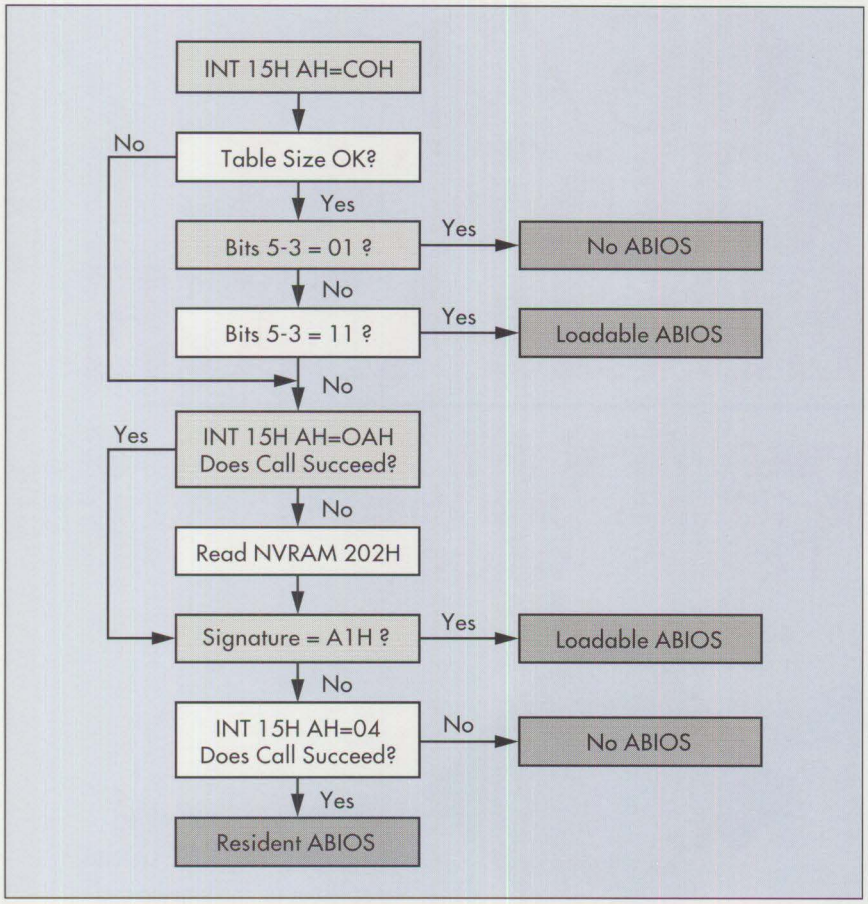


Figure 7. Detection Algorithm

ABIOS; *Resident ABIOS* indicates that ABIOS is supported and that no loadable ABIOS files are required; and *Loadable ABIOS* indicates that at least one loadable ABIOS file is required (and resident ABIOS may also be present).

Installation

The Reference Diskette contains one or more loadable ABIOS files, an `ABIOS.SYS` file, and one or more Device Driver Profile (`.DDP`) files. The loadable ABIOS files have an extension of `.BIO`. The `ABIOS.SYS` file lists all the `.BIO` files that may be required for the system. The `.DDP` file is a list of required `.BIO` files in the DDP format. The `.DDP` file can be used as a control file to drive the installation of loadable ABIOS. For example, the OS/2 program `DDINSTAL.EXE` operates using `.DDP` files.

The order in which the `.BIO` files appear in `ABIOS.SYS` and the `.DDP` file is important. The files that contain an initialization routine must precede all other files in the list. The initialization routine scans forward in memory for other ABIOS modules, so it must come before any other `.BIO` files.

Software (such as an operating system) that is preloaded or preconfigured on a computer system may not have to involve the user in the loadable ABIOS installation process. However, during a user-based installation, software that uses ABIOS must take an additional step to load ABIOS. If loadable ABIOS is detected, there must be a prompt for the ABIOS source diskette (the Reference Diskette), and the ABIOS must be loaded.

Developer Considerations

The runtime operation of operating system or special-purpose software should be identical for both loadable and resident ABIOS. There are some considerations that will help ensure this:

- Support the RAM-Extension Structure architecture.
- Do not assume ABIOS resides at any particular physical address (that is, in the E000h segment).
- Allow for adequate RAM extension space.
- Allow for more than one RAM extension per system.

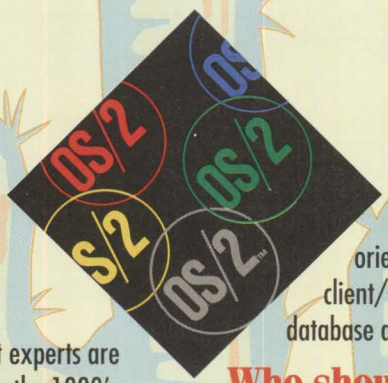
User Considerations

Users of PS/2s with loadable ABIOSs must be sure that the version of OS/2 install supports loadable ABIOS. If it does not, contact your dealer. Users of custom applications that access ABIOS directly should also ensure that the install portion of the application supports loadable ABIOS.

Richard Bealkowski *is an advisory engineer in Engineering Software, Entry Systems Technology Laboratory, Boca Raton, Florida. Since joining IBM in 1982, he has developed firmware for IBM personal computers and PS/2 systems. Richard has achieved the Tenth Level Invention Plateau. He also is the recipient of an Outstanding Innovation Award and an Entry Systems Division Excellence Award. At Florida Atlantic University, he earned a BS in mathematics, MS in computer science, and PhD in computer engineering.*

OS/2 TECHNICAL INTERCHANGE

February 28-March 3,1993
Pointe Hilton at South Mountain
Phoenix, Arizona



Seize the Next Frontier of Computing with OS/2!

OS/2 is the computing environment that experts are calling the winning software platform for the 1990's. You are invited to join top software professionals from throughout the world in Phoenix this February to explore the latest technical breakthroughs and the incredible potential of OS/2 for yourself.

The exciting OS/2 Technical Interchange promises to be the largest OS/2 conference and exhibition ever for programmers, technical coordinators and other software professionals.

New Tools...New Capabilities ...New Solutions

This exciting conference will feature some of the best speakers in the industry. Senior designers from IBM, along with top industry experts and co-sponsors Borland, Lotus and Computer Associates, will share the latest breakthroughs with you in interactive sessions that range from 32-bit graphics to object-oriented programming, LAN systems, client/server issues to multimedia and database applications.

Who should attend?

Technical coordinators, independent programmers, LAN experts, MIS managers, software designers, corporate developers, consultants and training executives will profit from the idea and information exchange. You'll learn powerful new programming skills and discover new ways to exploit the vast capabilities of OS/2 technology.

Each attendee will receive a CD ROM packed with valuable software, including the latest beta release of OS/2, application tools, utilities, programming libraries and more. (The software alone is well worth the price of registration!)

Keynote Speaker: J.A. Cannavino IBM V.P. & General Manager, Personal Systems

OS/2 The POWER of the future...NOW

For registration or exhibit information call 1-800-438-6720 or 203-261-6227.

A Preview of the fascinating OS/2 sessions* you can attend:

OS/2 Today and in the Future: Including Architecture, Application and Device Driver Considerations for a Microkernel-based OS/2

- The Latest Features and Capabilities
- OS/2 Architecture on the Microkernel
- Application Programming Considerations
- System and Application Performance Tuning
- Exploiting the Workplace Shell

OS/2 and Object-Oriented Programming Systems (OOPS)

- Object-Oriented Programming Development Overview
- Exploiting OS/2 Object Technology with the Workplace Shell and System Object Model (SOM)
- Objects in a Distributed Environment
- Object-Oriented REXX Technology and Visual Technology

Application Development

- Making the Most of OS/2 Development Tools
- Migrating Windows™ Applications to OS/2: Including a Hands-On Porting Lab
- Unleashing 32-bit Application Power in OS/2
- Migrating to 32-bit C
- Performance Tuning of 32-bit C OS/2 Applications
- Lab: Ask the Experts
- Lab: Using IBM C++ Class Libraries
- Lab: Performance Tuning Your C Application
- Lab: Introduction to Using C++

Developing Device Drivers

- Exploiting Printers and Display Subsystems
- Writing "Seamless" Display Drivers
- Technical Review of Virtual Device Drivers

PEN Computing with OS/2

- Portable PEN and Networked PEN Computing
- PEN Application Development Tools and Techniques
- Adding PEN Capabilities to Existing Applications

Speech Recognition with OS/2

- Speech Technology: State of the Industry
- Speech Technology: Application Development Strategy
- Workplace Shell Navigation Using Speech Recognition

OS/2 MultiMedia

- OS/2 MultiMedia Presentation Manager/2: An Overview
- Plugging into MultiMedia Presentation Manager/2
- Getting Started with OS/2 and MM PM/2 Ultimotion™ and Ultimedia Matinee™
- Tools for MultiMedia Application Development
- IBM MultiMedia Distributed Computing within the Enterprise

Local Area Networks (LAN)

- LAN Systems Strategy
- OS/2 LAN Server Tuning, Performance and Optimization
- OS/2 LAN Server Security
- Network Transport Services/2
- LAN Server/Novell/Banyan and TCP/IP Coexistence
- Peer Capabilities
- DOS LAN Requester
- Remote LAN Access
- Distributed Systems Services
- Future Network Operating Systems Support

Distributed Systems Management

- Framework Overview
- Application Overview
- Enabling Applications with Configuration, Installation and Distribution Services
- Problem Management: Fix/2 and Netview Tie/2
- LAN Automated Distribution Services

Communications Manager and Advanced Program-to-Program Communications

- Communications Manager/2: The Application Developer's View
- Installation and Configuration Tips and Techniques
- Communications Manager Client/Server Implementation
- What's New for APPC in Workstations
- Converting 3270 Applications to APPC Client/Server Applications

OS/2 Database Manager: Today and Future

- Distributed Database Connection Services/2
- Developing Client/Server Database Applications
- Database Manager Forward Recovery

Technical Coordinator Sessions

- Technical Coordinator Program
- IBM Personal Systems Services and Support

*** Sessions subject to change**

Save $100

Conference registration is only $595 if you register by January 15 ,1993. Space is limited. Register NOW!

For further information please call: **1-800-438-6720**
or mail this coupon today to:

IBM Corporation
OS/2 Technology Interchange
731 Main Street, Suite C-3
Monroe, CT 06468

Name _____

Company _____ Telephone _____

Address _____

City/State/Zip _____

Little Solutions

We invite you to share your "little solutions" in this column. Send them to us in care of the editor.

What is DIRCMD?

Just what is DIRCMD? DIRCMD is a little known environment variable that can be used to set a default list of flags normally used with the DIR command. Remember when the only flags for DIR were /P and /W? For some time now, there has been a growing list of flags that enable DIR to return all sorts of information and give several options for sorts and included files. Many of these flags apply to both DOS and OS/2, but this article is limited to OS/2 2.0. Look in the *Command Reference* for DOS or OS/2 1.X or use the online help to see what is available.

Who really takes the time to look at the manual for improvements or additions to the lowly DIR command? Not many of us. So let's take a look at what is available and then we will come back to the DIRCMD environment variable.

Flag	Description	Flag	Description
/W	Wide display	/O	Ordered list
/P	Paged mode	/S	List subdirectories
/F	Full path name	/B	Basic display
/N	HPFS-like FAT display	/L	List is in lowercase
/A	Controls attributes	/R	FAT long file names

Figure 1. OS/2 2.0 DIR Command Flags

There are now ten flags that can be used either individually or in combination with the DIR command, as shown in Figure 1. Two flags have additional options as well.

Let's take a look at what the different flags can do. The /W (wide) and the /P (page mode) flags have been around the longest. The /W flag lists all files (in the directory or subdirectory) in multiple columns on the display. This view enables you to see more items on a single screen than the normal DIR command, but does not display the size or the date and time that the file was last modified. The /P flag presents any of DIR's list formats one page (screen) at a time. This becomes really useful when using one of the new high-performance machines that can scroll a directory with hundreds of entries in the blink of an eye.

The /F and /N flags were available in OS/2 1.3. The /F flag lists a fully qualified path name of each file in a directory. No other information is displayed, including any heading data. The /N flag causes DIR to display the list of files on a FAT drive with the High-Performance File System (HPFS) format. Using /N on an HPFS drive has no effect. The four flags described above have been around for awhile and perhaps are not the most exciting features of the DIR command. Four of the next six flags offer some interesting functionality. You can decide about the last two.

The /A flag enables you to include or exclude files based on their attributes. For example, entering DIR /ASH on the command line will display only those files that are "hidden" or "system files." The options for the /A flag are shown in Figure 2.

Since the flags can be used in combination, so can the options to those flags as I've done in the command `DIR /AHS`.

The next flag is the `/O` (order) flag, which will modify the sorted order of a list of files. Files on HPFS drives are automatically displayed in ascending alphanumeric order when the `DIR` command is issued. Files on FAT drives are displayed in the order in which the `DIR` command found them – not always a logical order. This flag can be used to override the natural order for HPFS drives and to impose an order on FAT drives. Figure 3 shows the options. Try `DIR /OGS` and you will see that the subdirectories will be displayed first, followed by the files sorted by their size, smallest first.

There is finally a way to list files in all subdirectories. `DIR /S` will list all files in the current directory and all of its subdirectories. To list all executable files on a drive or partition, type `DIR *.EXE /S`. To list all files first (including hidden and system files), then subdirectories, in reverse order of their extension, in all subdirectories on the drive, and one page (screen) at a time, use the following:

```
DIR /OG-E /A /S /P
```

Two of the last three are rather simple and control how various listings are displayed. The `/L` flag will force all `DIR` command formats to display file names and subdirectories in lowercase format. The `/B` flag lists just the file and subdirectory names with no other data. Also, there is no header information or summary included. The last flag, `/R`, is not yet operational. However, it is intended

Flag	Description	Flag	Description
H	Hidden files	-D	No directories, files only
-H	Files that are not hidden	A	Files changed since last backup
S	System files	-A	Unchanged files
-S	Files that are not system files	R	Read-only files
D	Directories	-R	Files that are not read-only

Figure 2. Options for the /A Flag

Flag	Description	Flag	Description
N	Alphabetic by names, ascending	-D	Date and time, latest first
-N	Alphabetic by names, descending	S	Size, ascending
E	Alphabetic by extension, ascending	-S	Size, descending
-E	Alphabetic by extension, descending	G	Directories before files
D	Date and time, earliest first	-G	Directories after files

Figure 3. Options for the /O Flag

to display the original file name after a file has been copied from a system that supports long file names (HPFS) to a file system that does not support long file names (FAT). For example, if a file exists on an HPFS drive with the name `Long File Name.Text` and it is copied to a FAT drive, the name would be truncated and converted to a legal 8.3 FAT format. Something like `LONG_FIL.TEX` would result. However, the actual long file name would be stored in the file's Extended Attributes (EAs). `DIR /R` will retrieve the original file name from the EAs and display both.

Now it is time to get back to `DIRCMD` and explain where it fits. The new functionality of the `DIR` command is a powerful and welcome addition. Setting an environment variable with your favorite flags for `DIR` would make that same functionality more convenient. So place the following line in your `CONFIG.SYS` file, and each time you type `DIR`, you will get a list of directories followed by a list of files, both sorted alphabetically. All hidden and system files will be included. The list will be formatted in the HPFS style and will be in lowercase. Try it!

```
SET DIRCMD=/OGN /A /N /L
```

For more information, type `HELP DIR` on the command line or read the entire description in the online *Command Reference*.

—Larry Pollis, IBM Corporation, Dallas, Texas

Index to Past Issues of IBM Personal Systems Technical Solutions

October 1992 (G325-5019)

Exploring File Server Performance
PS/2 3.5-Inch Rewritable Optical Drive
Programming the XGA Video POS Registers
Video Monitoring on Personal Computers
Memory Address Space
OS/2 2.0 Installation and Performance Considerations
OS/2 2.0 Application Support
Cleaner Installation of Applications Under OS/2
Creating Resizable Pushbuttons
Configuring Parallel Ports for OS/2
Performance Characteristics of ES 1.0 Database Manager
AlertVIEW
Screen Reader/2

July 1992 (G325-5017)

IBM PS/2 Server 295: New Thresholds for Client/Server Networking
Comparing Architectures: Micro Channel and EISA (Part 2)
Synergy by Design
Pen-Based Computers
Why Doesn't My Portable's Battery Last Longer?
Planning Guidelines for Token-Ring Cabling
Installing and Migrating Applications in OS/2 2.0
Printing Under OS/2 2.0
Installing the IBM 4029 LaserPrinter Under OS/2 1.3
Serviceability Tools in OS/2 2.0
Online Communication Using the OS/2 2.0 PM Terminal
IBM Extended Services Database Manager
NetWare for SAA
Using the IBM DOS 5.0 Driver EMM386.EXE and Upper Memory
The Solutions Evaluation Tool

April 1992 (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

"IBM has a long history of establishing security procedures and incorporating security features into products. (page 1)

"The combination of the IBM 486SLC2 microprocessor, Micro Channel architecture, and high-performance subsystems provides excellent performance. (page 12)

"TrackPoint II is the first pointing device that outperforms a mouse for integrated typing and selection activities. (page 16)

"IBM's OS/2 2.0 is superior to Microsoft's Windows 3.1. To compete with IBM's OS/2, Microsoft has announced another system, Windows NT. (page 18)

"IBM's strategy for OS/2 Distributed Systems Management is a framework for systems management applications, and a set of applications that supports LAN resources. (page 37)

"The set of systems management capabilities providing remote, unattended installation of PC workstations is known as Configuration, Installation, and Distribution (CID). (page 43)

"Start/2 is an OS/2 2.0 application for planning networks, configuring and installing software products on networked workstations. (page 47)

"The Workplace Shell allows users to work the way they want to work. (page 69)

"There are three disadvantages to the polling protocol implemented in BIOS, and used by DOS and Windows. (page 78)

"Using COBOL to develop OS/2 PM applications has several advantages over other programming languages. (page 82)

G325-5020-00

Printed in U.S.A.