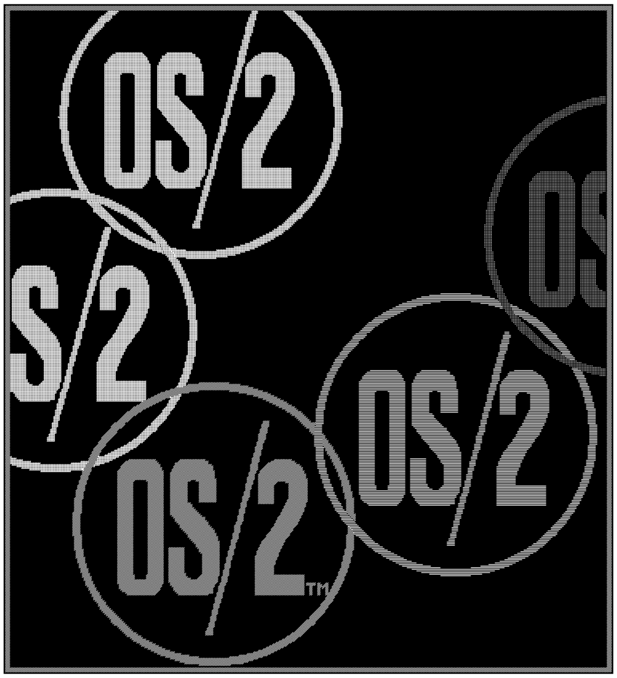
A Technical Guide to OS/2 2.0

January 1993

Martin McElroy

European Personal Systems Center (EPSC)
Basingstoke
UK
MCELROM at NHBVM7
+44 (0)256 343204

A Technical Guide to OS/2 2.0



A Technical Guide to OS/2 2.0

Notices

Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries:

OS/2
Presentation Manager
WIN-OS/2
AIX
DB2
Operating System/2
OS/400
PS/2
Systems Application Architecture
SAA Distributed Database Connection Services/2
System/370
Extended Services
NetView
Workplace Shell
IBM
DDCS/2
SQL/DS
RISC
RISC System/6000
SAA
SQL
Audio Visual Connection

All trademarks appearing in this document are owned by their respective companies.

Disclaimer

Some of the information in this paper concerns future products, or future releases of products currently commercially available. The discussion regarding Microsoft Windows is based upon information which the Microsoft Corporation has made publically available, and is subject to change. The description and discussion of IBM′s future products, performance, functions and availability are based upon IBM′s current intent and are subject to change.

Special notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM program, product or service is not intended to imply that only IBM′s program, product or service may be used. Any functionally equivalent program, product or service may be used instead.

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

Preface

This document describes the features and benefits of OS/2 Version 2.0 from a technical perspective. It is aimed at customer technical staff, software developers, IBM staff, dealers and business partners, and anyone else who needs to understand the detail of how OS/2 2.0 has been designed and implemented. Its primary aim is to help those evaluating the product to understand why OS/2 2.0 is the platform of choice for the 90s.

The structure of the document is designed to lead the reader from an awareness of the needs of the PC systems of the 90s, to an understanding of how OS/2 2.0 meets those needs. It therefore covers the design of OS/2 as well as its key features, and goes on to discuss future directions for OS/2. The guide can either be read sequentially, in its entirety, or individual sections may be used for reference purposes, relating to specific aspects of OS/2 (such as DOS compatibility). To help in the latter use, here is a summary of the aim and contents of each section:

OS/2 Version 2.0 sets the background against which OS/2 has been developed. It describes how the PC environment is radically different from that of the 80s, and the resulting needs of customers deploying PC systems. It also recounts how OS/2 has been designed to address those needs, culminating in release 2.0.

Why OS/2? outlines the most important reasons why OS/2 is the platform of choice for the 90s. It is a summary of the arguments provided in the rest of the document. It can be regarded as a management summary, and as such is designed so that it can also be used standalone, separate from the rest of the document. It is suitable for use in proposals, or customer reports on OS/2 2.0.

OS/2 2.0 key elements prefaces the three chapters on DOS, Windows and OS/2 support that follow, by describing some of the technologies that underpin the whole system. Features such as multi-tasking, hardware support, 32-bit design and Multiple Virtual DOS Machines (MVDM) are the foundation for many of the specific aspects described in the three sections that follow. They are also fundamental to understanding OS/2′s superiority to DOS-based systems.

Better DOS describes the extensive support for DOS applications, and the ways in which OS/2 2.0 provides a better environment for running DOS applications than DOS itself. This section includes discussion of the memory usage, protection and the steps taken to ensure the widest possible compatibility with existing DOS applications.

Better Windows begins by explaining some of the aspects of DOS/Windows 3.x, in order to show how OS/2 2.0 provides a better Windows environment. An important factor is the discussion of how DOS/Windows 3.x is used widely for running multiple DOS applications, perhaps even more than for running Windows applications. This leads on to a description of how OS/2 2.0 is a superior multi-DOS environment, which also runs Windows applications, supporting Windows features such as DDE, clipboard and OLE. The section concludes with a brief discussion of why many developers are already porting their Windows applications to OS/2, and some of the tools available to help.

Better OS/2 describes some of the additional features in OS/2 2.0 compared with OS/2 1.3. It illustrates the benefits of a 32-bit OS/2, and discusses migration from 16-bit OS/2 1.x.

Workplace Shell discusses the user interface of OS/2 2.0. It begins by outlining the reasons why OS/2 2.0 uses a new user interface model, and the benefits it provides. Emphasis is placed on the fact that users have greater flexibility to work the way they want, and concentrate on the information they are working with, not the steps the computer needs to follow. This section also describes some of the components and features of the Workplace Shell, comparing them with older GUI systems like DOS/Windows.

OS/2 in a connected environment points out how OS/2 2.0 is the client of choice in client-server systems, and is the base for a family of products which connect the OS/2 client into LAN, mini and host based systems, including open systems. Some of the OS/2 products that offer these features (Extended Services, OS/2 LAN Server, TCP/IP for OS/2) are briefly described. Another crucial element in considering connected, rather than standalone systems, is the ability to support systems management tools, so that the cost of maintaining and managing the system does not exceed the benefits. This section also discusses migrating from older connectivity products such as DOS-based terminal emulators and networking products.

Futures describes some of the areas in which OS/2 is expected to be enhanced to address the growing sophistication of PC usage, and emerging technologies such as distributed computing, multimedia and object-oriented technology. It aims to provide an understanding of the framework within which OS/2 will continue to develop throughout the 90s.

The ***Appendices*** give some background information on OS/2 2.0 and comparisons with other environments.

Appendix A compares OS/2 2.0 features with those of DOS/Windows 3.x, including a detailed comparison of support for DOS applications

Appendix B discusses the hardware requirements for OS/2 2.0 and gives a summary of performance considerations and tuning hints.

Appendix C gives a guide to other books, publications and materials relating to OS/2 2.0, where the reader may find more information.

To use the guide most effectively, it is important not only to understand its contents, but also its purpose. Although the document is technical in nature, it does not aim to be a technical reference. This is beyond the scope of a document of this length. Those looking for such details should consult the OS/2 Technical Compendium referenced in the Appendices (see Appendix C, “Further reference materials” on page 133). This guide stands in technical content and level of detail, between the product information brochure, and the comprehensive discussion of the OS/2 Technical Compendium.

The guide covers OS/2 function in the release made generally available at the end of March 1992. It gives some discussion of planned updates to be made available during 1992, but details may change before such updates are available. Readers are advised to check with IBM representatives or Authorised Dealers for specific dates and functions.

Although this guide is suitable as an introduction to OS/2 for software developers, it is *NOT* intended as a developers′ guide. Issues for programmers are dealt with only as they affect the fundamental OS/2 architecture or in as much as they concern the OS/2 user. More detailed information for OS/2 programmers may be obtained from your local IBM contacts or via the OS/2 Developer Assistance Program, or on local bulletin boards or Compuserve.

No guide to OS/2 can pretend that the product exists only in a standalone environment. The chapter ***OS/2 in a Connected Environment*** discusses how OS/2 2.0 is ideally suited to the needs of the connected PC, whether it be LAN-based, or integrated within an enterprise network. It also describes in brief some of the other members of the OS/2 family, such as Extended Services for OS/2, or OS/2 LAN Server. However, detailed discussion of these products would probably double the size of the document. A separate technical guide covering the OS/2 systems extensions is planned (contact your IBM representative for more information).

References to Microsoft Windows 3.x are made to denote the versions of Windows currently available (3.0 and 3.1) which act as an extension to DOS, running on a DOS base. The term, “Windows 3.x” is used throughout the guide where the comment is applicable to both 3.0 and 3.1. A specific version number is used when talking only about one or other release. References in this document to Windows/NT (which Microsoft has announced they expect to ship in 1993) are based on information which the Microsoft Corporation has made publicly available.

Contents

Figures

Tables

OS/2 Version 2.0

The changing PC environment

In 1981, when IBM introduced the first IBM Personal Computer, no-one anticipated how much it would change the face of the computer industry. In the first half of the 1980s, PC use was mainly confined to improving personal productivity by using spreadsheets, word processors and other widely available applications.

However, it has been clear from the mid-1980s and beyond that the PC needs to perform a broader role in the organisation. As greater amounts are spent on PC technology, more is being demanded, and increasingly the PC is being seen as a “window” on the enterprise, a single screen from which company-wide information resources can be accessed, and a base for mission critical, “line-of-business” applications that previously would only have run on a mainframe. Thus, PCs have been increasingly connected together in Local Area Networks (LANs) and to host computers, making them a critical part of the corporate data network.

In addition, new technologies like Graphical User Interfaces (GUIs) and multimedia, along with greater connectivity, offer the potential to provide even more information in an easily accessible manner, so that not only can more ***uses*** be found for PCs, but also more ***users***. In this way, PCs can be used to transform the business, not just increase productivity or automate existing processes.

And the trend towards “rightsizing” applications continues, moving critical applications to PCs on a LAN, using a client-server approach. The market analysts Forrester Research published a report in May 1992, identifying the growth in demand for a new “super client,” driven by the migration of business applications towards LANs and client-server.

Unfortunately, much of this remains an aspiration, rather than reality. An article in a June 1990 issue of *PC Week* said that the PC must now “grow up,” saying that PCs must be considered as “business tools,” not “microcomputers.”

But a number of requirements must be in place before the PC can “grow up” and before the “super client” can emerge:

- ***“Industrial Strength” reliability:*** unless the PC can demonstrate itself to be as robust an environment as the host computer, it cannot expect to take over some of the applications the host runs today, nor even participate fully in sharing those applications via a client-server or co-operative processing setup. There is little point in having a reliable server or host if the client platform is unstable.
- ***Networking “Built In”:*** the PC environment needs to allow straightforward and simultaneous connection to a variety of other platforms: LAN, mid-range, UNIX and mainframe, and to be able to handle the variety of communications protocols that results from the multi-vendor environment in most companies.
- ***Performance:*** the PC platform must demonstrate that it can adequately share the processing load, by exploiting the power of the base hardware. Since most client-server and connectivity applications will require multiple processes, the ability to run several concurrent tasks efficiently is fundamental. In this respect, the requirement for performance measurement goes beyond the simple benchmarking of one application at a time, and leads towards determining overall throughput and concurrency.
- ***Wide Application Choice:*** the platform of the 90s should build on compatibility with existing productivity applications, and allow “mission-critical” applications to be developed for the same platform, so that the user′s system can handle both business and productivity applications.

- ***Ease of use and low training costs:*** in expanding the use of the PC, additional complexity in terms of communications and multiple applications are inevitably introduced. These must be implemented while presenting the user with a way of working the system that is as easy to learn and adapt to as possible, both for the user′s sake (for better user adoption) and for the organisation′s (in less “down time” while learning the system). This applies not only to existing users: to expand the use of the PC and bring in more users, to make it a truly business system, barriers to learning the system must be lowered.
- ***Easily managed:*** put simply, the costs of installation, integration, maintenance and updates most not exceed the benefits of running the system.
- ***Investment protection:*** all of these aims must be achieved without completely starting from scratch. To retain users′ comfort and capitalise on the existing investment in applications and hardware, maximum use must be made of the systems and applications in place, where possible integrating them with the new systems. But the design of such a system should offer compatibility, but not be constrained by the past. Investment ***exploitation*** is as important as protection in the long term.

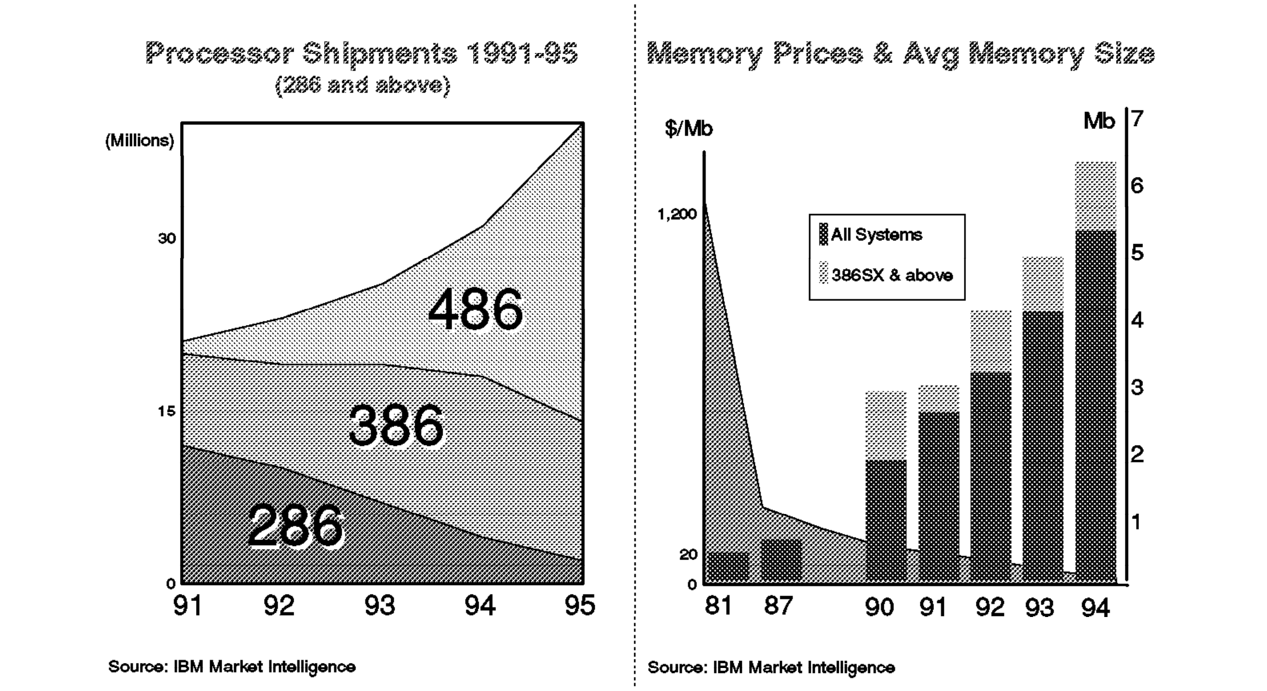
All in all, the requirements of the PC systems of the 90s can be summed up in one word: ***integration***. The means of integration will be software, and in particular the operating environment. Once the basis is there, applications to exploit it will follow. In short, an advanced operating system is needed for the PC of the 1990s, one that can exploit the benefits of the ten years′ worth of productivity applications, while moving the PC platform forward to address the needs of the 1990s and become a “business” machine, rather than just a productivity tool.

Cheaper processors and memory

Some of the obstacles to the widespread use of an advanced platform, such as high memory costs and insufficient processor capacity, are now being removed. Memory prices have fallen dramatically over the last three or four years, and the recent development of the 16 megabit DRAM chip means that memory is being packaged in ever larger units (4MB or in future 8MB at a time). This will continue the rise in the average amount of memory installed in PCs: in 1992, nearly all IBM PS/2s using an Intel i386SX processor or above, ship with at least 4MB of memory installed as standard, and nearly all other PC vendors are doing the same. Indeed, 8MB is no longer an unusual configuration, and many high end machines can be installed with 16 and even 32MB of memory.

Furthermore, processor power is increasing rapidly, and Intel′s i386 processor family (i386SX and i386DX) has become the largest volume Intel processor shipped in PCs. According to industry estimates, in 1991, the majority of new PC shipments had a 386SX or above, and by the mid-1990s this figure will increase to over 90% as the i486 ships in greater volume. In fact, recent changes in the competitive microprocessor market, have led to large reductions in prices on even 486 chips, particularly the 486SX, so that some vendors are even using the 486SX as the mainstream processor across their product range. Therefore, 32-bit processors, whether 386 or 486, completely dominate new shipments of PCs.

The result of this is that the new PCs shipped have a minimum of a 386SX and 4MB of memory, and commonly a 486SX and perhaps 8MB. This enables substantially more to be done with the PC platform and, in particular, an advanced operating system base that can really begin to fulfil the PC′s potential.



Source: IBM Market Intelligence

Source: IBM Market Intelligence

Figure 1. Processor shipments and decreasing memory prices

OS/2 so far

In the mid-1980s, IBM realised the need for an advanced platform, and saw that single-tasking DOS was not likely to be able to meet these demands. IBM therefore set out with Microsoft to develop OS/2. In 1987, the first character-based version, OS/2 1.0 appeared, followed by the inclusion of the Presentation Manager GUI in Version 1.1 (1988), and by Version 1.2 in 1989. These were all based on the Intel 80286 (286) processor′s protected mode, and were therefore 16-bit releases. Intel 386-based machines were supported, but their full 32-bit potential was not exploited because of the requirement to support the large base of 286 machines at the time.

These first releases provided the basis for a “platform of the 1990s,” but the learning curve involved in providing the operating system function itself and in developing applications to exploit it, proved a greater challenge than expected.

In 1990, IBM took over chief responsibility for the 16-bit versions of OS/2, and later that year, produced Version 1.3, which has become a widely used platform among companies who are beginning to develop in-house “line-of-business” applications, benefiting from its multi-tasking, large memory, industrial strength robustness and protection between processes. In addition, 1.3 provided these capabilities with lower memory requirements than previous releases, while adding extra features such as high quality font support via the built-in Adobe Type Manager.

Although OS/2 1.3 has been highly regarded as a platform for line-of-business applications, limits in the capability of the 286 processor set bounds on what could be achieved in terms of compatibility with the existing range of DOS productivity applications. This, in turn, placed limits on the level of integration between new applications and the existing, extensive installed base of DOS applications.

OS/2 2.0, which first shipped in March 1992, has been designed to build on the strengths of OS/2 1.3, such as multi-tasking and threading, robust protection between applications, and large memory support. It adds to those facilities, greater compatibility with the existing installed base of DOS and Windows applications, so that users can choose from the widest range of applications on an

Intel-based platform. Because it is designed for 32-bit processors like the 386 and 486, it can escape the limitations of the 286, and provide not only a better platform for supporting old applications, but a new foundation for 32-bit applications which can fully exploit the capabilities of the 32-bit hardware. That is why OS/2 2.0 is the ***Integrating Platform***, bringing together the investments of the past, and providing a base for the future.

Why OS/2?

This section summarises the reasons why OS/2 is the platform of choice for the 1990s:

The best of both worlds

In the PC environment of the 90s, where both personal productivity and line-of-business applications are required, only OS/2 can satisfy both needs. It provides a better platform for DOS applications than DOS itself, and runs the widest range of DOS and Windows applications, as well as being the best platform for running in-house mission critical applications, with its industrial strength, robust protection, and powerful multi-tasking. You don′t have to choose between different systems for your different needs - OS/2 can do both.

Broad appeal

OS/2 2.0 is, therefore, a platform of broad appeal, not just for high end usage, or for niche applications like servers, but as a client system, and a productivity machine. You do not need to be a "power user" to appreciate the benefit of running several DOS and Windows applications on a reliable base. Not only the features of the product, and the applications it can run, but its price and the breadth of PC systems it can run on, have been planned to address the widest possible audience of PC users. And this has already been proved to be true. On August 12th, 1992, IBM announced that it had shipped one million copies of OS/2 2.0 since its initial shipment in March. This represents, according to data from independent estimates, an initial success comparable to that claimed for DOS/Windows 3.0 in 1990.

During the first ten years of the PC, users, IS staff, and developers found it difficult to arrive at a common platform: DOS and Windows satisfied the users′s needs for productivity applications, but lacked the reliability and full connectivity support to be trusted in mission-critical environments by IS staff and developers, whose systems remained mostly on the mini and mainframe. However, OS/2 2.0 is a platform that can appeal to all three communities; the right choice for the IS strategy, can now also be the platform that developers and users can choose for themselves.

Freedom of choice

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

OS/2 2.0 can run DOS, Windows and OS/2 16-bit applications; it provides the widest choice of applications on an Intel-based platform. More and more 32-bit OS/2 applications are appearing, making the choice even greater. In fact, OS/2 2.0 is such a good environment for DOS and Windows applications, that even if you only use DOS applications on a 386-based machine, OS/2 2.0 is the best environment to run them in.

Furthermore, all applications running under OS/2 2.0, whether DOS, Windows or OS/2 applications, gain added value from working together; they can share information and be run from the common Workplace Shell desktop. This not only protects your current investment in DOS, Windows and OS/2 applications, but adds value to them by integrating them together.

OS/2 2.0, and Extended Services and OS/2 LAN Server, are supported on a wide range of IBM-compatible hardware as well as IBM PS/2s. This means you can run OS/2 with confidence on hundreds of machines from vendors including Compaq, Olivetti, Dell, Hewlett Packard and Toshiba, and with IBM support. In fact, even though IBM cannot test OS/2 on all the models and manufacturers in the market, it is likely that most PCs equipped with an Intel 386SX or above processor, will work.

A productive environment for the user

OS/2 provides an object-oriented user interface, the Workplace Shell, which allows users to think of the information they want to work with, not think first of what application needs to be loaded. This is a business-oriented, rather than computer-oriented way of working. In this way, users become more productive. They can concentrate more on what they want to do, and less on how to do it. The Workplace Shell also provides a single, consistent environment in which multiple applications can be loaded from different sources. It is an easy environment to learn, since once you know how to drag a file′s icon with the mouse to put it into a folder, you can use the same operation to print it, and to copy it to another disk or folder. In addition, companies can benefit from a standard interface, which 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 multi-tasking means that long-running processes can simply be switched to the background, while the user continues with something else. This results 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 via consistent interfaces like the clipboard.

Better DOS, Windows and OS/2

OS/2 2.0 doesn′t just run the widest range of applications on an Intel platform. It also aims to improve even on the native environment of each application, and provide a better environment for that application to work in:

A better DOS: it adds to the DOS environment: multi-tasking, more memory and application integration (see “Better DOS” on page 33).

A better Windows: it provides a superior environment for running DOS applications than either Windows 3.0 or Windows 3.1, and also runs a wide range of Windows applications with no loss of function, as well as taking advantage of the better multi-tasking, memory support, and reliability of OS/2 (see “Better Windows” on page 47).

A better OS/2: it improves even on previous releases of OS/2 itself, providing a new object-oriented user interface, a graphical install program, and access to powerful 32-bit applications (see “Better OS/2” on page 69).

A platform you can rely on

When the PC becomes the focal point of information processing, as in today′s environment it often is, then the PC platform must show the stability and reliability of the host environment. The main reason why the PC has not been trusted to fulfil this crucial role to date, is because its operating system - DOS - was not designed for “mission critical” use, but as a personal productivity environment. Nor can extensions to DOS such as Microsoft Windows, offer the required stability while they continue to be based on DOS. Only OS/2, which has been *designed* to protect applications from one another, can deliver the stable platform required for full multi-tasking and greater protection from system crashes. It is no use having the most fault tolerant server or host, if the client keeps going down. And even the productivity user′s PC is “mission critical” from that user′s perspective, so that reliability is a requirement for every PC.

Superior connectivity

OS/2 2.0 is both the server and the client platform of choice. Its strong multi-tasking and robust protection make it the best available base for working in a connected environment, in client-server and distributed processing. It provides a consistent platform for both server and client, can handle multiple concurrent communications protocols (eg NETBIOS, APPC, IPX, TCP/IP) with ease, and even provides a LAN-independent user interface to mixed vendor networks. It is enabled for automated LAN-based installation. But 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 which depend on communications with various systems can be implemented much more safely on OS/2 than on DOS or any of its extensions.

OS/2 2.0 is also the base system for a family of networking and communications products from IBM, including Extended Services for OS/2 and OS/2 LAN Server.

Extended Services for OS/2 provides powerful communications and database function in a single integrated package. It supports multiple communications protocols (EHLLAPI, NETBIOS, APPC, X.25 and asynchronous) and terminal emulators (3270, 5250, ASCII, VT100). It includes a full function client-server relational database management system (RDBMS). This RDBMS is part of an SAA family of relational databases which also includes DB2 and SQL/DS. Extended Services for OS/2 also includes query and database management tools.

OS/2 LAN Server works with OS/2 to provide Local Area Network support to DOS and OS/2 machines. It includes facilities for sharing resources such as files and printers, and to manage those shared resources, providing security control, applications management, and network statistics.

Both Extended Services for OS/2 and OS/2 LAN Server 2.0 run on both OS/2 2.0 32-bit and the 16-bit OS/2 1.3.1 bases. OS/2 LAN Server 3.0 runs only on the OS/2 2.0 base. Both Extended Services for OS/2 and OS/2 LAN Server are also supported on selected non-IBM as well as IBM equipment. More details on these and other networking products are available in the chapter on OS/2 in a connected environment (see “OS/2 in a connected environment” on page 97).

The integrated system

OS/2 not only makes DOS, Windows and OS/2 applications run together, but also provides a GUI, and, with Extended Services for OS/2 and LAN Server, database, communications, and LAN support. For developers, all the Application Programming Interfaces (APIs) and services have been designed to work together. This means customers don′t have to do the systems integration work with a variety of DOS-based packages to include all the function required, and work around any incompatibilities or problems. Instead, the OS/2 function has been designed and tested to work together - IBM has already done the integration work.

OS/2 also has advanced features built in and integrated with the rest of the system: the Presentation Manager GUI itself, font handling with Adobe Type Manager (ATM) (now for both OS/2 applications and Windows applications), REXX (the extended batch language which can be used to integrate different applications and automate common routines)

The Workplace Shell environment integrates DOS, Windows and OS/2 applications and makes them work together, even though they may have been written by different vendors and never designed to do so. Also, OS/2 can permit working combinations (such as a 32-bit OS/2 word processor and a 16-bit DOS spreadsheet) that would never have been possible before. In summary, the whole is greater than the sum of the parts: in running not only OS/2, but DOS and Windows applications, OS/2 makes them work together from the same user interface. This helps reduce the differences between applications, and between the local machine and the network, and make it all act as one system. That′s why OS/2 is the ***integrating*** platform.

32-bit power

OS/2 2.0 is the first mainstream 32-bit platform for Intel-based PCs. It offers the ability to take full advantage of the performance of today′s 32-bit PCs. And over over 1000 32-bit OS/2 applications are being developed (several hundred of which had already shipped by September 1992), to demonstrate what can be achieved with a 32-bit system.

The OS/2 32-bit API also allows developers to create richer, more sophisticated applications. It overcomes the constraints imposed by existing 16-bit systems (see "Why 32-bit OS/2?" on page 71). This allows applications like multimedia to exploit their full potential and power. OS/2 provides this foundation ***today***. Moving to the OS/2 32-bit API now, gets developers ready for future developments in OS/2, such as object-oriented technology, distributed computing and portability to RISC. So 32-bit is not just about exploiting the power of today′s environments, but also to move forward to build for the future. But most of all, OS/2 2.0 gives you the benefits of a 32-bit system NOW - no need to wait for other alternatives with uncertain delivery dates.

Platform for growth

The design of OS/2 not only preserves the investments of the past, but allows maximum flexibility for future growth. It also makes the OS/2 system itself ready to develop further (for portability to other processors, for example).

OS/2 will be the base of new developments for many of the features that will be a requirement for the workstation of the mid-90′s, such as multimedia, object-oriented systems, support for the Distributed Computing Environment (DCE) and portability across different processors. It has the extra power to support such advanced features as they emerge. Requirements of this kind will demand a robust, architected and powerful 32-bit system, and that system is OS/2.

With OS/2 2.0, IBM has reaffirmed its commitment to OS/2, and its conviction that the workstation of the 1990s requires a real advanced platform, not a series of extensions to DOS, which is fundamentally ill-equipped for these requirements.

Value for money

OS/2 2.0 offers a "3 in 1" environment, with everything you need to run DOS, Windows and OS/2 applications in the one package. It also includes a series of productivity applications, utilities and games for which you need to pay extra in the DOS world. OS/2 provides scalable font support for both Windows and OS/2 applications with Adobe Type Manager at no extra charge. OS/2 2.0 offers all this function at a price less than the combined cost of DOS and Windows 3.1, even before taking into account the extra utilities you would need to buy under DOS or Windows to achieve the same function. Upgrading from DOS or Windows makes the cost of moving to OS/2 even less.

Exploits today′s investment, and is a base for the future

OS/2 supports the widest choice of applications from the ***past*** ten years of the PC, and is the best platform for the ***present*** requirement of client-server and reliable connectivity. It also provides the best base for ***future*** technologies. That′s how OS/2 2.0 can integrate past, present and future requirements. It already has ***NOW*** what other environments can only promise for the future, with the best and most reliable migration path - why wait?

OS/2 2.0 key elements

This section examines some of the key technologies and features of OS/2 2.0, which affect all applications running in the system. The following sections cover OS/2 from an application point of view, showing how it exceeds the capabilities of DOS, Windows 3.x and previous releases of OS/2.

32-bit

OS/2 2.0 is the first version of OS/2 to support a 32-bit addressing system and programming model. Although implemented on the Intel 386 and 486 family of processors, it is really a 32-bit system rather than an Intel-specific system. Mike Kogan, one of the lead designers of 32-bit OS/2 has said in his book *The Design of OS/2 2.0* (see Appendix C, "Further reference materials" on page 133), "OS/2 2.0 was not designed to be 386-specific, but rather 32-bit OS/2 implemented on the 80386 platform." Part of the design of OS/2 2.0 has been to leave the maximum possible scope for future portability, both in the API and the subsystems like Presentation Manager (PM).

OS/2 2.0′s 32-bit design provides significant benefits to the user and the programmer, including better performance, simpler programming and ease of migration from 16-bit applications, as well as reinforcing the benefits of inter-process protection associated with previous releases of OS/2.

Flat memory model

OS/2 2.0 features a different memory addressing model from 1.3 - the flat memory model. This enables each process to look at memory as a large linear address space which can be addressed by a simple 32-bit offset (sometimes referred to as "0:32"), as opposed to the segment/offset combination ("16:16") required by OS/2 1.3 and other 16-bit systems like DOS and Windows. The 0:32 model therefore hides all details of segmented memory management from the 32-bit programmer, resulting in :

- much simpler programming
- better performing code
- greater portability to other instruction sets (ie non-Intel).

One example of better performance is in calling Dynamic Link Libraries (DLLs), which are common in both the OS/2 system and OS/2 applications. Since all code and data are addressable within the same linear address space, there is no longer any need for segment switching.

Large memory address space

Using the 32-bit flat memory model, the 386 processor supports up to a 4 gigabytes (GB) linear address space. In fact, up to 64 terabytes can be addressed by the processor using a non-linear addressing model. To get some idea of the enormity of such numbers, this means that, if an average page in a book contained 2000 characters, and each book 500 pages, a task could address any character in any book in a library with more than 70 million books (70,368,744,177,664 characters to be precise)!

For reasons of compatibility with OS/2 1.x 16-bit applications, the address space per process in OS/2 2.0 is restricted to 512MB (since this is the maximum address space possible for 16:16 applications). This is addressed using the flat memory model, allowing programmers to reference memory under OS/2 2.0 as one single huge address space. Although 512MB is hardly likely to be a limitation in practice, this memory limit will be removed in a future release of OS/2.

Within a given process, memory objects can be allocated for any size, no longer limited to the 64KB segment maximum, but any size between 1 byte and 512MB. This gives the programmer much greater flexibility in memory management. For the user, this means that programmers can spend more time on making their programs more powerful and even easier to use, and less on segmented memory management.

Virtual memory

OS/2 2.0 can use more memory than is physically installed in most PCs, by using the hard disk for additional memory, usually referred to as virtual memory. Although 16-bit systems like OS/2 1.3 and Windows 3.x also offer memory overcommit via swapping segments to disk, the larger process address space available under OS/2 2.0 means that virtual memory is effectively limited only to available disk space, whereas Windows 3.x can offer only 16MB maximum, or four times the physical memory installed in the machine, whichever is less. OS/2 2.0 takes advantage of the 386 processor′s support for fast and effective paging of memory to and from disk (see "4KB demand paging" on page 15). It is also not limited to the segment-based swapping of OS/2 1.3.

Mixed 16-/32-bit environment

Not only are OS/2 1.x applications supported, and are binary-compatible with 2.0, but OS/2 2.0 can support mixed model programming. In order to support applications written for 16-bit, the OS/2 designers had to develop an architecture in which 16- and 32-bit modules could reside simultaneously. This was not only for ease of conversion from 16-bit programs, but also because the system itself contains a mixture of 16- and 32-bit service routines. This is a difficult task because segmented and flat memory models are so different.

OS/2 provides address conversion between 16:16 and 0:32 addresses, to allow 16-bit API calls to be serviced internally by 32-bit routines, and vice versa. A series of procedures called "thunks" within the system facilitate this process, by handling any necessary parameter conversion for APIs. This technique is used internally between 16- and 32-bit parts of the system itself, but the user never notices - the system takes care of it. They are tools internal to the system, and not APIs for the programmer to learn and use - and certainly no user needs to know how to handle them.

Note that the mixed model gives great flexibility, both in migrating applications from 16- to 32-bit, and also in allowing 32-bit applications to make the best possible use of existing 16-bit service routines, window classes etc., developed for previous releases of OS/2. The important point here is how OS/2 2.0 helps the migration process to 32-bit.

Intel 386/486 exploitation

Since OS/2 2.0′s 32-bit capabilities are implemented on Intel′s i386 and i486[1] family of processors, it is not surprising that the system takes advantage of some of the processor′s features. Indeed, some of them (especially virtual 8086 mode) are vital to some of the most notable features of the system. It is for that reason that OS/2 2.0 requires a 386SX processor or higher to run. Unlike OS/2 1.3, it will not run on machines equipped with an 80286, nor will it run on 8088 and 8086 machines supported by DOS.

[1] In this document, the i386 and i486 processors will often be referred to as 386 and 486 respectively. Where discussion relates to 386 or 486 SX or DX models specifically, terms such as 386SX will be used. Otherwise the terms will be used to refer to any member of the 386 or 486 family as appropriate. The term 80X86 is sometimes used to denote the range of 32 bit processors currently shipped by Intel (ie 386 and 486 together, but NOT the 80286).

Growth in processor power

It is widely recognised how substantial have been the improvements in processor speed and capacity, even since 1981 and the launch of the first IBM PC. Since then, processor clock speed has increased by a factor of seven, and memory address space by four thousand to one. This growth in processor power lowers the cost of the hardware technology to run advanced operating systems and applications. The software and hardware are now at a sufficiently advanced stage, to enter a new phase of personal computing, beyond the basic productivity applications that have characterised its usage to date (see “Cheaper processors and memory” on page 2).

386/486 now majority of shipments

PCs using the i386 family (which includes the SX as well as DX models) are now shipping in greater volumes than any other Intel processor. Add to the number of 386s the growing volumes in 486SX and DX processors, and there are a growing number of 32-bit processors in the marketplace. Until OS/2 2.0, there has not been a 32-bit system for Intel processors which has provided sufficient compatibility with existing 16-bit applications, to become a standard.

386SX versus DX

It is important to understand that both 386SX and 386DX processors are supported by OS/2. Even though the 386SX has a 16-bit external data bus, it is a 32-bit processor internally, fully compatible with the 386DX processor′s instruction set, supporting multiple Virtual 8086 sessions (essential for the Multiple Virtual DOS Machines features of OS/2 2.0), protected mode operation and full 32-bit addressing. It can therefore support all of the features that OS/2 2.0 provides, just as the 386DX does. The main difference is in the throughput for I/O, owing to the 16-bit data bus. Nevertheless, many users find it an excellent combination of 386 function, compatibility and performance at a reasonable price. IBM′s own 386SLC chip, which appears in, among others, some models of the PS/2 Model 57, adds performance features such as caching to the basic 386SX design; it represents an excellent platform for OS/2 2.0.

486SX and DX

All the 386 features used by OS/2 2.0 are also supported by the 486. Therefore machines using a 486SX or 486DX will also run OS/2 2.0. In many operations, the 486 is faster than the 386, which will provide performance benefits to users of OS/2 2.0 on 486-based machines. The 486SX is very similar to the 486DX, but does not have the built-in math co-processor function which the 486DX has. The price of the 486SX processor has fallen rapidly during 1992, so that machines using 486SX can often be obtained at prices competitive with 386DX machines. The 486SX therefore represents an economical entry into 486 power for OS/2 2.0.

Features of 386 chip used by OS/2 2.0

Among the specific features of the 386/486 processors supported by OS/2 2.0, and used to provide additional function beyond OS/2 1.3, are:

Protected mode operation
Virtual 8086 mode
4KB demand paging
Numeric co-processor support

Protected mode operation

To understand the significance of protected mode, let us briefly review some of the different models of memory management which the 386 processor can use:

Table 1. Memory management models

Model	Processor architecture	Max physical address space	Addressing style (segmented vs. flat)	Protection between processes	System software example
Real mode	8086	1MB	segmented (64KB)	none	DOS
Segmented Memory Model	80286	(on 286) 16MB	segmented (64KB)	yes	OS/2 1.3, Windows 3.x
Flat Memory Model	i386	4GB (16MB on 386SX)	flat	yes	OS/2 2.0

In reality, operating systems like OS/2 create an architecture with indirect selectors/descriptors rather than direct segment/offset addressing. This provides memory protection between processes to support multi-tasking. This is just one example of the ways in which an operating system builds on the base capabilities of the processor. Virtual memory is another function provided by the operating system but relying on a processor feature (in this case on the 286 and above).

In OS/2, all process address spaces are kept separate from each other. This allows OS/2 to maintain protection between different processes running in the system, whether they be DOS, Windows or OS/2 applications, or other parts of the system itself. The result is that, if a process attempts to reference memory outside its own address space, it is trapped by the system. In contrast, multi-tasking systems based on single-tasking DOS, such as Windows, cannot always offer protection. Windows applications, for example, share the same Local Descriptor Table as Windows itself, which limits the memory protection Windows can provide to its applications (see “Reliability” on page 54 and “Reliability and protection” on page 39).

Virtual 8086 mode

The 386 offers a mode called virtual 8086 (V8086) mode, which emulates multiple instances of an Intel 8086 processor to provide some compatibility with real mode applications (such as most DOS applications) under a protected mode system (like OS/2 2.0). Applications running in V8086 mode, can run concurrently with other 8086 applications and protected mode applications, and take advantage of the virtual memory and paging facilities of the 386. V8086 mode is a superset of protected mode.

The CPU provides high performance hardware support to enable switching between V8086 and protected mode - this eliminates the performance overhead of mode switching associated with lower processors such as the 286. This is why multiple DOS capability can be provided so much more easily and more effectively in OS/2 2.0, which requires the 386 facilities, than in OS/2 1.3 which was based on a 286-style architecture.

V8086 processes are protected from each other in OS/2 2.0. This gives compatibility with the real mode world of DOS applications while providing greater address space and protection. In OS/2 2.0, each Virtual DOS Machine (VDM) is encapsulated in its own unique linear address space, and thus cannot corrupt another application′s code or data. Traps or exceptions are handled by the VDM Manager in OS/2 2.0, and execution is passed to an exception handler, or the VDM terminated. Therefore, OS/2 2.0 runs in protected mode all time, even for DOS applications, hence the greater protection of the system against application errors or failures.

In real mode, an application can directly address any object in memory between 0 and 1MB, including portions of the operating system (whether this be DOS, DOS/Windows, or OS/2 1.3, all of which provide varying amounts of real mode execution). A DOS program that accidentally wrote to a system area, or directly addressed a hardware device and left it in an unknown state, could cause system integrity problems. Real mode is less of an issue in a single-tasking system like DOS, where it is more than likely that the application is the only one executing in the system; but in a multi-tasking system, it represents a “trap door,” which can endanger system integrity. This is part of the difficulty in running a multi-DOS environment like Windows 3.0 or 3.1 on a DOS base; as long as there is real mode access (as long as it runs on DOS as we know it today), there is a potential risk. In fact, even Windows 3.1 , which is claimed to offer greater protection and stability, does not change the design of Windows in this respect. It is still possible for a DOS Terminate-and-Stay-Resident (TSR) program to switch the system into real mode and open up the potential for endangering system integrity (see “Reliability” on page 54 for more on this).

The only solution is to run the operating system in protected mode, and provide a multi-DOS mode using features like V8086, so that the system never runs in real mode. This is exactly what OS/2 2.0 does, and it is one of the key differences between it and Windows 3.x in terms of overall integrity: Windows implements V8086 function in a system running on DOS, and some real mode access is inevitable - hence its occasional fragility when working with DOS applications or TSRs; OS/2 2.0 never executes in real mode.

4KB demand paging

Demand paging is the 386′s method of providing virtual memory to the system; when physical memory is exhausted (memory “overcommitment”), the disk may be used to provide additional virtual memory. Memory overcommit is provided on 286-based operating systems such as OS/2 1.3 as well, but is usually based on a segment-swapping mechanism. As the name suggests, segment swapping is closely tied to the segmented model used in OS/2 1.3, and the swapper algorithm has to do much work to compact segments of varying sizes into a unit capable of being swapped to disk, or much I/O work in swapping large segments.

OS/2 2.0 manages memory internally using pages, each of which is 4KB in size. Each memory object handled by the system is regarded as a set of one or more pages, and therefore memory is allocated in units of 4KB (although to optimise memory management, programmers may handle a page as multiple smaller objects). Paging offers a number of advantages over swapping, which was the mechanism used in OS/2 1.3:

Better granularity: when memory becomes over-committed (ie there is no more real memory left to load applications), individual 4KB pages may be swapped to and from disk, rather than entire memory objects (or segments as in OS/2 1.x). In turn, this will improve performance, especially in terms of the lower I/O cost of moving 4KB pages as opposed to whole segments.

Programmers can write their applications to take advantage of the smaller granularity of memory object (ie 4KB page as opposed to a whole segment). This will reduce the working set (overall memory usage) of applications and thence improve performance. Memory objects can be allocated in logical units, without the artificial constraints of segment sizes; this means that you can allocate smaller or larger memory objects according to the needs of the application, rather than the constraints of the system. Once again, this makes programming much simpler.

It is important to remember that paging is carried out without any awareness on the part of the application. Programmers may allocate memory to co-operate with this mechanism for performance reasons, but the user need not be aware of the paging system at all. It will simply translate into the performance benefits mentioned above.

Simpler swap algorithm: because of the more granular paging, OS/2 2.0 does not need to move around different sized segments to compact them into a single swap segment of the right size, as was done in

1.x. This makes the swap algorithm simpler and therefore faster, improving overall system performance. Indeed, unlike OS/2 1.x, the swapper file, SWAPPER.DAT is designed to actually shrink as well as grow during use of OS/2 2.0. This is in order that applications requiring high availability (such as LAN servers) do not need to be rebooted to recover swap disk space.

The 386 has hardware support, such as buffering and caching, to support paging. Once again, OS/2 is taking advantage of the processor′s built-in features.

Numeric co-processor support

Some OS/2 applications can make use of an i387 numeric co-processor if one is installed in a 386 PC running OS/2 2.0 (or 486DX systems, which have co-processor function built in). Where a 387 is not installed, OS/2 2.0 provides a 387 emulator to all applications running in protected mode, available to both 32-bit and 16-bit applications, but not VDMs running DOS applications. DOS applications must continue to use whatever mechanisms they are currently using to detect co-processor presence.

Multiple Virtual DOS Machines (MVDM)

OS/2 2.0 features a totally redesigned environment for DOS compatibility. It is based on the Virtual 8086 mode of the 386 processor, and results in a DOS environment where multiple DOS applications can be run, each in their own separate “virtual machine.” “Virtual machine” means that the virtual 8086 mode process simulates a self-contained DOS environment, in which the application runs. Access to I/O resources is usually not done directly to the physical device, but via a virtual device; this virtualisation allows the DOS application to believe it owns all the system resources, just as it does under DOS. Behind the scenes, the OS/2 system manages concurrent access to these physical resources, from both DOS and OS/2 applications.

Each Virtual DOS Machine, or VDM, emulates an entirely independent instance of DOS. It is in fact a generic emulated version of DOS (which resembles DOS 5.0), not the real DOS retail product. Each VDM is a separate process, protected from the others, and multi-tasking alongside the others. This design allows OS/2 to provide superior support for DOS applications than in previous releases of OS/2.

Contrast with OS/2 1.3 DOS support

It is useful to look at the OS/2 2.0 DOS support in the light of 1.3′s DOS support. The following table gives a summary of the key difference in DOS support between 1.3 and 2.0:

Table 2. DOS environments - OS/2 1.3 and 2.0 compared

	OS/2 1.3	**OS/2 2.0**
Processor mode	Real mode	V8086 mode
Protection/Integrity	Low	High
Number of DOS applications	1	up to 12 full screen; up to 240 windowed
Background execution	No	Yes
Windowed	No	Yes
Cut and Paste	No	Yes
Conventional memory free	512KB	633KB
DOS Extended Memory	None	16MB per application (XMS)
EMS/XMS	No	Yes
DPMI	No	Yes
Overcommit	Swap	Page - Available disk
Timing Dependent applications	Foreground	Foreground / Background
Recover from hang/crash	Sometimes	Usually

Let′s take a look at some of these aspects in more detail:

286-based architecture

OS/2 1.3 On the 286, there is no V8086 mode, so 1.x had to run DOS applications in real mode, thus requiring a protect-to-real mode switch when moving from an OS/2 to DOS application. This involves significant performance degradation (see below).

OS/2 2.0 DOS applications run in V8086 mode. No protect-to-real mode switching is needed.

Relative performance

OS/2 1.3 A switch from real mode to protected can be done on a 286 (though it costs many CPU cycles). A switch the other way, from protected to real, means effectively rebooting the CPU, but OS/2 has to preserve all the system control information, process states etc, at the same time, so that the normal processing can continue after the switch. This switching back and forth from real (DOS box) to protected (rest of 1.x) generates significant overhead. This can mean that OS/2 background tasks are slowed significantly when running a DOS application in the foreground.

OS/2 2.0 DOS applications run in V8086 mode. No protect-to-real mode switching is needed.

System integrity

OS/2 1.3 Since 1.3 does not run always in protected mode (unlike 2.0), any switch into real mode allows an errant DOS application to directly address any object in memory under 1MB, including portions of the OS/2 1.x kernel (the same is true in Windows 3, hence its comparative lack of integrity.)

Also, DOS applications could directly address a hardware device and leave it in an unknown state, which may cause OS/2 device drivers to fail.

Either of these scenarios could cause the whole system to crash.

OS/2 2.0 All OS/2 2.0 processes, including DOS applications in VDMs, are protected from each other, and cannot write outside their own address space. Hardware access is controlled through Virtual Device Drivers (see "Virtual Device Drivers (VDDs)" on page 21), which handle the DOS-OS/2 hardware concurrency problem.

Background execution

OS/2 1.3 DOS applications are suspended when not in the foreground. This was a decision made in 1.x, mainly to avoid the overhead in real-protected-real mode switching; it means that timing dependent applications like communications programs are unsuitable for running in an OS/2 1.3 DOS box, as they cannot receive interrupts while suspended.

OS/2 2.0 In 2.0, the virtualisation of interrupts and hardware access allows most communications programs to run in background in a VDM.

Amount of memory

OS/2 1.3 Since DOS applications run in real mode, some of the OS/2 device driver and kernel code has to be located below 1MB to service the real-mode (DOS box) requests. This has an inevitable result on the amount of memory available below 1MB, and therefore on DOS application space, which, in 1.3, depending on the configuration, is about 520KB. This prevents some larger DOS applications from even loading. No EMS or XMS support is provided in 1.3.

OS/2 2.0 OS/2 2.0 locates most driver and even DOS emulation code outside the DOS application′s address space in a VDM. This provides even more conventional memory than under standalone DOS, and full support for EMS and XMS (see below).

Memory: Conventional, Expanded, Extended

Memory management under DOS today is quite complex, and often requires the user to know about various different memory types. To support some DOS applications, OS/2 2.0 needs to provide more than just conventional memory.

Definitions

The diagram below shows some of the types of memory commonly referred to, and is a useful reference for the discussion below:

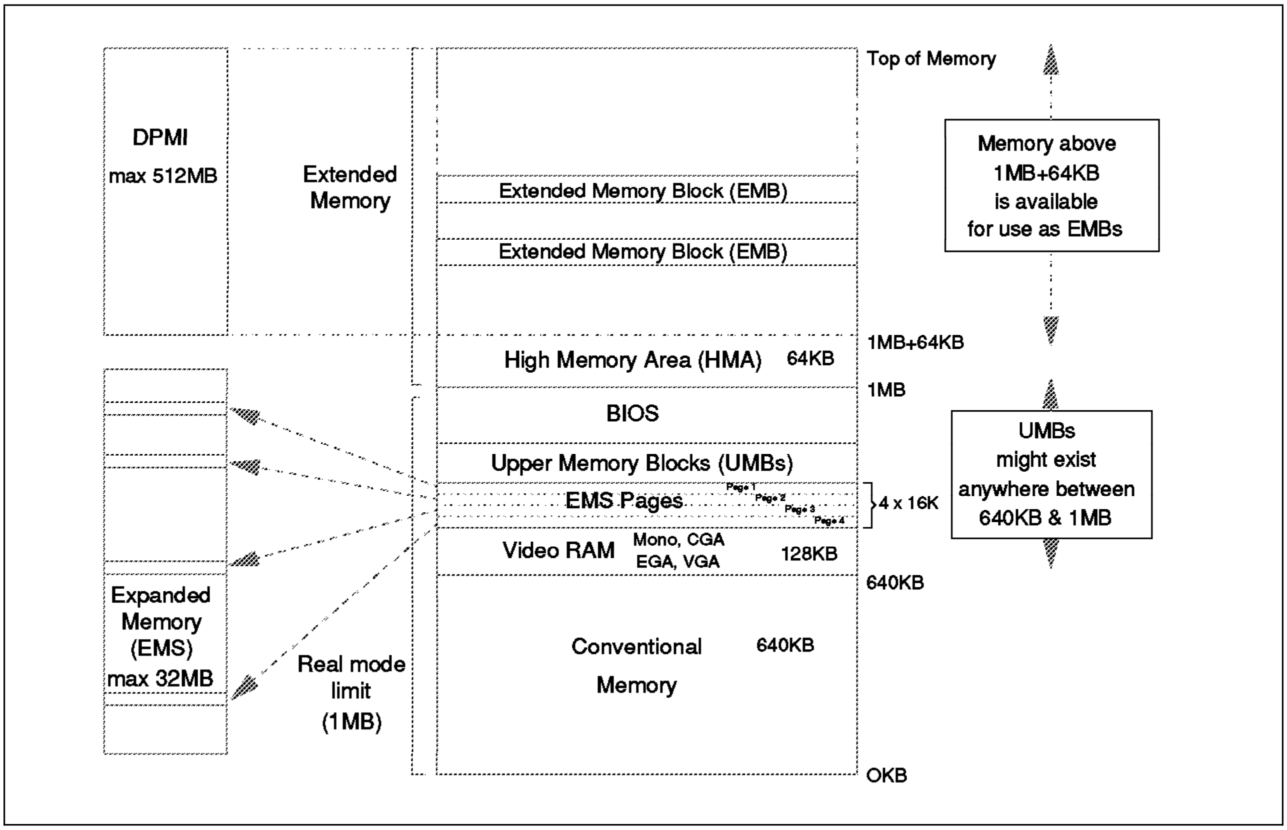


Figure 2. Memory: Conventional, Expanded, Extended

Conventional memory is the name given to the memory area up to 640KB accessible by DOS applications.

Extended Memory refers to any memory above the 1MB line addressed by the processor in protected mode. (1MB is the normal limit addressable by the processor in real mode). The LIMA (Lotus/Intel/Microsoft/AST) Extended Memory Specification (XMS) version 2.0 provides a standard for the use of extended memory on 80286 and above computers. The specification provides for moving code and data objects to and from extended memory to base (conventional) memory, and is operating system independent (even though the technique for determining that XMS is present relies on the DOS interrupt vector 2Fh). XMS manages 3 different kinds of memory, described below:

- High Memory Area (HMA)
- Extended Memory Blocks (EMBs)
- Upper Memory Blocks (UMBs)

High Memory Area (HMA) is the region of memory between 1MB and the 64KB above it (minus 16 bytes). This can be addressed by enabling one of the processor′s address lines to allow the processor to access an extra 64KB beyond the normal 1MB limit in real mode. Its operation is due to an anomaly in the 286 and above, which has been exploited by this technique to give a valuable extra address space. Drivers like HIMEM.SYS, which appear in DOS 5.0 and MS-Windows 3.x, exploit this technique.

Extended Memory Blocks (EMBs) are blocks of extended memory above the HMA, not accessible from real mode and serve usually as data storage. An XMS driver can move memory between extended and conventional memory to offer up to 64MB of extended memory in up to 255 blocks.

UMA, UMBs: between 640KB and 1MB (the limit addressable by the processor in real mode) lies the Upper Memory Area (UMA). In this area are reserved portions of memory for BIOS, video buffers etc. Usually, however, there are a number of address ranges not used (for example, since there are address ranges for monochrome, CGA and EGA adapters, it is not possible to use all of these ranges at the same time. Thus there will be a number of “gaps” in the UMA. These can be used by memory managers such as exist in DOS 5.0 itself, or by third party utilities such as Quarterdeck′s QEMM product. They can be used to load various drivers, or in DOS 5.0′s case, even part of the system itself, to free more memory below the 640KB line. The address ranges used in this manner are usually called Upper Memory Blocks or UMBs. The number and size of UMBs will depend on the hardware configuration.

Expanded Memory (EMS) is a page mapping technique that provides additional memory support, by allowing DOS applications to allocate and access up to 32MB of additional memory. This is done by creating memory objects in expanded memory that can be mapped into the real mode 1MB address space, thus allowing DOS applications to access address spaces beyond 640KB at the cost of having to quickly remap the memory that is to be accessed. In effect, parts of the 8086 address space become moving “windows” into larger virtual memory objects in an expanded memory area. The expanded memory specification was developed by Lotus, Intel and Microsoft, and is thus known as LIM EMS. The latest version of the specification is 4.0. EMS is provided under DOS either by a special memory adapter and driver, or by defining portions of memory above 1MB for use as expanded memory by a driver that sometimes ships with DOS (eg EMM386.EXE with DOS 5.0).

DOS Protect Mode Interface (DPMI): The DPMI specification provides a standard interface that can access memory above 1MB and is addressable by computers with an Intel 80x86 (or later) microprocessor. The specification was created by a group of eleven companies in the industry (including Microsoft, Lotus and Intel, as well as Quarterdeck, Rational Systems and Phar Lap), to allow multiple DOS extender applications to multi-task reliably under a multi-tasking environment. It superceded the VCPI specification (see below).

DPMI is a specification that exists in two versions. The 0.9 version was implemented in Windows 3.0. The 1.0 specification added new features while remaining fully compatible with 0.9. OS/2 2.0 is compatible with the 0.9 level, and adds some 1.0 features.

Virtual Control Program Interface (VCPI) is an earlier DOS extender specification, created by Phar Lap and Quarterdeck. While it allowed 386 expanded memory managers and DOS extenders to coexist, it did not address the problem of DOS extenders coexisting within a multi-tasking environment. Examples of applications written to this specification include Lotus 1-2-3 version 3.0 and Autocad/386 (Lotus 1-2-3 3.1+ changed the type of extender used from VCPI to DPMI).

MVDM memory management

MVDMs support all of the above types of memory, providing Virtual Device Drivers for both EMS and XMS functions, thus allowing DOS applications all the memory management functionality possible under DOS, including use of UMBs for the DOS 5.0 LOADHIGH, DEVICEHIGH commands and for TSRs. Indeed, DOS emulation in OS/2 2.0 VDMs owns the UMBs, but can free them for application use with statements in CONFIG.SYS or in DOS Settings. Note that the availability and size of UMBs depends on the same principle as under DOS (ie configuration of that DOS virtual machine). VDM DOS settings like MEM_INCLUDE_REGIONS and MEM_EXCLUDE_REGIONS allow some detailed configurable control over UMBs for separate VDMs. This is an advantage over DOS itself, which for obvious reasons can only apply one UMB configuration for all DOS applications that load, leading to many DOS users maintaining multiple CONFIG.SYS files for their different required configurations, and rebooting between them.

OS/2 2.0 provides Virtual Device Drivers (VDDs) for both EMS and XMS. The Virtual Expended Memory driver (VEMM.SYS) supports up to 32 MB of expanded memory per VDM, and the Virtual Extended Memory driver (VXMS.SYS) up to 16MB per VDM. These figures could be made higher, but are restricted for compatibility with EMS and XMS support under DOS. DPMI applications can access up to 512MB in OS/2 2.0. In contrast, DOS and DOS extenders like Windows 3.x can offer extended memory only up to four times the physical memory in the machine or 16MB, whichever is less.

Virtualising EMS, XMS access means one VDM′s use of memory does not affect others - more than one VDM can use XMS and EMS.

EMS and XMS memory is mapped into the system′s linear address space, and managed just like any other allocated memory. VEMM is compatible with LIM EMS 4.0 specification, and VXMS compatible with LIMA version 2.0 functions. The amount of EMS or XMS is configurable in CONFIG.SYS or in DOS settings.

Protection

VDMs operate in entirely separate process address spaces, and are controlled by a VDM Manager. They can thus be terminated on detection of illegal instructions without affecting the rest of system, or when the application is "hung." Thus problems in one VDM do not corrupt others, nor the system.

Memory management

Since VDMs take advantage of the virtual memory and paging facilities of the 386, they are swappable and, therefore, starting several DOS sessions will not significantly increase system memory requirements. Note, also, that each DOS session is individually configurable, so that EMS and XMS support can be switched off if not required, reducing overall virtual memory requirements. As always, of course, there is a balance to be made between memory usage and performance when overcommitting memory.

DOS emulation

All DOS services (eg I/O) are emulated within the MVDM kernel or passed to the OS/2 kernel (eg file services). Most of this emulation runs in protected mode outside the VDM (hence the large amount of DOS memory available). All documented (and some undocumented) DOS features are supported (such as device driver loading/support, program loading and execution, memory management) as well as all documented (and some undocumented) DOS interrupts (INT 20h, INT 21h, INT 27h etc). This provides a highly compatible "DOS 5.0-like" environment.

Input/Output

File Input/Output (I/O) in VDMs is made through the OS/2 file system, which, via VDM DOS emulation, provides a compatible interface to file I/O for DOS applications. Thus DOS applications can, without modification, take advantage of OS/2′s Installable File Systems (IFS) like HPFS, and the enhanced FAT (see "File systems" on page 29). Other I/O is performed either by DOS emulation or via Virtual Device Drivers (VDDs), such as BIOS, video, printer, keyboard (see below). DOS applications may continue to run and access system resources quite unaware that those resources are virtual, and real access to devices is provided by OS/2 2.0 itself. This guarantees compatibility but also allows DOS applications to take advantage of the inherent strengths of the OS/2 environment.

Virtual Device Drivers (VDDs)

OS/2 2.0 isolates DOS and Windows applications from I/O devices that are controlled by OS/2 device drivers, by emulating, or virtualising them for one or more DOS or Windows applications. This is done by Virtual Device Drivers (VDDs). VDDs provide a virtual instance of the real hardware, which is controlled by a physical protected mode driver (PDD - see "PDDs" on page 22 below).

VDD features

VDDs therefore provide the following support:

1. **Protection:** VDDs allow DOS applications to access hardware and BIOS without affecting other VDMs or other protected mode processes. This prevents VDMs from corrupting each other, or the system.
2. **Virtualisation:** VDDs avoid direct hardware access from DOS applications, but provide a virtual, or emulated, hardware state that lets the application think it is doing so. To do this, they maintain a virtual hardware state for each VDM. This means applications can, for example, access BIOS and video RAM (as Lotus 1-2-3 does), and receive hardware and software interrupts. In addition, VDDs can either perform I/O through a PDD, or directly address an I/O device itself for greater performance. In this way, virtual video drivers can support fast screen I/O to match performance expectations of DOS users working with programs like 1-2-3.
3. **Sharing:** VDDs allow sharing of devices across the system between DOS and OS/2 applications where there is a VDD/PDD combination for that device. There is also support for many DOS device drivers, to allow devices that do not have a VDD/PDD combination, to be supported in a VDM, for access exclusive to that VDM.
4. **Memory:** the VDD is a protected mode driver which allows virtual device driver support to be loaded above the VDM′s first megabyte of address space. This means that there is no memory impact on the use of VDDs for DOS applications. Only where DOS device drivers need to be loaded is there any impact under 1MB, and then the same as in DOS.

PDDs

Physical Device Drivers (PDDs) correspond to OS/2 device drivers in 1.3, but with an important difference: they execute entirely in protected mode, unlike the bimodal 1.x device drivers. Real mode interrupt handling is no longer required since VDMs run in V8086 mode - all interrupt processing is therefore done in protected mode. Most OS/2 1.x device drivers work “as is”: the part of the driver capable of being run in real mode (needed for the 1.x DOS compatibility box) is not supported, but their protected mode portion will usually work. However, an old 1.x device driver is unlikely to service VDMs, because the VDD requires support from the PDD that does not exist in older drivers.

This also means that no portion of the OS/2 device drivers needs to be located below the 1MB line, and therefore increases the size of the environment available to DOS applications. But DOS applications may still make use of these PDDs through the VDD interface.

How VDDs work

VDDs work by facing two ways: to the application, providing a virtual hardware state, and to the PDD or device, performing the physical I/O.

At boot time, VDDs are loaded, and many establish communications with the corresponding PDD via a direct call interface. The list in the section “VDD examples” on page 24 shows which VDDs have a direct interface with an OS/2 PDD. The VDD controls the DOS application′s access to the device, and relies on the PDD to manage the physical hardware operations.

The example below, and the accompanying diagram, show how the virtual COM driver (VCOM.SYS) works.

1. The COM PDD services all hardware interrupts from the asynchronous ports, and buffers the data being transmitted or received.
2. The direct call interface between the VDD and PDD allows the VCOM to emulate asynchronous BIOS functions to send and receive characters, or to set and query the state of a COM port, as well as receive the interrupts passed on by the PDD.

3. All of the time, the DOS application believes it is controlling these functions itself (just like it would under DOS), and is having interrupts and data from the PDD and the COM port reflected back to it by the VDD to maintain the illusion. The DOS application is therefore accessing only a virtual copy of the COM port. The VDD gets control when a DOS application performs direct I/O to a port (IN our OUT instructions), or via BIOS or other software interrupts (INT instructions).
4. OS/2 applications can simultaneously access the COM functions via the PDD using normal OS/2 function calls like DosOpen.

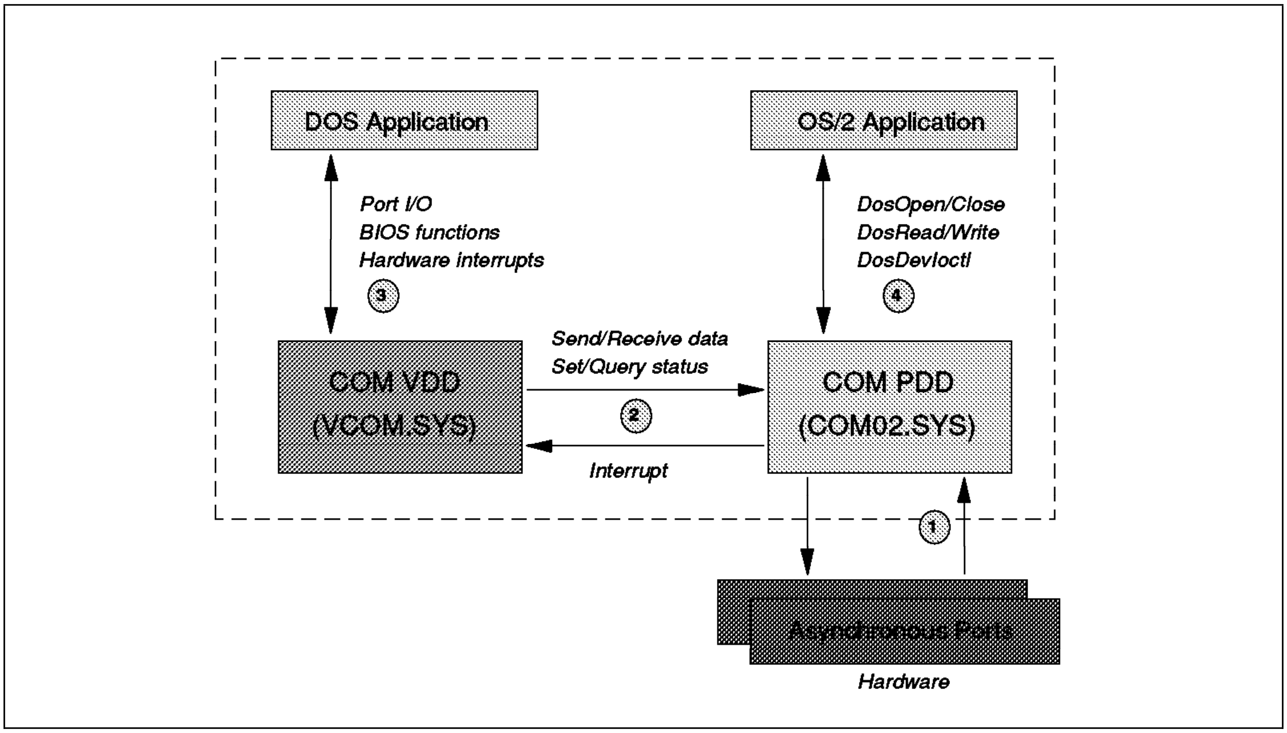


Figure 3. How VDDs work

Not all VDDs need to operate with a PDD in this way. Some VDDs (eg video) directly access the device for performance reasons. DOS applications still do not address the device directly themselves - the VDD still virtualises the I/O to the DOS application.

The VDD/PDD interface is required where hardware interrupts need to be simulated into one or many VDMs (COM is a good example). This is important so that DOS applications that want to control the hardware device directly, do not need to get control at interrupt time, but can be deferred until the OS/2 kernel dispatches the VDM task. This preserves system integrity and maintains overall system performance. This interrupt latency, as it is called, may cause problems for a minority of applications that are highly dependent on real-time interrupts. But the vast majority of DOS applications, even high speed communications, can be dealt with successfully.

VDDs can be made swappable, and are installed using DEVICE= statements in CONFIG.SYS.

Additional VDDs can be written by third parties for their devices, via a published programming interface, using the Virtual Device Helper (VDH) services provided by the MVDM kernel.

A VDD is required only if a device will be shared with other VDMs or OS/2 processes. If a particular device is to be used exclusively by one DOS application, the DOS device driver may be used. Thus, OS/2 2.0 can provide “generic” support to most DOS device drivers, but such support is limited to that VDM. It does provide greater compatibility and a wider support of devices, enhancing the DOS compatibility of OS/2 2.0 (see “DOS device drivers” on page 40 for more). DOS Device drivers can be loaded via the DOS Settings (see “DOS Settings” on page 41).

VDD examples

VDDs exist for most of the common device types. The following table lists some of the ones that are included in OS/2 2.0:

Table 3. Virtual Device Drivers

Name	Description	Interfaces with PDD
VBIOS ♣	DOS system areas like ROM BIOS and interrupt vector tables are mapped from physical memory into the VDM address space by VBIOS.SYS.	
VCDROM	CD-ROM support	*
VCMOS ♣	CMOS data area and Real time clock support	
VCOM	Asynchronous COM ports	*
VDMA ♣	Direct Memory Access (DMA)	
VDPMI	DOS Protect Mode Interace - used in WIN-OS/2 and applcations like Lotus 1-2-3 3.1+	
VDPX	DOS Protect Mode Extender	
VDSK ♣	Disk/diskette, only for INT 13 copy-protection	*
VEMM	Expanded memory support - up to 32MB per VDD	
VFLPY ♣	Floppy disk interface	*
VKBD ♣	Keyboard - also has INT 9h emulation code to perform functions usually performed by CBIOS, such as key and scan code queuing, update of keyboard LEDs, and processing for Print Screen, SysReq, Break and Pause (INT 5h, INT 15h, INT 1Bh respectively).	*
VLPT ♣	Printer	*
VMOUSE	Mouse	*
VNPX ♣	Numeric co-processor (x87)	
VPIC	Programmable Interrupt Controller (8259)	
VTIMER ♣	Timer	*
VWIN	For WIN-OS/2 “seamless” support	*
VXMS	Extended Memory (XMS) specification - up to 16MB per VDD	
Video (VCGA, VMGCA, VEGA, VVGA, V8514, VXGA, VSVGA)	Supports all documented video modes from mono to XGA, including the documented modes of VGA and 8514/A.	*

♣ refers to base set of VDDs automatically loaded at system initialisation time; others can be loaded via DEVICE= statements in CONFIG.SYS.

Multi-tasking

OS/2 has always been highly regarded for its industrial-strength multi-tasking design. OS/2 2.0 continues those strengths, harnessing the CPU-level support of the 386 processor for even more efficient multi-tasking.

Why do I need multi-tasking?

Some people believe that they do not need multi-tasking, on the premise that “I can only manage one thing at a time myself.” However, the simplest way of understanding the need for multi-tasking is to imagine what happens when you get an interruption at work. Consider this simple scenario:

You are working on a report (using your word processor) when the telephone rings. It′s the boss, and he wants the latest sales figures from you ***now***. Quickly, you stop what you′re doing, put aside the word processor, and connect to the database to run the report routine. While logging on and running the report, you start your spreadsheet program, so you can put the report from the database into the spreadsheet, to make some extra calculations that are required. When the report has run and the spreadsheet has recalculated, you take the figures into your charting program to make a graph. You may also wish to cut and paste this graph into a new document that you create with your word processor (which you call up quickly after you put it aside) to explain the figures to the boss. In addition, if you are a cautious person, you may back up the report you have created and keep your own copy on diskette (in which case it would have been nice to have started that formatting five minutes ago when you were logging on!) Once you′ve created the new document, with the latest sales figures and calculations shown in a graph, you may wish to send it to the boss via electronic mail. Finally, you can return, after a brief rush of activity, to your original report where you left it.

Such a scenario is characteristic of much of our work. Many of us are driven by interruptions. In fact, it is not an exaggeration to say that any PC user who has a phone can benefit from multi-tasking. The benefits are obvious:

- handle interruptions
- less waiting for one operation to complete (the “hourglass” in Windows 3.x is a sign of a platform with only limited multi-tasking)
- let the computer do the work while *you* stay productive
- handle “beneath the covers” function like logging on, printing and running database reports
- give smooth, even performance between different tasks

But perhaps the most persuasive argument is to imagine how the above scenario would have been achieved using DOS!

What is pre-emptive multi-tasking?

Most users understand that multi-tasking allows more than one application to run at once. Of course, without multiple processors, they do not actually execute instructions simultaneously - the CPU still only processes one instruction at a time, but the operating system can divide CPU time between several processes to make it appear to the user that those processes are executing simultaneously. Therefore most multi-tasking operates on the basis of some form of CPU time-slicing.

Furthermore, it is important to distinguish between multi-tasking, where applications continue to execute when in the background, and context switching, where they lie dormant until given the user focus. Some environments (such as OS/2 1.3 DOS Box, Windows 3.x standard mode, and DOS 5.0′s Task Switcher) are only context switching. In these cases, the background task does not receive CPU cycles, and time is devoted only to foreground tasks.

Some of the tasks in the scenario outlined above can be achieved using systems that offer only context-switching, or non-pre-emptive multi-tasking. But a pre-emptive multi-tasking system would allow the whole job to be achieved faster and more efficiently, because the overall performance of the system would be more balanced (see "Why is it important?" on page 26). Systems management tasks like background "agents" to collect configuration and performance data, are very difficult to implement except in a pre-emptive multi-tasking system (see "OS/2 for client-server" on page 98).

Multi-tasking is pre-emptive when the processor allocates a finite time to each task, and then switches the processor to another task, even if the first task is not "ready" to give up the processor. Non-pre-emptive implies that tasks can "hold on" to the processor within certain limits.

Why is it important?

To understand the significance of the difference between pre-emptive and non-pre-emptive multi-tasking, it is useful to contrast the approach taken by a DOS multi-tasker like Windows 3.x, with OS/2′s approach.

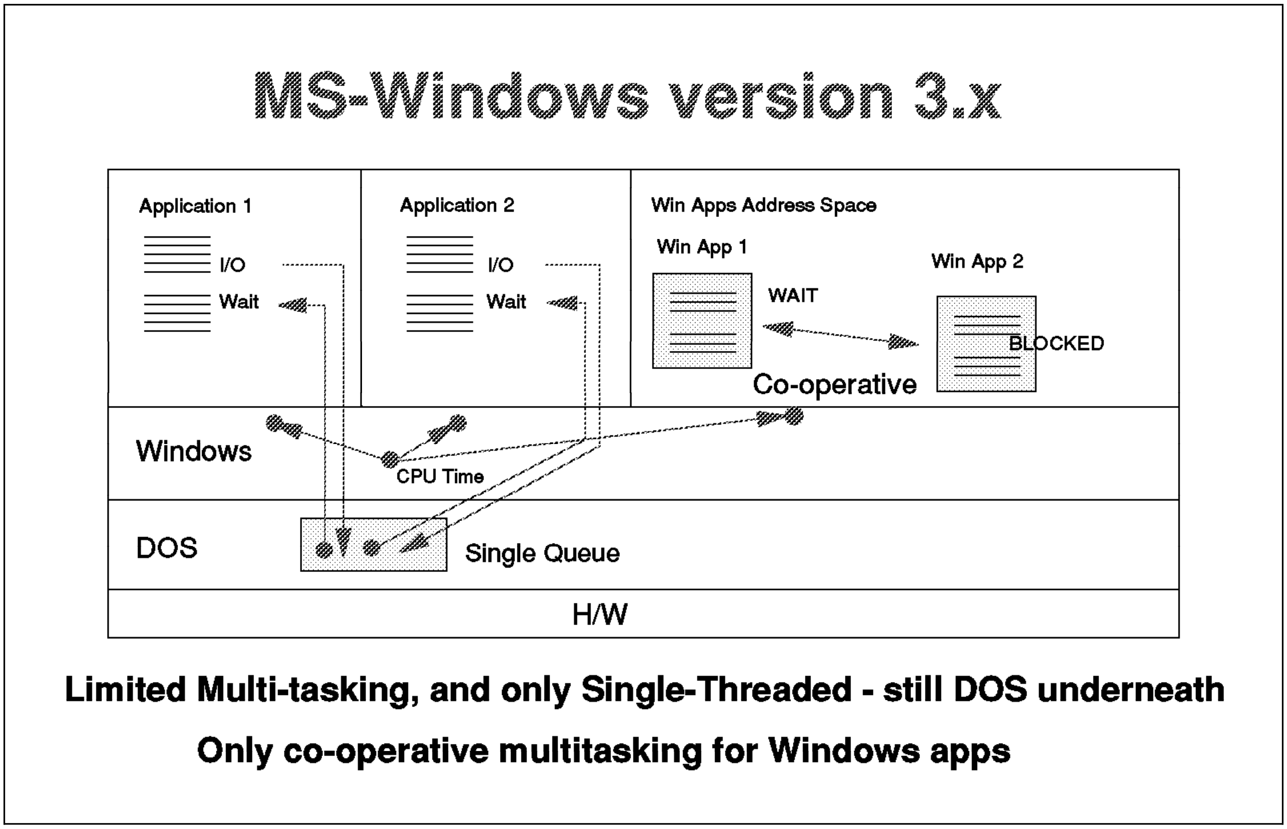


Figure 4. Multi-tasking under Windows 3.x

Windows 3.x implements a time-slicing scheduler on top of DOS, which is in itself a single-tasking operating system. In Windows Enhanced mode (which requires a 386 or above - see "Windows 3.x modes" on page 48), DOS applications run in virtual machines which are pre-emptively multi-tasked; Windows applications run together in one separate virtual machine; this process is pre-emptively multi-tasked in relation to the other DOS sessions.

However, Windows applications themselves only support co-operative multi-tasking, which means that Windows applications need to be "well behaved" to give up the processor attention to allow other tasks to proceed, with specific use of functions such as yield(). Windows applications are therefore pre-emptively multi-tasked with respect to the rest of the system, but only co-operatively among Windows applications. When a Windows program needs to do some lengthy processing, or is waiting for I/O, it needs to take special precautions against halting all other Windows programs for this duration. Needless to say, some do not, and that is why the user can be frequently faced with the hourglass icon for periods of time when doing extensive processing or I/O. Furthermore, a Windows application has no guarantee of processor time within any period, which can potentially cause problems for applications needing regular processor attention, such as communications programs.

Even though DOS applications are pre-emptively multi-tasked in Windows, the scheduler algorithm is fairly unsophisticated, time-slicing on a static allocation basis (the proportions can be tuned by the user, but they remain static). This means that, even though an application may be unable to make any effective use of its processor share (for example when it is tied up with an I/O request), it will get it anyway, even though more deserving candidates are waiting their turn. Thus the multi-tasking can be bogged down with applications that are “I/O bound,” like Application 1 in the above diagram, waiting for an I/O device to respond.

Finally, since DOS itself is single-tasking, it has only one I/O queue, and cannot therefore easily handle multiple I/O requests except by processing them serially. So extenders like Windows that rely on DOS may be further weighed down by the single-tasking character of the operating system.

One of the best illustrations of the I/O limitations of Windows 3.0 or 3.1 is to try running a spreadsheet recalculation, and a download from a host machine to the hard disk, while also attempting to format a floppy disk from the DOS prompt (not unlike the scenario we discussed previously). You will see that at least one of the tasks (usually the diskette format if it is not in the foreground) will process in a very “jumpy” way, and often will pause for very long periods. It is not unusual in such a scenario, to see one of the tasks come to a near standstill as the others progress. If one of the background applications is a communications program, the lack of processing time available to the background application may even result in loss of data.

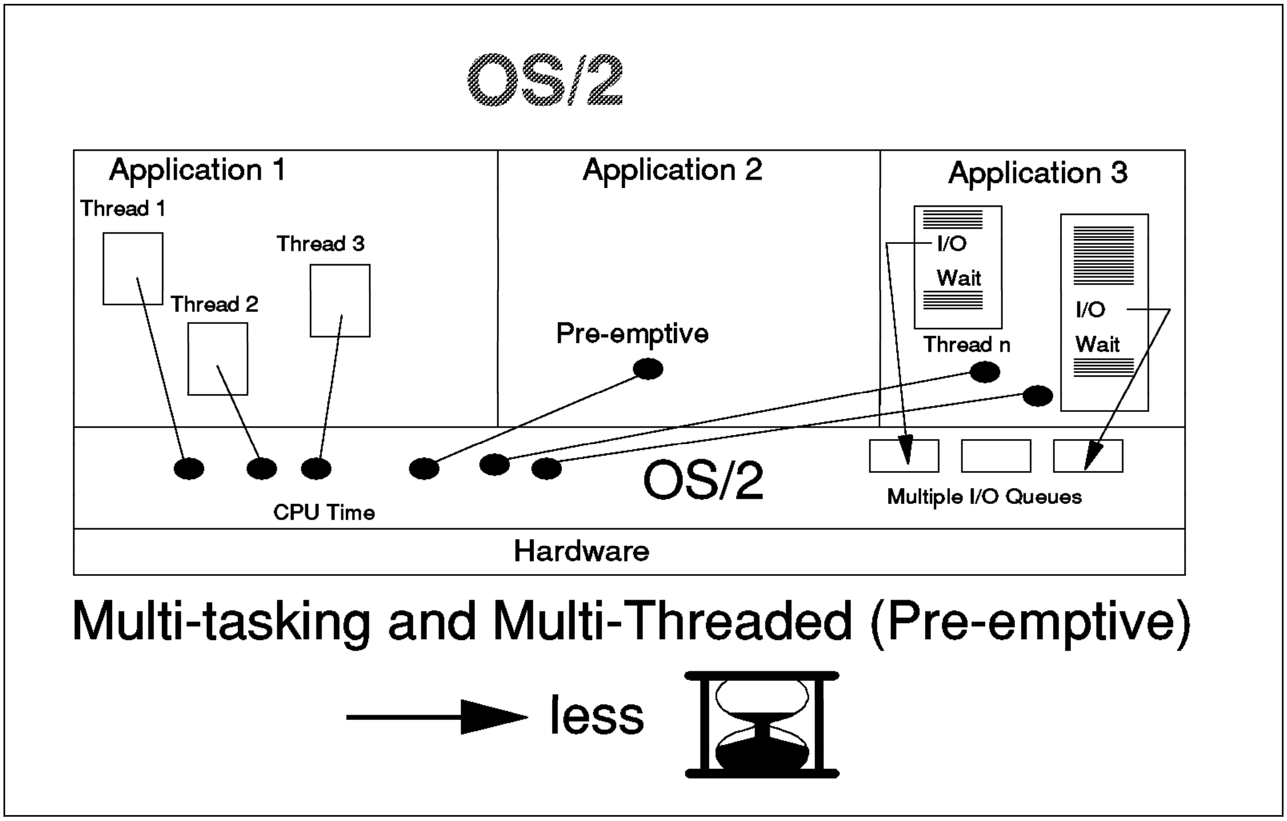


Figure 5. Multi-tasking under OS/2

In contrast, OS/2 pre-emptively multi-tasks all processes, and also provides another, more granular unit of execution - the thread. Threads can be thought of as distinct subroutines within a process, which can execute without immediate reference to the main logic, such as file I/O, database read/write, recalculation. In OS/2 these threads can be dispatched separately, and be multi-tasked with each other, and along with other processes. In fact, the basic unit of execution for the scheduler is the thread, and all processes contain at least one thread.

The advantage of threads is that more time-consuming operations can be put into separate threads, and the main thread of the program devoted to user input; this helps the application to maintain an interactive and responsive feel even when processing other sub-tasks in the background (see Application 1 in the diagram). In this way, threads help prevent the kind of “I/O bound” feel to many DOS and Windows programs. But it should be noted that use of threads applies to more than just I/O operations: it is a powerful tool that can be used for any operation that can be run in background while allowing the user to regain control of the application, for further input.

Threads do not exist in any currently available version of Windows (at the time of writing). Indeed, though the promised future product, Windows/NT (a different operating system) is claimed to offer multi-threading, Windows applications will need to be rewritten to take advantage of this facility, since Windows 3.x programs are inherently single-threaded. Porting from a Windows 16-bit program to a future 32-bit version of the Windows API will not be enough, nor can threads be easily added later; they must be woven into the basic design of the program to be used effectively. In fact, another promised feature of Windows/NT, symmetric multiprocessing, may have little value until multi-threaded Windows programs exist. Symmetric multiprocessing is the ability of a system to run multiple threads of execution on different processors concurrently (eg on multiple 486s within the same machine). By definition it requires a multi-threaded system, and therefore, to take full advantage of it, multi-threaded applications, and Windows today is only single-threaded. So this feature will need new versions of applications to show its advantage. In fact, the most likely early candidates for multi-threaded Windows/NT programs are those (like Sybase SQL Server and Oracle) that are already multi-threaded by design because they exist today as multi-threaded OS/2 programs. It is one of many examples where Windows/NT promises little that is not already available with OS/2 Version 2.0.

Again, in contrast with DOS and Windows 3.x, the OS/2 scheduler is much more intelligent than a simple time-slicer. It can detect when applications are I/O bound (like Application 2 in the diagram), and shift CPU time to another thread or process. Thus, priorities can be changed dynamically to preserve overall system responsiveness.

And, to complete the picture, since OS/2 itself has a multi-tasking, multi-threaded design, it can provide multiple I/O queues and therefore overlapped I/O between processes. This helps to improve the perceived performance, especially in heavy system loads.

The following table summarises the differences between Windows 3.x and OS/2 2.0:

Table 4. Multi-tasking - Windows 3.x and OS/2 2.0

	Windows 3.x	**OS/2 2.0**
Multi-tasking DOS applications	pre-emptive	pre-emptive
Multi-tasking Windows applications	co-operative	pre-emptive (if in separate VDMs)
Priority	static	dynamic
I/O processing	serial, single queue	overlapped, multiple queues

What this means to the user is that OS/2′s multi-tasking is smoother, more consistent, less likely to be bogged down by I/O or by an application attempting to monopolise the processor, and more responsive and interactive in feel.

Multi-tasking and the user interface

One of the advantages of a graphical environment is that it can represent concurrent tasks by icons and windows on the screen, thereby giving a visual indication of the number and variety of tasks. This is true of both Windows and OS/2.

However, OS/2 2.0 builds on the basic benefits of the “first generation” GUIs like Windows 3.x and OS/2 1.3, and adds some features specifically designed to aid smoother multi-tasking and switching between tasks:

- the “Window List” is more functional than the Windows 3.x task list, allowing several applications to be “resurfaced” on the screen, or closed, at once, rather than moving to and clicking on discrete icons for each task
- the ability to “hide” windows when not in use, rather than reducing to an icon on the screen[2] , results in a less cluttered desktop, making it easier to find what you want quicker, especially in screen resolutions with limited screen “real estate,” such as VGA.
- OS/2 2.0 allows whole groups of applications and data to be grouped together in a logical unit (a “project,” as it were) by putting them together in a common folder. This is more convenient and flexible than just creating directories and copying files manually, as you would have to in DOS and Windows. (You can also be more flexible in naming such folders, going beyond the eight character limits even in a FAT file system - see “Descriptive names” on page 81)
- Each folder can also be given a “workarea” property. This allows several applications to be opened, hidden or closed, together, simply by opening or closing the folder, rather than going to each window individually. For example, a folder containing a spreadsheet, word processor and graphics applications may have all these programs running together, but then close them or hide them in a single operation on the folder, and restore them all to the state they were in previously, in another single operation. This means the whole state of work in progress can be quickly set aside, and another begun or resumed easily. OS/2′s ability to save and restore what you were doing from one boot to another is a specific example of this workarea feature

None of these user interface features appear in either Windows 3.x or in the current beta test releases of Windows/NT. The result of all of this is that the Workplace Shell is much easier to use in a multi-tasking scenario than older GUIs. The user interface matches the internal design in being optimised for multiple tasks, making it easy to move from one task to the next.

File systems

OS/2 offers superior file system support, which leads to increased performance. The OS/2 architecture also allows other file systems to be installed in a modular fashion (Installable File Systems or IFS). It is therefore more flexible in design to accommodate future enhancements; for example, OS/2 now includes CD-ROM support via an IFS.

OS/2 also provides support for very large disks, a consideration that is particularly relevant to server environments. OS/2 supports hard drives up to 2GB in size. In Windows 3.1 the limit is only 1GB.

[2] Windows can either be hidden (the installed default), minimised to the desktop, as happens in Windows 3.x and OS/2 1.3, or minimised to a Minimised Window Viewer, according to the user′s preference. This can be set on a per-object basis.

OS/2 2.0 provides file I/O services not only to OS/2 applications, but to DOS and Windows applications running in VDMs. Therefore, DOS and Windows applications can take advantage of the advanced function, without having to be modified, since the DOS emulation of MVDM provides a compatible interface to the file system for DOS applications.

Both major OS/2 file systems have been improved under OS/2 2.0, allowing better performance not only for applications themselves, but also in the paging performed by the system when using virtual memory.

High Performance File System (HPFS)

HPFS was first introduced to OS/2 in version 1.2, and is an example of the kind of advanced function that has not yet been implemented in less sophisticated systems like DOS. It was introduced as an alternative to the File Allocation Table (FAT) system which came from DOS. HPFS is particularly good for managing large disks and partitions or large files. It provides fast and consistent performance, outperforming DOS-based FAT systems in tests run by IBM, in nearly all cases. (OS/2 2.0 implements an enhanced FAT system that uses some similar caching features as HPFS, which can also give high performance in many circumstances - see below).

HPFS is particularly good in disk utilisation (compared to FAT). It uses a highly contiguous file allocation system, which results in especially good performance (relative to FAT) in accessing files or data in a cluttered or full partition. It implements a B-Tree directory structure and search algorithm, as opposed to sequential under FAT. HPFS also allows for multi-threaded I/O, caching of directory pointers in memory for quicker access of last directories used, and read-ahead and lazy write (lazy write buffers up write requests from applications and commits them to disk after a given time or during disk inactivity.) These advanced features allow for substantial performance increases and greater tuning. HPFS can also provide write error recovery on the fly with “hotfix” facilities.

Since it is FAT-compatible at API level, applications running under OS/2 can use either system, and do not have to be written specifically for one or the other. It also presents a consistent interface to other components of OS/2 like MVDM, to allow DOS and Windows to use HPFS volumes as if they were FAT.

HPFS also supports the use of long file names, for greater usability, so instead of having LJS1290.TXT, you can have a file name “Letter to John Smith December 90.” Obviously applications need to be coded with this in mind. DOS and Windows applications can use files that adhere to the 8.3 naming system on HPFS without any difficulty.

HPFS has been enhanced in version 2.0 to add performance-related features such as command chaining (providing a list of contiguous sector requests required to fulfil an I/O request) and scatter/gather facilities such as are supported in Small Computer Systems Interface (SCSI) adapters to gather physically discontiguous pages in a data buffer, and perform I/O in a single operation.

Enhanced FAT

OS/2 2.0 includes an enhanced version of the FAT file system which is completely compatible with the FAT system under DOS. This gives greater performance but full compatibility with existing FAT systems. It adds features like lazy write and improved caching to FAT. This means that DOS applications running with a FAT file system under OS/2 2.0 will be substantially faster for disk-based operations than under DOS. All of these benefits can be obtained without having to reformat your hard disk - the OS/2 enhanced FAT driver works with existing DOS FAT volumes.

SCB exploitation

OS/2 2.0 implements the Subsystem Control Block (SCB) architecture for more intelligent and efficient disk access. SCB defines a standard way of communicating between device drivers running on the main system CPU, and I/O processors located on advanced function adapters (like SCSI) that are capable of operating independently from the CPU. SCB therefore fits naturally with the use of SCSI adapters, and will allow better exploitation of SCSI functions like scatter/gather and command chaining. The implementation is transparent to the OS/2 user and the developer; there are no application considerations, simply improved performance. It is simply a means of getting more out of a system that provides advanced intelligent adapters for disk I/O. Exploitation of newer, more advanced devices such as SCSI is better under OS/2 than under Windows, which does not provide such features.

Broad hardware support

OS/2 1.x was separately distributed by both IBM and Microsoft. IBM delivered IBM OS/2, which was specifically optimised for and supported on IBM equipment (AT and PS/2 family), while Microsoft distributed MS OS/2 to its OEMs like Compaq and Olivetti (many of these versions had manufacturer-specific modifications too). This meant that there were different versions of OS/2 according to the machine it ran on. Although this was not in essence any different than had always been the case for DOS, the perception was that supporting OS/2 in a mixed hardware environment was hampered by the lack of a “generic” version of OS/2.

This problem has been overcome in OS/2 2.0, which supports a broad range of machines with an Intel 386SX or above, with a single version of OS/2. This shows that OS/2 2.0 does not run only on IBM equipment. Open hardware support applies not only to the base system, but also to Extended Services for OS/2 and OS/2 LAN Server (some models may require more memory installed to run the systems extensions). IBM supplies regularly updated lists of the models it has tested via bulletin boards (see the list in Appendix D, “OS/2 Bulletin Board Systems” on page 139) and IBM representatives. Among the vendors whose models are supported, are Compaq, AST, Olivetti, Toshiba, Hewlett Packard, Dell, Gateway, Wang, DEC, NCR, Tandy, ACER, CompuAdd and many others. Ask your IBM representative to look for the PCMTABLE package on the MKTTOOLS disk.

In fact, even though IBM cannot test OS/2 on all the models and manufacturers in the market, it is likely that most PCs equipped with an Intel 386SX or above processor, will work. More models are being tested all the time, and IBM is committed to working with as many manufacturers as possible to determine OS/2 compatibility. Indeed, the evidence from user registrations for OS/2 2.0 in Europe alone, suggests that users are already running OS/2 2.0 on hundreds of machines that have not yet appeared on any “official” list. By 2 September, 1992, 802 models had been recorded from registration cards that do not appear on the list of tested models.

OS/2 is not limited to machines with Micro Channel architecture. Even the supported PS/2 models include AT-bus machines like the Model 40, as well as Micro Channel machines like the Model 57. AT-bus, Micro channel and EISA machines from a variety of vendors are supported.

Furthermore, IBM is now making OS/2 2.0 available, preloaded on the hard disk of selected IBM PS/2 models, including the Models 56 and 57. This is a convenience for users, who do not need to install the operating system at all, but can switch on the PC to discover the system already set up for their use, along with a number of new applets and help on using the system. IBM is making this facility available to other IBM-compatible hardware manufacturers, and arrangements have already been made with a number of vendors, including Dell and Olivetti, to provide OS/2 2.0 with selected models in their hardware range. Contact your supplier for the latest information.

OS/2 also has broad support for different displays, disks and printers. Most of the major display standards (VGA, 8514, XGA) have full support. SuperVGA (SVGA) support exists for some adapter types already (some existed in OS/2 1.3), and more drivers are appearing from the SVGA chip set vendors like Trident and Tseng. Many SVGA DOS and Windows drivers are also supported in full screen DOS and WIN-OS/2 sessions, even if the main desktop has to run in VGA mode if there is no PM screen driver. IBM plans further enabling of SuperVGA support in an update option to be made available by the end of 1992.

OS/2 supports 205 models of printer, which covers over 90% of the printer market, from dot matrix, to inkjet and laser printers. Models from the major vendors like Epson, Hewlett Packard and IBM receive wide coverage (look at the file PMSETUP.INF in the \OS2 directory for the complete list). There is also generic text printing support which should work on any printer. Furthermore, if a Windows driver exists for a device, it can be installed in WIN-OS/2, and used for Windows applications running under OS/2 2.0 (the product includes additional Windows printer drivers for Canon, HP and NEC printers). Where both a Windows and an OS/2 printer driver exist, they can both be installed in one step at installation time.

Generic INT13 disk support is provided in OS/2 2.0, which should allow most of the common drive types to work. IBM and Adaptec have developed a common specification which will make the development of vendor-specific SCSI drives much easier. Drivers already exist for some of the OEM SCSI adapter vendors. More are in development, and IBM is producing tools to help support the development of drivers for displays, printers and disks.

OS/2 1.x suffered from a relative lack of OS/2 device driver support, but OS/2 2.0 can make use of many devices via their DOS device driver (see “DOS device drivers” on page 40). Devices such as hand scanners and fax cards can often be supported in this way, broadening the range of devices available to OS/2 users.

OS/2 device support is broadening as the industry increases its support, but IBM continues to develop drivers itself and work with other vendors to ensure that even wider support is provided.

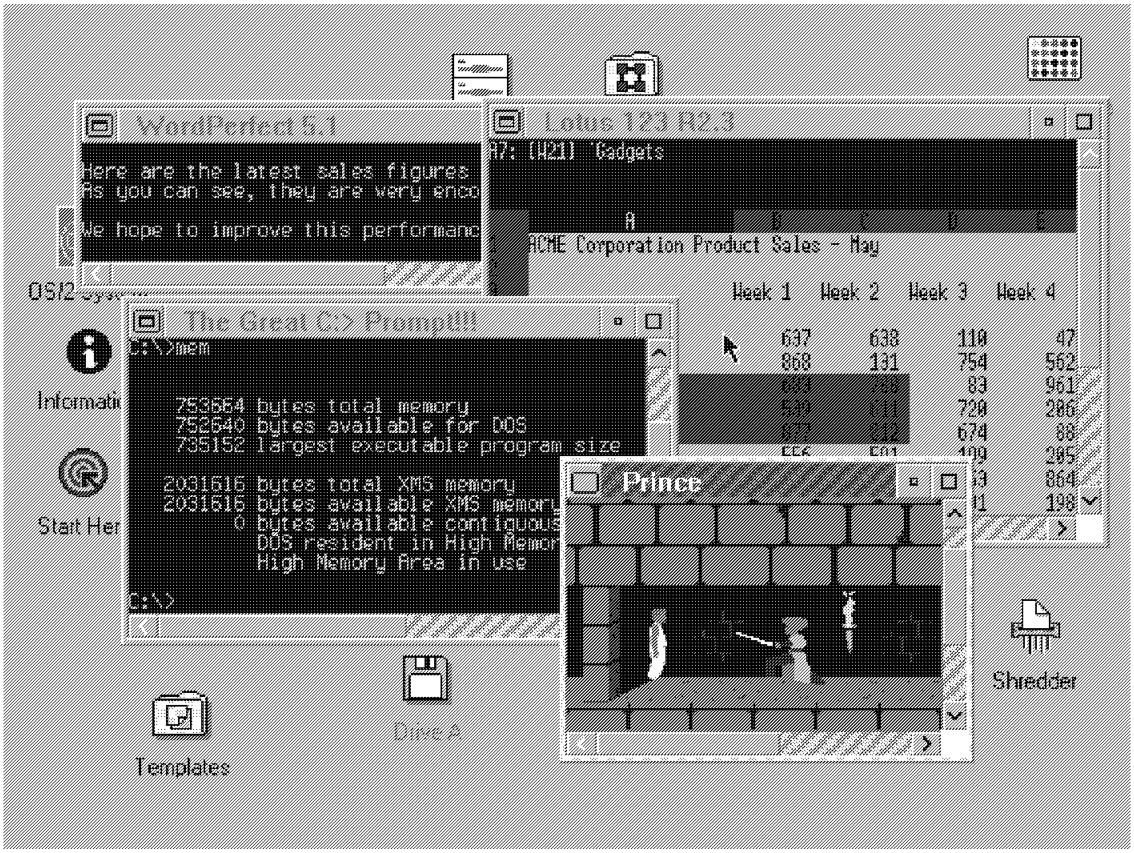
Better DOS

There are over 20,000 DOS applications available today. In order to be an integrating platform for PC-based applications, the primary task for OS/2 2.0 is to fully support that wide range of applications.

OS/2 version 1.3 was limited in the support it could provide for DOS applications, mainly due to the limitations of the 286 architecture on which the OS/2 1.x base was designed. However, use of the 386, and in particular the virtual 8086 mode of the processor, allows DOS applications support to be much more extensive in version 2.0 - indeed, a better DOS than DOS itself.

Multiple DOS applications

Several DOS applications can be run at once in OS/2 2.0. All DOS applications can be invoked from an icon on the Workplace Shell desktop, or from any DOS or OS/2 command prompt. They can be run in a full screen session (rather like they were in 1.3), or windowed on the Workplace Shell desktop. Both text-based applications (such as Lotus 1-2-3 v2.2 and WordPerfect) and applications that run in graphics mode (such as Lotus Freelance Plus or Harvard Graphics, as well as the Microsoft Flight Simulator or other games programs) can be run. Graphics mode applications can be run windowed alongside text-based applications running in windows. This cannot be done without restrictions in Windows 3.0, and although Windows 3.1 does allow VGA graphics to run in a window, this feature will not be available under Windows/NT, where, according to Microsoft, the application will have to run full screen.



```
Here are the latest sales figures
As you can see, they are very enco

We hope to improve this performanc
```

```
A7: [W21] 'Gadgets

          A                B        C        D        E
ACME Corporation Product Sales - May

                        Week 1   Week 2   Week 3   Week 4

                           637      638      110       47
                           868      131      754      562
                           683      288       83      961
                           539      611      720      206
                           677      212      674       88
                           556      501      109      205
                                             ..3      864
                                             ..1      198
```

```
C:\>mem

    753664 bytes total memory
    752640 bytes available for DOS
    735152 largest executable program size

   2031616 bytes total XMS memory
   2031616 bytes available XMS memory
         0 bytes available contiguous
           DOS resident in High Memor
           High Memory Area in use

C:\>
```

Figure 6. Multiple DOS applications in windows on the Workplace Shell desktop

DOS applications running in a window on the Workplace Shell desktop can take advantage of many of the ease of use features of the Workplace Shell, such as sizing the window to a convenient size, tiling windows so several applications can appear alongside each other, and hiding them to make more space on the screen while leaving the application running. Furthermore, many DOS applications will be able to take advantage of the font support in PM windows: you can change the font size to better suit the size of the window you are running in (the range of sizes available is determined by the display adapter you are using - on VGA, for example, there are 10). Again, all of this support is provided by PM: no modification is required to the DOS application. Contrast this with DOS, where getting a small font size for applications like Lotus 1-2-3, requires re-running the Install program, and the risk of creating an invalid setup.

DOS applications that support the mouse get use of the mouse, even when running in a window (there are DOS settings to give the DOS application exclusive use of the mouse pointer while in the application window - see “DOS Settings” on page 41). This is much easier to set up in OS/2 2.0 than it is under Windows 3.1 (Windows 3.0 did not support DOS windowed applications using a mouse), as it only requires one change to the DOS settings (which are automatically set up for many programs via the Migrate option - see “Migrating applications” on page 43); in Windows 3.1, you need to have already installed the DOS mouse driver in AUTOEXEC.BAT or CONFIG.SYS (thus reducing the memory available to all DOS applications whether or not they use the mouse).

All of this means that, without changing any of your DOS applications, the Workplace Shell environment can “add value” to the applications while preserving full compatibility with the way they are used under DOS.

Application integration

OS/2′s DOS support is not simply a question of compatibility - under OS/2 2.0, you can make your applications work together in a way that is not possible under plain DOS. It is possible to copy information from one application to the clipboard, and paste it into another. This can be between two DOS applications, between a DOS application and an OS/2 application, or any combination of DOS, Windows and OS/2 applications.

Moreover, both text and graphics can be cut and pasted between applications, (although graphics can only be received by those applications that are capable of handling bitmap data - mainly OS/2 and Windows applications). This allows two-way sharing of data between applications.

This means that DOS applications not only integrate better with GUI applications written for Windows and OS/2, but also with other DOS applications, even though they may not have been written to work together.

Imagine, for example, if you were using the DOS versions of Lotus 1-2-3 and WordPerfect, and you wanted to incorporate some figures from your spreadsheet into the report you are writing on the word processor. Under DOS, you would need to quit one application (the word processor) to load another (the spreadsheet), retrieve the spreadsheet file and make the conversion into a file format the word processor understands (perhaps plain ASCII, thus losing any formatting you have created in the spreadsheet), exit the spreadsheet application, load the word processor and the report again, and import the data into your report. Whereas in OS/2 2.0, you can simply use the clipboard. With the applications running side by side in windows on the desktop, mark some data from the spreadsheet, copy it to the clipboard, and paste it into the word processor - all without closing files or applications. The clipboard takes care of any data conversion. As the number of applications you need to integrate grows, so the difficulty of remembering different methods of data conversion grows, but in OS/2 2.0, the user simply works with copy and paste each time.

Thus, standard, unmodified DOS applications can be integrated under the OS/2 Workplace Shell. DOS applications can also be written to take advantage of the OS/2 environment, or OS/2 applications written to start DOS applications. One example of this is the OS/2 command prompt in OS/2 2.0, which can start DOS applications simply by typing the name of the .EXE file, just as one would from a DOS command prompt. This helps to reduce the need to differentiate between DOS and OS/2 applications, from the user′s point of view.

Another method of integration is via the named pipes mechanism, which allows DOS and OS/2 applications to be written to communicate with each other.

Multi-tasking of DOS applications

Not only can several DOS applications run at the same time, in separate windows on the Workplace Shell desktop, but each continues to run in the background while you are working on another. In this way, DOS applications can take advantage of OS/2′s pre-emptive multi-tasking, overlapped I/O, and OS/2 file system performance. Let′s take a look at each of these in turn:

Pre-emptive multi-tasking

As was explained earlier ("Multi-tasking" on page 25), OS/2 was designed to manage processor time smoothly between multiple applications, not only OS/2 applications, but also DOS and Windows applications running in Virtual DOS Machines (VDMs). This is more sophisticated than the simple time-slicing mechanism employed by Windows 3.x, and takes into account, for example, when applications are "I/O bound," waiting for an I/O device to respond. In OS/2 priorities can be changed dynamically by the processor according to such criteria, in contrast to the static approach employed by Windows 3.x. Therefore, even though both environments employ pre-emptive multi-tasking for DOS applications, the OS/2 scheduler is more sophisticated than Windows′ scheduler. Relative performance between applications can be tuned by a number of DOS settings (see "DOS Settings" on page 41 for some examples of DOS settings).

The result of all of this, is that OS/2 provides a more balanced performance between applications, not only those from DOS, but between them and Windows and OS/2 applications.

Overlapped I/O

All DOS applications have their I/O requests serviced by OS/2 (although the application makes calls to DOS I/O in the same way as usual, they are trapped and serviced by OS/2).

This allows DOS applications to take advantage of OS/2′s better I/O capabilities without needing to be rewritten. Just by running under OS/2 they can benefit from OS/2′s ability to overlap I/O requests and services. This improves performance in many scenarios, particularly where multiple applications are running.

Access to OS/2 file system

Although DOS applications have not been written to use the OS/2 file systems, they can without modification use them and derive the same benefits as OS/2 applications do. All file access uses the OS/2 file system. Both the HPFS and enhanced FAT file systems under OS/2 provide superior performance compared to DOS[3]

[3] See, for example, the benchmarks run by NSTL in their Software Digest Ratings Report Volume 9, Number 4

Therefore despite the inevitable multi-tasking overhead, many applications will perform just as well, and some will perform better, than under native DOS. In test cases run by IBM, multi-tasking performance of DOS applications was superior to that under Windows 3.x.

Memory usage

Memory has been one of the biggest constraints on the development of the DOS environment. The 640KB limit was for a long time an inhibiting factor for many users and application developers. Despite many workarounds, such as the LIM expanded memory, use of the HMA area, and the growth of DOS extenders (see “Memory: Conventional, Expanded, Extended” on page 18 for a more detailed discussion), memory management under DOS is a complex and frustrating task, and there still remain limitations for the large number of DOS applications written to use only conventional memory below the 640KB mark.

Comparison with memory usage under DOS

Part of the problem has been that successive releases of DOS have added function, but removed some of the vital conventional address space. The following chart shows the relative memory sizes free under various versions of DOS:

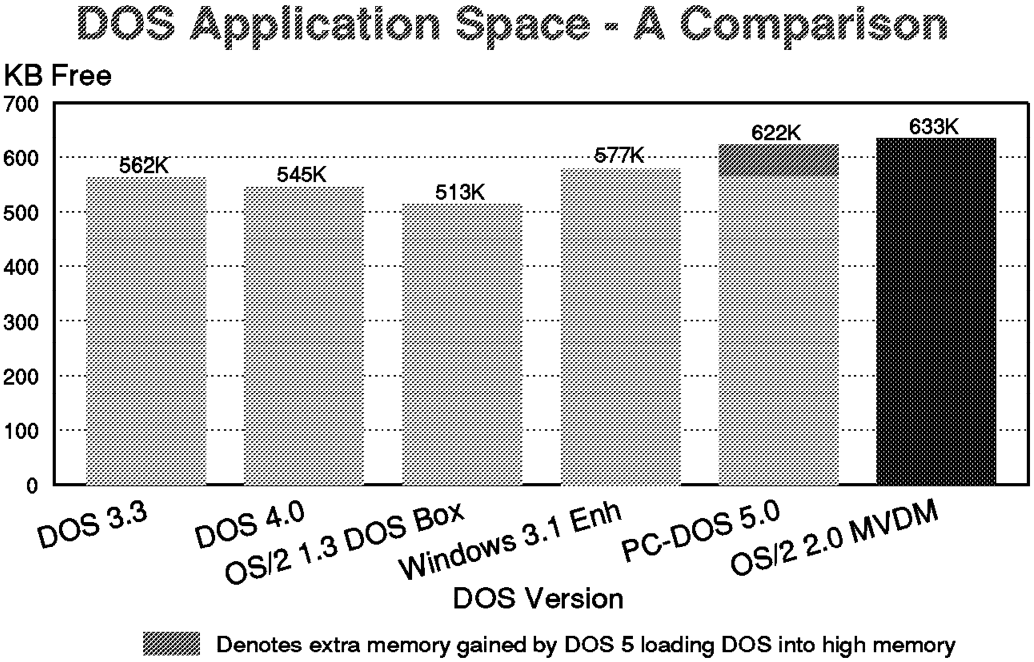


Figure 7. DOS application memory space - a comparison

As the chart also shows, OS/2 Version 2.0 provides a DOS conventional memory of 633KB, at least as good, if not better, than even DOS 5.0 or DR-DOS 6.0. (The respective vendors of both of the latter products promote the amount of memory free as key selling points for their systems).

And that is only for default configurations. In the real world, many device drivers are needed for various applications, whether for memory (expanded or extended), connectivity (LAN, 3270), or mouse. In OS/2 2.0 these features are supported without taking precious conventional memory away, whereas, even in systems like DOS 5.0 that can move certain device drivers into the HMA area, there is nearly always a conventional memory impact. The following table shows some sample configurations which illustrate this point vividly:

Table 5. Memory comparison: OS/2 2.0, DOS 5.0, Windows 3.1 enhanced mode

	OS/2 2.0	**DOS 5.0**	**Windows 3.1**
Conventional DOS memory after default install	633KB	622KB	577KB
Various scenarios, built on default configuration:			
CONFIG #1 - POPULAR FEATURES			
file I/O	YES	1	1
EMS	YES	8	8
mouse	YES	14	14
fast file access	YES	20	20
REMAINING APPLICATION MEMORY	633KB	579KB	534KB
CONFIG #2 - LAN CONNECTIVITY			
LAN adapter drivers and LAN requester	YES	100	100
REMAINING APPLICATION MEMORY	633KB	522KB	477KB
CONFIG #3 - 3270 CONNECTIVITY			
3270 adapter drivers and emulator	YES	28	28
REMAINING APPLICATION MEMORY	633KB	594KB	549KB
CONFIG #4 - NLS PACKAGE/COUNTRY SUPPORT			
NLS keyboard	YES	7	7
NLS display	YES	8	8
NLS printer	YES	11	11
NLS other (eg codepage)	YES	5 (min)	5 (min)
REMAINING APPLICATION MEMORY	633KB	591KB	546KB

Of course, the above configurations are by no means mutually exclusive: it will be very common to have elements from at least two of the configuration groups above. Indeed, groups 1, 2 and 3 together could be considered a fairly essential configuration for a corporate, connected workstation, running any modern DOS applications. That means the difference between OS/2 and DOS grows greater. Note also, of course, that because Windows 3.1 and 3.0 are DOS extenders, and run on top of DOS, that any reductions in the memory available for DOS will be passed on to Windows, and to any DOS applications executing under Windows. While it may not be a problem for Windows applications running in standard or enhanced mode, which use extended memory anyway, it will substantially reduce the effectiveness of most DOS applications running under Windows because of the constraints on conventional memory.

The reason OS/2 can provide all these features without loss of DOS application space is because many features are supported by OS/2 device drivers, which are available to OS/2 and DOS applications, and which do not affect the amount of application space available to DOS applications. Only the DOS emulation kernel resides below 1MB. In this way, the MVDM architecture makes available to DOS applications the maximum amount of conventional memory.

In fact, if that were not enough, it is possible to increase DOS application space still further. One of the DOS settings is VIDEO_MODE_RESTRICTION (see "DOS Settings" on page 41). This frees much of the video buffer space above the 640KB line by restricting applications to text or CGA graphics mode only, allowing more than 720KB memory free - an unheard of figure for DOS users!

This increased size of conventional memory space may even mean that, where DOS users could not load TSR programs with larger DOS applications, they can under OS/2! Furthermore, there are many scenarios encountered by Windows 3.x users today, where they require access to a network, and a Windows application, at the same time as using a DOS application that requires 590K or more memory (the latter is not uncommon with the more modern DOS applications). This combination is usually not possible under Windows 3.x, and the user has to exit Windows to run the DOS application, or remove network support. Under OS/2 2.0, this setup can be achieved easily.

And, of course, more conventional memory means more space for data and, for some DOS applications, better performance, making OS/2 a better DOS environment than DOS itself, by overcoming one of DOS′s most fundamental limitations - memory.

Expanded and Extended Memory

OS/2 also supports the full range of other memory types supported under DOS and Windows 3.x: HMA, XMS, expanded and extended memory and DPMI (see "Memory: Conventional, Expanded, Extended" on page 18 for an explanation of these memory types). This means that not only do all DOS applications have more conventional memory available, but those which have been written to access more memory can work in the same way under OS/2 2.0.

For example, expanded memory conforms to the LIM 4.0 specification, and will provide expanded memory to applications like Lotus 1-2-3 release 2 using the standard DOS INT 67h services. Each VDM is provided with a separate EMS emulation, so each can access as much expanded memory as necessary, and not conflict with each other. The amount of expanded memory is configurable by the user in DOS settings, as well as a limit set across all VDMs if required. In Windows 3.x, expanded memory can only be used if a physical expanded memory adapter is in the machine, therefore limiting the number of users who can mix applications needing expanded and extended memory.

XMS support (LIMA version 2.0 level) is provided via a virtual device driver (the same is true of EMS), and manages three types of memory: High Memory Area (HMA), Upper Memory Blocks (UMBs) in the Upper Memory Area (UMA) and Extended Memory Blocks (EMBs). These are used by various DOS applications and TSRs, and by Windows 3.x and DOS 5.0. Just as DOS 5.0 can load part of the system into the HMA, and device drivers into UMBs, so can OS/2 2.0 via the statements DOS=HIGH and DEVICEHIGH= in CONFIG.SYS; these settings can also be made for an individual VDM by changing the DOS Settings DOS_HIGH and DOS_UMB (see "DOS Settings" on page 41). Furthermore, more extended memory can be provided to DOS applications under OS/2. DPMI-compliant applications such as Lotus 1-2-3 version 3.1+ can access up to 512MB of extended memory in an OS/2 VDM.

The key point is that OS/2 provides compatible services for extended and expanded memory for DOS applications, as well as a larger conventional address space, and also provides higher limits for many of the DOS applications that use extended memory. And this can usually be provided as default without any configuration effort on the user′s part.

Multiple DOS applications - effect on memory

Runing several DOS applications will not cause an excessive effect on memory use or performance. Because OS/2 operates a virtual memory system, you do not need to have extra physical memory to have large memory applications, nor to run several of them at once. Any memory required above what is physically installed will be found by using the disk as virtual memory. VDMs are swappable when inactive, reducing the overall overhead on physical memory. Furthermore, applications that do not use EMS, XMS or DPMI, can have the default VDM settings changed to remove any expanded/extended memory requirements, thus reducing the overall virtual memory required.

Reliability and protection

In DOS, most applications run in the real mode of the processor. DOS extender applications execute mainly in protected mode, but still have to switch back to real mode to perform certain I/O instructions (such as most calls to DOS services) or when passing control to TSRs like device drivers. Protected mode, as its name suggests, provides a measure of protection in the CPU between processes, but while in real mode, applications can write to any area of memory. Errors can exist even in the best-written and well tested of DOS applications, which can cause such system corruption under DOS, often leading to a system crash or hang, forcing the machine to be rebooted. Windows 3.x, as a DOS extender, can be prone to some of these problems. The problem can be particularly acute under Windows, since one application that crashes may affect all the programs running at that time. Therefore, the potential for data loss and for wasted time in resetting the system, is even greater. That is why it is crucial that a multi-tasking environment provides as much protection as is possible. There is a limit to how much can be achieved under DOS or a DOS extender.

In fact, although Windows 3.1 is claimed to provide greater protection than Windows 3.0, by “rebooting” individual DOS sessions and adding parameter checking, it is still as prone as before to the risk of a DOS TSR (such as a DOS network driver) taking the system out of protected mode into real mode (see “Reliability” on page 54). Neither Windows 3.0 nor 3.1 can prevent this “trap door” in its system integrity as long as they run on DOS. It is, in architectural terms, fundamentally prone to such problems, no matter what error checking code is implemented at higher levels.

In OS/2 2.0, each VDM emulates an entirely independent instance of DOS, and each VDM has its own separate address space (just like other processes under OS/2), applications are protected from one another, and the system is protected from applications. The VDM Manager (VDMM) within the OS/2 kernel can terminate VDMs when an application or device driver performs an illegal operation, while allowing other VDMs to continue running. While it is not impossible to crash OS/2 (no system is secure from applications designed specifically to subvert the normal means of protection), it provides a higher degree of crash protection than any alternatives running on DOS.

Therefore, OS/2 is better protected than DOS, and provides all the benefits of a DOS extender, including multi-tasking of DOS applications, without the uncertainty of using an operating system (DOS) that was never designed for multi-tasking.

Compatibility

Of course, none of the benefits of extra memory, application protection, integration, multi-tasking and so on, are of much value to users unless they can be certain that their DOS application will run. It is impossible to guarantee all applications will run, of course (there are more than 20,000 commercial applications without even counting the investment made by in-house corporate developers - an impossible testing task!). But it is worth understanding how OS/2 2.0 has been designed to provide the maximum compatibility possible in a multi-tasking environment.

Overcoming OS/2 1.3 limitations

Since OS/2 2.0 is not constrained by some of the limitations of the OS/2 1.3 DOS box, particularly with regard to more memory (some DOS applications simply did not have enough memory free to run in the OS/2 1.3 DOS Box), and support for interrupt-dependent programs like communications, it is able to offer wider compatibility for DOS applications. However, OS/2 2.0 has taken compatibility still further, and has been designed to take into account some of the aspects of DOS applications which show their single-tasking heritage.

For example, some so-called “bad” applications directly address hardware devices, and assume they have sole control over them. Many applications write directly to the video memory buffer to improve screen refresh performance. Perhaps the best known example of such an application is Lotus 1-2-3. In a multi-tasking environment, it is important to be able to handle multiple applications wanting to write to the screen at once, and maintain visual consistency on the screen, while keeping maximum compatibility with existing applications. OS/2 does this via its virtual device drivers. There is a virtual video device driver, and also a virtual COM driver, which handles contentions between applications wanting to use the COM port, and provides all DOS applications with COM services at the same time.

The video virtual device driver is of further interest in that it provides applications with fast screen I/O by allowing the foreground application to write directly to the video hardware, but still insulating the physical hardware from background VDM screen activity. It also provides services for DOS applications that use BIOS video routines, by intercepting the ROM BIOS video interrupt (INT 10h) and performing the requested operations directly, thus improving performance.

DOS device drivers

Many common device types (video, keyboard, mouse, COM, EMS, printer) are supported by virtual device drivers. If neither a virtual device driver nor OS/2 protected mode device driver is available, OS/2 2.0 can still provide support for many DOS device drivers via a “generic” DOS device support. This would usually entail exclusive access to that adapter and device driver from one single VDM. This will be more than adequate for many DOS applications, which use many device drivers for only one application anyway. And it has the great advantage of allowing the large number of DOS device drivers, for the plethora of adapters now available for PCs, to be used “as-is” under OS/2 2.0. Scarcity of protected mode device drivers was an inhibitor to adoption of OS/2 1.x; it need not be for 2.0.

IBM has demonstrated how DOS device drivers for scanners, FAX cards, MIDI adapters can all be supported, and DOS applications that depend on them work unmodified. Furthermore, there are a variety of devices in the banking and manufacturing industries that can also be supported via this approach. Even 3270 applications can be supported with the DOS device driver in this way.[4] However, a better approach would be to use an OS/2 device driver, as provided with Extended Services for OS/2, for example, which can support both DOS and OS/2 3270 applications simultaneously. This is better than running the DOS driver in a VDM, which only allows access exclusive to that VDM. A similar restriction exists in the March General Availability code, with DOS network drivers (eg for Token Ring adapter). These can often be used, but allow the adapter only to be used within a single VDM (so that, for example, DOS 3270 communications via Token Ring cannot be used concurrently with DOS-based networking through the same Token Ring adapter). This restriction can be overcome for Token Ring using the Network Transport Services/2 product (NTS/2) - see “IBM Network Transport Services/2” on page 102 running on an OS/2 2.0 base. This product provides a virtual device driver (VDD) for 802.2

[4] The Microsoft document ″Microsoft Windows/NT Operating System - An Overview″ confirms that ″...OS/2 supports some MS-DOS programs requiring some special-purpose, custom device drivers, whereas Windows NT does not. Examples are 3270 emulators, fax boards, scanners and MIDI boards.″ (p13)

access. This VDD allows 802.2 device sharing between VDMs so that, for example, Personal Communications/3270 and program using the LAN can run together.

Even so, not all DOS device drivers are supported. Some block device drivers (usually disk and tape drivers) are not supported, though some can run by booting the “real” DOS inside a VDM (see “Virtual Machine Boot (VMB)” on page 45).

The result of this is that OS/2 opens the door to an even wider range of DOS applications and devices that can run as before, but taking advantage of the OS/2 benefits described here.

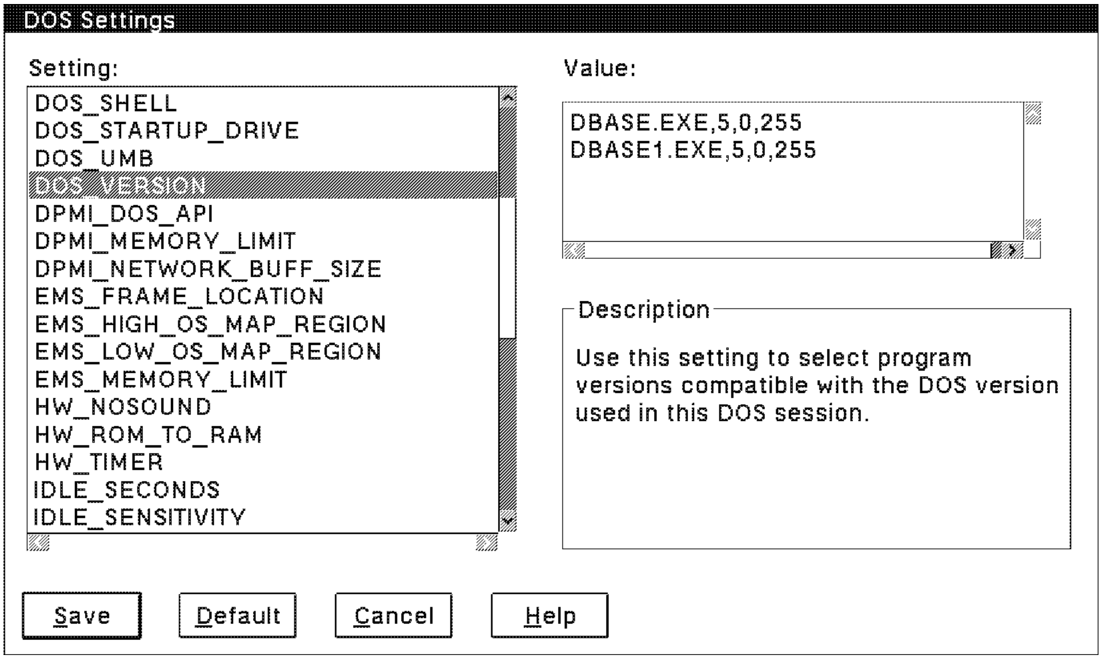
What DOS version?

OS/2′s DOS emulation is just that - an OS/2-specific DOS kernel that emulates DOS services. It is “DOS 5.0 - like,” in that it should run all applications supported by DOS 5.0.

Should applications require dependencies on a specific version of DOS there are a number of strategies available to the OS/2 2.0 user. First of all, it is possible to “fake out” the DOS version, in other words, fool the application into thinking you are using a specific version of a DOS component. This is using the DOS_VERSION setting in DOS Settings (see below). If this is not enough, it is possible to boot the real version of DOS inside that VDM only (see “Virtual Machine Boot (VMB)” on page 45).

DOS Settings

The DOS_VERSION option described above is just one of a number of “DOS Settings” that can be changed to control the environment in each VDM. These are set for that VDM only. Some settings can be made for all VDMs via AUTOEXEC.BAT and CONFIG.SYS, such as DEVICE=... statements (ANSI, VXMS, VEMM, VMOUSE, VVGA, VCOM), and some (though not all) can be changed while the application is running. Several of these settings can also have an effect on DOS compatibility, others can improve performance.



DOS Settings

Setting:

DOS_SHELL
DOS_STARTUP_DRIVE
DOS_UMB
DOS_VERSION
DPMI_DOS_API
DPMI_MEMORY_LIMIT
DPMI_NETWORK_BUFF_SIZE
EMS_FRAME_LOCATION
EMS_HIGH_OS_MAP_REGION
EMS_LOW_OS_MAP_REGION
EMS_MEMORY_LIMIT
HW_NOSOUND
HW_ROM_TO_RAM
HW_TIMER
IDLE_SECONDS
IDLE_SENSITIVITY

Value:

DBASE.EXE,5,0,255
DBASE1.EXE,5,0,255

Description

Use this setting to select program versions compatible with the DOS version used in this DOS session.

Figure 8. DOS Settings notebook

This means that each individual VDM can have its own optimised settings: some with EMS, others without; some have DOS loaded high, others have special DOS device drivers loaded in that VDM only. This gives great flexibility in configuration, and allows the user to run different DOS applications concurrently without having to maintain different CONFIG.SYS files and reboot between them.

Among the commonly used or interesting settings are:

Table 6 (Page 1 of 2). Some VDM DOS Settings

Setting	Description	Default setting
DOS_BACKGROUND_EXECUTION	When set off, suspends execution of the program when it is in the background. Useful for preventing degradation of multi-tasking performance when a single application is polling heavily for keyboard input. Rather less fine control than using IDLE_SENSITIVITY and IDLE_SECONDS, which should be attempted first (see below).	On
DOS_DEVICE	Allows DOS device drivers to be loaded in an individual VDM, rather than across all DOS sessions. It adds to the drivers specified in CONFIG.SYS, which apply to all DOS sessions. The path and name of the device driver are entered, and this driver is available only for that VDM.	Existing drivers from CONFIG SYS are listed
DOS_UMB	Gives the DOS emulation kernel use of Upper Memory Blocks, so that DOS TSRs and device drivers can be loaded into addresses between 640 and 1024KB, thus freeing memory for DOS applications below 640KB. This is similar to the DOS 5.0 function, allowing use of the DEVICEHIGH and LOADHIGH commands. The UMB ownership can be relinquished by turning this feature off, if an application program needs to manage UMBs. This setting can also be made globally in CONFIG.SYS.	On
DOS_VERSION	Allows OS/2 to report a “fake” DOS version number to a request from a program in the VDM, in order to support applications that check for a specific DOS version number. This is important for applications such as Lotus 1-2-3 version 3.1+, which look for the presence of DOS 3.3 or above. The parameters set are the name of the DOS executable (eg 123DOS.EXE) followed by the DOS major and minor version number (eg 3,30 for DOS 3.3), then the number of times to “fool” the application (255 means “every time”)	20 (ie OS/2 Version 2.0)
DPMI_MEMORY_LIMIT	Specifies the maximum amount of protected mode memory available to DPMI applications running in a VDM (in MB). It is important to set this figure high enough for a WIN-OS/2 VDM running multiple Windows applications in the same VDM.	2
IDLE_SECONDS	Disables the IDLE_SENSITIVITY (see below) setting for a period of time after “useful” work by the application has been detected. Some programs appear to be waiting for input, but then change and continue other work. The setting needs to be made high enough to allow the application to run fast enough, but not too high as to give the program more resources than it needs	0

Table 6 (Page 2 of 2). Some VDM DOS Settings

Setting	Description	Default setting
IDLE_SENSITIVITY	Sets a threshold for judging when applications will be deemed idle (ie waiting for I/O, polling etc). OS/2 2.0 can detect idle programs, especially those with a high rate of polling for input, and gives them less time to run, assigning the CPU to more "deserving" applications. This setting allows a user to modify OS/2′s "best guess" at what it considers idle. The lower the number, the more likely OS/2 will judge the application idle.	75% (refers to % of maximum possible polling rate)
MOUSE_EXCLUSIVE_ACCESS	Allows a VDM to run applications that maintain their own mouse pointers, and which manage their own mouse positions and movements, by forcing the physical mouse driver to send its movements directly to the virtual mouse driver (and therefore to the DOS application) rather than going through PM. Only one mouse pointer appears when that VDM window has the focus. Useful when running DOS applications that require use of the mouse, in windows on the Workplace Shell desktop, and prevents the situation where you can see 2 mouse pointers, one for the DOS application, and one for PM, and have difficulty synchronising them. An example of a DOS application that can use a mouse in a VDM is WordPerfect 5.1 for DOS.	Off
PRINT_TIMEOUT	Sets the time in seconds after which the spooler will close a print job initiated by the VDM, and begin to print. This means that printing can begin from DOS applications that do not close their print jobs, without having to exit the DOS application	15
VIDEO_FASTPASTE	Improves the speed of paste operations from the clipboard to a DOS application. Not all applications can support this.	Off
VIDEO_MODE_RESTRICTION	Extends DOS address space beyond 640KB by limiting video mode support to CGA, providing up to 96KB (depending on video adapter installed) extra for DOS applications	None

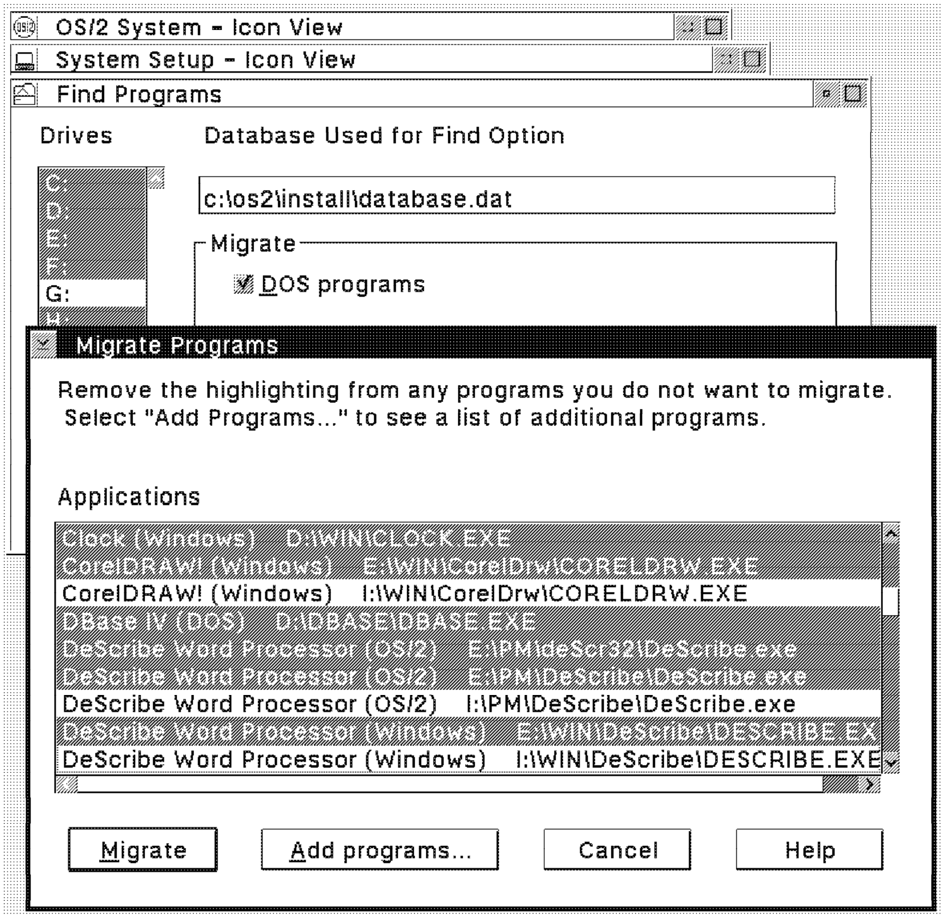
Help is provided with OS/2 2.0 to guide users in modifying these settings, but it should be noted that much of the time, DOS settings will neither need to be examined nor changed. The vast majority of applications will work with the default settings. Many of the most common DOS applications can have their settings automatically created by the Migrate program (see below).

Migrating applications

OS/2 2.0 provides a migration database (DATABASE.DAT) that contains parameters and settings for commonly used DOS, Windows and OS/2 programs. This binary database file is used by the **Migrate Applications** object to place the program icons onto the desktop and customise their DOS or WIN-OS/2 settings to the recommended values (which have been determined during pre-release testing).

This means that for many common DOS and Windows programs, there will be no need to modify DOS settings, since the Migrate program will automatically place a program object on the desktop with the optimal settings. These are placed in a "DOS Programs" folder and a "Windows programs" folder as appropriate. Migrate also allows applications not in the migration database to be migrated, but with the default DOS settings, which can then be modified manually if preferred. These programs are placed in folders called "Additional DOS programs" and "Additional Windows programs" as appropriate. All folders created by Migrate are given the appearance of the group icons used in Windows 3.x and OS/2 1.3, for visual familiarity. This aim is furthered by the fact that Migrate detects the presence of the program icon for a Windows application, and adds it to the program reference it creates in the OS/2 folder. This means that Windows applications appear in OS/2 with their normal icon. DOS applications are given the standard DOS icon for full screen or windowed sessions as appropriate. The settings for each object can be modified to add a user-defined icon as necessary. Many tools, such as CVTICO, exist on bulletin board systems, to convert icons from Windows format to OS/2 format, so that any public-domain icons that exist in Windows format for common DOS applications, may be used in OS/2.

Migrate can be run during the install process or at any time after (the Migrate object is found in the System Setup folder).



Database Used for Find Option: c:\os2\install\database.dat

Remove the highlighting from any programs you do not want to migrate. Select "Add Programs..." to see a list of additional programs.

Applications

Clock (Windows)   D:\WIN\CLOCK.EXE
CorelDRAW! (Windows)   E:\WIN\CorelDrw\CORELDRW.EXE
CorelDRAW! (Windows)   I:\WIN\CorelDrw\CORELDRW.EXE
DBase IV (DOS)   D:\DBASE\DBASE.EXE
DeScribe Word Processor (OS/2)   E:\PM\deScr32\DeScribe.exe
DeScribe Word Processor (OS/2)   E:\PM\DeScribe\DeScribe.exe
DeScribe Word Processor (OS/2)   I:\PM\DeScribe\DeScribe.exe
DeScribe Word Processor (Windows)   E:\WIN\DeScribe\DESCRIBE.EX
DeScribe Word Processor (Windows)   I:\WIN\DeScribe\DESCRIBE.EXE

Figure 9. Migrate Applications object

System Administrators can set up their own custom migration database, to set up applications unique to their environment, using the PARSEDB tool supplied with OS/2 2.0. Thus, an optimal collection of DOS Settings can be determined for a given program, and then supplied to other users.

Virtual Machine Boot (VMB)

Should neither the basic DOS emulation of OS/2 2.0 nor any DOS settings provide an answer to compatibility problems with a given DOS application, the ultimate recourse is to boot the real version of DOS on which it depends, in a VDM. This can be done either from the DOS boot diskette or from an image file created on the hard disk. This applies not only to PC-DOS and MS-DOS versions, but also non-IBM systems such as DR-DOS (and in theory any 8086 operating system kernel). A VMDISK utility is supplied with OS/2 to create disk images.

Once booted, the VDM is running that real copy of DOS, and the boot image path or drive becomes the A: drive for that VDM.

The VMB feature, by booting the “real” DOS, provides the maximum achievable compatibility in a multi-tasking environment.

Even so, there will be some architectural compatibility issues that even VMB cannot address. Most of these are limitations that arise from application or device driver features that are fundamentally at odds with the principles of a multi-tasking system. These issues include VDM interrupt latency (which may prevent some DOS “real-time” applications from giving sufficient performance), support for VCPI DOS extender applications (see below), and I/O to system-managed DASD that bypasses the file system (the latter is prohibited for obvious reasons - this could pose a threat to the integrity of the whole file system and affect other applications.) But these are fairly minor restrictions in an immensely wide scope of compatibility.

DOS extenders

OS/2 2.0 can support DOS extender applications, those applications that have broken the 640KB limit by running in protected mode and including their own memory manager which “takes over” from DOS′s. These include DOS multi-tasking environments such as DESQview or Windows 3.x, as well as applications that include DOS extender code, such as 1-2-3 3.1+.

OS/2 2.0 supports the DPMI (DOS Protected Mode Interface) specification for extender applications, allowing applications written to this specification to run under OS/2 2.0 and access up to 512MB of extended memory (see the description of DPMI in “Definitions” on page 18). OS/2 2.0 is a DPMI ***host***, which provides DPMI services to DOS extender applications (DPMI ***clients***).

VCPI applications will not run under OS/2 2.0 because they are architecturally incompatible with a multi-tasking operating system (see “Definitions” on page 18). VCPI applications do not run under Windows 3.0 or 3.1 either. However, many VCPI applications have been rewritten to support DPMI, in order to support Windows 3, and OS/2 2.0 can take advantage of all of these.

Better Windows

One of the most popular DOS extenders is Microsoft Windows. Windows 3.0 shipped in May 1990, and gained a lot of media attention in the market. In April 1992, Microsoft shipped an updated version, Windows 3.1, which was originally described by Microsoft as a “maintenance” update, but has added a number of new features, to address some of Windows 3.0′s limitations. In this chapter, most discussion will refer to both Windows 3.0 and 3.1 together as Windows 3.x, since, except for some details, they compare to OS/2 2.0 in roughly the same way.

Though Windows has undoubtedly had a significant impact on the market, there has been considerable speculation in the press about how successful it has been. Microsoft claims that over ten million copies of Windows 3.0 were sold up to March 1992 (although IDC estimate that more than half of these may have been shipped with the purchase of a new machine, of which they and Creative Strategies estimate only 30 and 55% respectively, are in use). The company also said that it shipped 3 million copies of Windows 3.1 in the first three months after its shipment. However, in a report in the *Wall Street Journal* in August, a spokesperson for the software retailer, Egghead, commented that the preloading of systems like Windows with hardware, had affected retail sales. She said that the company overstocked Windows 3.1 in anticipation of a heavy retail demand for the system that failed to materialize. ″We bought strong quantities of Windows 3.1 and they′ve been selling at a slower rate than we anticipated,″ she said.[5]

Windows Magazine, in its October 92 issue, estimated that the “real situation” with regards to the number of Windows users was around 4.5 million. A recent Dataquest study estimated that as of September 1992, 18% of the PCs in the US are running Windows, which equates to around 2.6 million users. This concurs with figures quoted in *OS/2 Magazine* from ADAPSO/ITAA, saying that Windows has penetrated only just over 10% of the total DOS market. Whatever figures are used, *Windows Magazine* says that “financial analysts and software companies...uniformly dismiss [Microsoft′s figure of 13 million Windows users] as meaningless.”[6]

There have also been a number of ISVs developing applications for Windows. According to Microsoft, over 4,500 applications had been developed for the Windows environment by March 1992.

Windows 3.x is, of course, in one sense an alternative or competitive solution, since it provides a GUI and a basic multi-tasking environment. On the other hand, some customers require IBM to support the Windows applications that they have purchased, under OS/2 2.0. This section examines OS/2′s relationship to Windows 3.x, and how OS/2 can be a “better Windows.”

What is MS-Windows?

First of all, it is important to understand what Windows is. When it was first released in 1985, it provided a GUI environment running under DOS, for which applications had to be specially written to take full advantage. Although versions before version 3.0 had gained some developer support, market success was comparatively limited, and many of the major ISVs (eg Lotus, WordPerfect, Borland) had not developed Windows-specific applications. Some of the reasons for this were perhaps the lack of Windows-capable machines, the limited multi-tasking, and the fact that it provided only limited relief for the two main perceived limitations of DOS: the 640KB memory barrier, and multi-tasking the DOS applications that people already had. Windows 3.0 overcame many of these problems, especially in

[5] Quoted in *Wall Street Journal*, 6th August, 1992

[6] *Windows Magazine*, October 1992, p16

being able to multi-task DOS applications and allowing Windows itself to take advantage of extended memory.

Windows 3.x modes

The Windows environment itself has three modes, though the options available to any user depend upon machine processor and installed memory. These modes are important to mention here as they are relevant to the discussion later of how OS/2 runs Windows applications. The italicised description of the three modes comes from the MS-Windows User′s Guide:

Real mode (NOT AVAILABLE IN WINDOWS 3.1)

An operating mode that Windows runs in to provide maximum compatibility with versions of Windows applications prior to 3.0. Real mode is the only mode available for computers with less than 1MB of extended memory.

Real Mode is equivalent to previous versions of Windows (2.x), and can address 640KB conventional memory, plus LIM 4.0 expanded memory (extended memory can be used for a virtual disk or disk-caching only). Real Mode requires an 8088 processor or above, and 640KB memory (384KB free conventional memory after DOS and other memory resident software including network drivers).

Microsoft withdrew support for real mode in Windows 3.1, so that only standard mode and 386 enhanced mode are now available. This means that programs requiring Windows real mode (eg programs written for Windows 2.x) will not run in Windows 3.1. However, Windows real mode programs (including Windows 2.x applications) ***WILL*** run under OS/2 2.0. In this respect, OS/2 2.0 is providing better support for Windows programs than Windows 3.1 itself.

Note that although Windows real mode runs in the real mode of the processor (see “Protected mode operation” on page 14), the two are not identical terms; even though Windows 3.1 has eliminated ***Windows*** real mode, it cannot by its very design prevent the system moving out of protected mode into the real mode of the ***processor***, and this represents a potential hole in system integrity (see “Reliability and protection” on page 39 and “Reliability” on page 54 for more on this subject).

Standard mode

The normal operating mode for running Windows. This mode provides access to extended memory and also lets you switch among non-Windows applications.

Standard Mode uses 286 protect-mode to give Windows and Windows applications direct access to up to 16MB extended memory. Expanded memory for DOS applications is only supported with physical expanded memory cards (not emulation of expanded memory using extended memory). Standard Mode requires a 286 processor or above, 1MB memory (minimum 256KB of free extended memory), and the XMS driver HIMEM.SYS loaded. Windows applications need to be written to comply with the memory management rules for Windows 3.x to run in Standard Mode.

Standard mode is recommended by Microsoft when running only Windows applications (ie no DOS applications) in certain configurations, even on a 386.

386 Enhanced mode

A mode that Windows runs in to access the virtual memory capabilities of the Intel 80386 processor. This mode allows Windows to use more memory than is physically available and to provide multi-tasking for non-Windows applications.

386 Enhanced Mode uses protected mode to give Windows and Windows applications direct access to up to 16MB extended memory. The Virtual-8086 mode of the 386 is used to provide multiple DOS environments for non-Windows applications, and virtual memory support is provided (for Windows applications only) using the demand paging feature of the 386 processor. Most DOS applications can be run in a window. 386 Enhanced Mode requires a 386 processor or above, 2MB memory (minimum 1024KB of free extended memory), and the XMS driver HIMEM.SYS loaded. Windows applications need to be written to comply with the memory management rules for Windows 3.x to run in 386 Enhanced Mode.

Windows 3.x key aspects

Windows 3.0 added a number of extra features to previous Windows releases, but many people believe that its three main features are as follows:

DOS extender (memory > 640KB)

Windows 3.x is first and foremost a DOS extender. It is itself a DPMI host application running on top of DOS, which provides applications (in this case Windows applications written specially for that environment) with extended memory support to break the 640KB DOS limit.

Windows can provide additional memory for applications that have been written to take advantage of this capability. But the vast majority of applications still are written for the DOS environment, and even under Windows can only address up to 640KB. In fact, as we saw in our discussion on memory available under OS/2 2.0 and Windows earlier, since Windows inherits the DOS environment for each DOS application it runs, it provides less memory in most scenarios than would be available under OS/2 2.0, for the same DOS application in the same configuration.

Nevertheless, since Windows, in standard and enhanced modes, runs much of the time in protected mode (requiring a 286 or 386 processor for these modes), it can itself take advantage of extended memory to run, leaving conventional memory free for DOS applications (but at the cost of some overhead in conventional memory for Windows itself - see “Comparison with memory usage under DOS” on page 36).

Multi-DOS environment

Windows 3.x also offers, in its 386 enhanced mode, multi-tasking of DOS applications. Several DOS applications can be run at once, and most text-based applications can appear in a window alongside other DOS or Windows applications, in 386 enhanced mode. However, Windows 3.0 enhanced mode does not offer as flexible support as OS/2 2.0 does, for running graphics-based applications - like Microsoft Flight Simulator - in windows on the screen alongside text-based applications. DOS applications running in VGA mode can be displayed in a window on the Windows 3.1 desktop, but according to Microsoft, Windows/NT will not support windowed DOS VGA graphics at all.[7]

In enhanced mode, Windows also provides a pre-emptive time-slicing mechanism to multi-task between these DOS applications, though, as we saw before (“Multi-tasking of DOS applications” on page 35), OS/2 2.0′s mechanism is more sophisticated and will tend to provide a more balanced

[7] See Microsoft′s document entitled ″Microsoft Windows NT Operating System″

performance when multi-tasking DOS applications. For many users of 386 machines running on a DOS base, Windows enhanced mode provides welcome additional function, and the ability to swap between their DOS applications. Standard mode allows multiple DOS applications to be loaded, but they do not run in background nor can they be windowed.

Windows applications

Of course, there have also been a number of applications written specifically for Windows 3.x. These are graphical applications written to the Windows APIs, and can take advantage of extended memory (up to 16MB), the GUI environment, as well as Windows system-supported functions like DDE (Dynamic Data Exchange). Although the Windows environment was ignored by many of the major vendors at first, there are now many applications developed specifically for Windows 3.x. Applications developed for previous releases of Windows cannot run reliably under the two higher modes of Windows, but only under real mode. This means that if you are running Windows 3.0 in standard or enhanced mode, and wish to run, for example Excel 2.1, you must close all your Windows and DOS applications, exit Windows and restart in real mode. Windows 3.1, because it does not support real mode, will not run Windows 2.x applications at all. OS/2 2.0 can run real and standard mode Windows applications side by side on the Workplace Shell desktop.

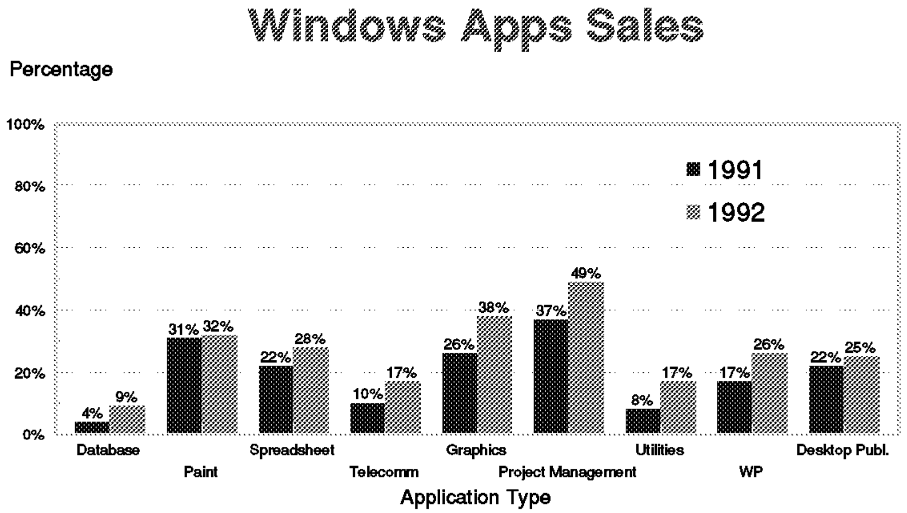
Although Microsoft claims that there are over 5,000 Windows applications in the market, the majority of applications sales are still in DOS applications (to say nothing of the installed base). There are many items of evidence to back up this statement:

- Sales of DOS applications were still $112 million more than those of Windows applications in the second quarter of 1992 (Software Publishers′ Association - SPA)[8]
- In figures quoted in *Windows Magazine*, based on projections by Dataquest, Windows applications still represent less than 25% of the market in most application categories, and never as much as half in any (see the graph below which illustrates these figures in detail)[9]
- Even in 1997, DOS applications will outsell Windows applications (Dataquest)[10]

[8] SPA figures quoted in *Windows Magazine*, November 1992

[9] Quoted in *Windows Magazine*, February 1992

[10] Quoted in *Computing* (UK), April 23, 1992

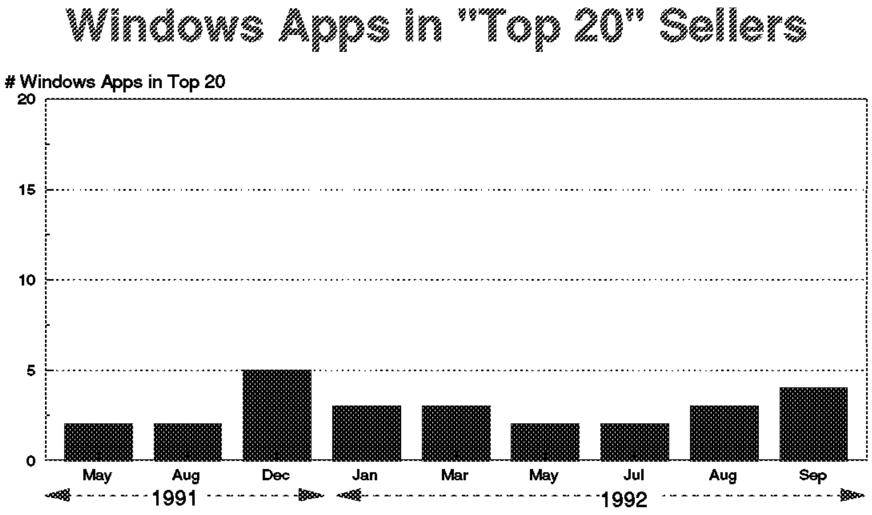


IN MOST PRODUCT CATEGORIES, WINDOWS APPS ARE LESS THAN 25% OF MARKET

Source: Windows Magazine, based on projections by Dataquest

Figure 10. Windows application sales as a proportion of total PC application sales

- Many PC publications both in Europe and the US track sales of applications from a variety of channels, in order to determine a “Top 20” list of applications. The trend since May 1990 is clear: despite the apparent success of Windows in the market, few Windows applications make it into the Top 20, as the graph below shows. In fact, there are rarely more than 5 Windows applications in the top 20, and only two of them regularly appear - Excel for Windows and Word for Windows by Microsoft.



NEVER MORE THAN 5 IN THE TOP 20

Source: PC Magazine. PC Sources. PC User

Figure 11. Number of Windows applications in the top 20 selling applications

- There have also been several news items showing how the Windows applications market may not be as large as some vendors may have suggested. Here are some references from recent reports:
  - "The smaller software companies are not making money out of Windows, and Windows 3.1 is not going to change that" (International Data Corp.)[11]
  - "The Windows market is really only the size of the Macintosh market" (Gordon Eubanks, CEO of Symantec, who produce Norton Desktop for Windows, one of the best selling Windows applications)[12]
  - Lotus warned analysts of lower than expected revenues in June 1992, blaming "weakness in the market for Windows spreadsheets rather than loss of [market] share."[13] "We′re disappointed that the overall Windows apps market didn′t explode like the channel thought it would" (Robert Weiler, Senior VP, Lotus)[14]
  - Lotus is not the only major Windows vendor to suffer falls in share prices owing to lower sales of Windows applications: Aldus reported lower earnings at the end of June 1992, owing to, among other reasons, "lower than expected sales of its Windows-based products."[15]
  - Ruthann Quindlen, a software industry analyst, was quoted in an interview with *Infoworld*′s editor, that she "regret[ted] saying... that Windows was going to create opportunities for small PC software companies." Listing many software companies suffering from slowing growth, earnings shortfalls, falling stock prices, she "had hoped that Windows would grow the market...But no." She concluded that "the only Windows apps doing well are Microsoft′s"[16]
  - "Companies that have invested a lot of money in developing Windows applications are battling for a small share of what is a very small pie" (*Personal Computer Magazine* (UK), May 1992)
- Windows applications are still heavily outnumbered by the 20,000 or more DOS applications, which also tend to have been longer in the market and built up a substantial installed base.

In fact, there is also much evidence to suggest that Windows itself is outselling Windows applications by a factor of three or four to one:

- in the year to March 1992, when Microsoft claimed there were ten million copies of Windows shipped, IDC estimated 3.5 million Windows applications were sold.
- In early 1992, Microsoft stated that there had been $1.6 billion of Windows applications sold. If we take an average price of $500, this would give just over three million copies of all Windows applications together.
- In October 1991, Windows annual run rate, according to Microsoft, was 7.8 million. The SPA′s estimate of the volume of Windows application sales for 1991 was $1000 million. This could mean that for every copy of Windows shipped, there was just over $120 spent on Windows applications, or about one quarter of a copy of Lotus 1-2-3 for Windows at its current recommended retail price!

If Windows were being used primarily to run Windows applications, one might expect the user to have more than one Windows application. On this assumption, the number of Windows application users may be only 15% or less of the total of Windows users. Either many people have bought Windows and are not using it, or they are using it for something else.

A study of *PC Magazine* subscribers indicates a possible answer. The study was conducted in the US by Ziff Davis Research Department and Alpert Research, Inc., of 1840 *PC Magazine* readers. The study

[11] Quoted in *PC Week*, March 30, 1992 - p131

[12] Quoted in *PC Week*, March 30, 1992 - p132

[13] Reported in *Computergram Online* Issue #1952 - 29th June 1992

[14] Quoted in *PC Week*, March 30, 1992 - p132

[15] Reported in *Computergram Online* Issue #1949 - 24th June 1992

[16] Article in *Infoworld*, September 14th, 1992 - p51

has been reviewed by William Zachmann, a well known independent commentator on the industry, in the June 11, 1991 issue of *PC Magazine* (Volume 10, Number 11, Page 97). Zachmann reported that respondents who personally used or were familiar with Windows, confirmed that the most desirable features of Windows were its memory management, multi-tasking and extended memory capabilities. GUI capabilities, and the availability of Windows applications were of only secondary importance.

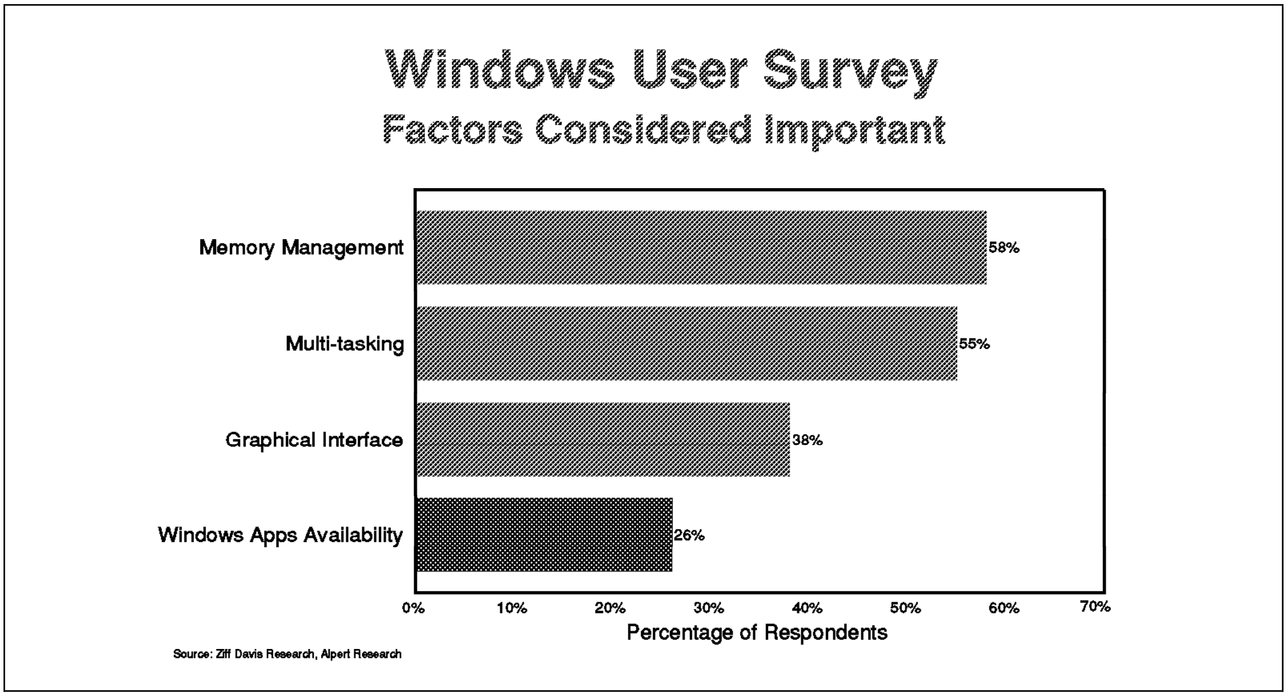


Figure 12. Windows user survey quoted in PC Magazine June 11, 1991 issue

Zachmann's conclusion was:

> *"All of this strongly supports my contention that Windows 3.0's success is due more to its usefulness as a reasonably good DOS memory manager and multi-tasker, than to any groundswell of support for GUIs and Windows applications. The study results also corroborate my observation that sales of Windows applications lag far behind reported sales of Windows itself."*

Many observers agree with Zachmann, that it has been the combination of DOS extender and basic DOS multi-tasking, implemented on a DOS base, that has been the principal factor in Windows' success. This indicates that Windows is being used more as a DOS extender and multi-tasker than as an environment to run Windows applications. Even vendors such as Lotus, who are developing a suite of applications for Windows, have claimed that Windows is being used to run DOS applications more than it is used for Windows applications.

This is an important consideration when we look at what it means to be a better Windows than Windows. It seems as if many users look to Windows as a DOS environment first, and that Windows applications are currently a secondary issue. Therefore, to be a better Windows than Windows, OS/2 2.0 needs to:

- first, be a better DOS extender/multi-tasker
- and run Windows applications, first ensuring compatibility and, if possible, some advantages over running it under Windows.

As it happens, OS/2 2.0 is able to do both: create a better DOS environment for DOS applications, as well as provide full compatibility for Windows applications, and even some advantages in that respect.

OS/2 versus Windows as a multi-DOS environment

It is clear that any discussion of Windows cannot ignore its DOS heritage. In fact, this is both its great advantage, but also the source of most of its shortcomings. Windows runs under DOS, which is an advantage from the point of view of migration from DOS, but is also a disadvantage in the limitations that DOS imposes on it.

Memory

For example, we have already seen that since each DOS session created in Windows inherits the whole DOS environment in place before Windows loaded, DOS programs may have limited space in which to load. Although the combination of DOS 5.0 and Windows 3.x helps to relieve some of the pressure on conventional memory, most DOS applications have significantly less memory under Windows than they do under OS/2 2.0 (see “Comparison with memory usage under DOS” on page 36 for details.)

Reliability

DOS was designed as a single-tasking environment. Although DOS extenders like Windows can provide multi-tasking under DOS, the base system was never designed for it, and many of the applications running under such an environment assume they are the only application in the system. DOS runs in real mode, and does not provide protection between processes. Windows provides this protection instead, allowing a measure of multi-tasking. However, Windows still needs to use DOS for certain system functions, and so do many of the DOS-based TSRs (eg Network and Communications drivers) that are needed to run alongside Windows, and hence real mode is difficult to avoid when running Windows 3.x. Any time that is spent in real mode is a potential “trap door” for application errors to violate the system′s integrity. Even though DOS applications run in separate address spaces in enhanced mode, DOS and DOS TSRs still require access to real mode, and the potential remains for an application conflict or bad pointer to corrupt the DOS system area or Windows itself.

Furthermore, Windows applications all run in the same address space under Windows 3.x, and therefore it is possible for an error in a Windows program, to take down all the Windows applications together, and sometimes the whole system, thereby affecting the DOS applications as well. These symptoms occur in a variety of ways, not always a “crash” or “hang” but sometimes messages like “This application has violated system integrity” or “Unrecoverable Application Error.”

Windows 3.1 is claimed by Microsoft to provide greater protection from these errors, by checking the parameters passed by applications, The users of Windows 3.1 will judge how effective this will prove, but it is a significant technical challenge to provide full protection under DOS, unless an attempt is made to eliminate real mode (of the processor, not Windows real mode) execution altogether. Indeed, although the message “Unrecoverable Application Error” no longer appears, it has been replaced in many scenarios by “Application Error” or “General Protection Fault.” In some circumstances, Microsoft still advise that data in other sessions be saved, but then to exit and restart Windows, perhaps even reboot. In this case, the effects on a user would be little less drastic than before. In fact, though some have claimed that Windows 3.1 heralds “the demise of the UAE,” the essential aspects of the Windows architecture remain unchanged. Windows applications still execute in the same address space, still sharing a common Local Descriptor Table, or LDT (this is why parameter checking is needed, but it only attempts to catch problems to which the basic architecture leaves the system prone). Parameter checking is much more important to Windows than it is for OS/2, which has other mechanisms (notably separate address spaces) to protect the system from wayward applications.

Furthermore, although many I/O and other functions are taken over by Windows (leading Microsoft to call Windows 3.1 an “operating system”[17]), DOS and DOS TSRs are still available (and indeed required if you want to run Windows on a network, or with host communications), so the system still does not close the real mode “trap door.”

Any access to real mode is a potential danger; this is true even in OS/2 1.3, where the DOS compatibility box ran in real mode and was therefore a potential danger to system integrity. The only certain way to high reliability is to run the operating system only in protected mode. No current version of DOS can do this, but OS/2 2.0 runs entirely in protected mode: even when running DOS applications, there is no real mode operation.

OS/2 has been designed for high reliability from the start, and version 1.3 is already widely respected for its industrial strength design. In OS/2 2.0, all applications, both DOS and OS/2, and even Windows applications, run in separate process address spaces, protected from each other, and much care has been taken in the design to avoid giving wayward applications the potential to corrupt any other applications or the system. Usually in OS/2 2.0, the worst a “bad” application can do is to cause an error that will cause the kernel to have that process stopped and closed. In these circumstances, the other applications continue running unaffected. This is why OS/2 has already been chosen by many companies as the development environment for line of business, mission critical applications. If your business depends on your application, you do not want to see it brought down with a UAE.

Some have tried to show that OS/2 can be “broken” by wayward applications, but no operating system is safe from code written with inside knowledge of the system internals, designed specifically to manipulate parts of the system in a destructive way that mainstream applications would never attempt. The added protection OS/2 offers compared to other DOS extender alternatives, is against the occasional bugs in applications that cause them to write to memory where they should not. The issue is not really whether either environment is totally “crash-proof” but the overall integrity of the operating system in day-to-day usage. In this case, we are not really comparing OS/2 to Windows, but OS/2 to DOS.

Performance

OS/2′s multi-tasking design is more sophisticated than Windows, allowing DOS applications to multi-task more smoothly. Windows has only a simple time-slicing algorithm, which can be adjusted by the user to give “bias” to certain applications. This means that priority levels are static, and do not take into account, for example, when applications are I/O bound. On the other hand, OS/2 can set priority levels dynamically, detecting when applications are “idle” through polling or waiting for I/O, and assigning CPU cycles to other applications. This advanced scheduler works in a similar way with OS/2 applications, pre-emptively multi-tasking them alongside the DOS applications. Even Windows applications running in VDMs take advantage of this superior scheduling mechanism for smoother multi-tasking. Windows 3.x offers only co-operative multi-tasking between Windows applications, but pre-emptive with respect to the rest of the system. This means that applications have to be specifically written to give the processor up, with yield() calls, to allow other Windows applications to have their fair share of the processor. (See “Multi-tasking of DOS applications” on page 35 and “Multi-tasking” on page 25 for a more complete discussion of this topic.) In OS/2 2.0, Windows applications can be pre-emptively multi-tasked by running them in separate VDMs.

OS/2 offers a choice of two superior file systems, which DOS applications running under OS/2 use whenever they do file I/O. OS/2 2.0 has the High Performance File System (HPFS) and an enhanced FAT file system, both of which provide superior performance than under DOS in nearly all situations. This means that in file-based operations, OS/2 will nearly always be quicker than DOS or Windows running the same combination of applications under the normal DOS-based FAT system. Furthermore,

[17] Though how this “operating system” is able to load without DOS, has not been made clear by Microsoft.

OS/2 is able to overlap I/O requests to the system, taking advantage of its own multi-tasking design, allowing the system to be more responsive to I/O request and keep applications waiting less.

The result of this is that the same combination of multiple DOS applications is likely to run faster under OS/2 than under either DOS or any DOS extender like Windows, if it does any significant disk-based I/O. Applications that are more compute-bound may show less difference (and in single tasking, DOS itself is obviously likely to be faster in all except file-intensive operations).

And when more than one task is being done, OS/2′s performance advantage becomes evident. Because of OS/2′s superior multi-tasking, it can run background tasks, such as file copying, communications, or spreadsheet recalculation, with no visible impact on foreground work. With Windows, the cursor movement can lag behind the mouse movement, and displaying of characters can lag behind keyboarding to the point where system becomes almost unusable until the background job is done.

Here is an illustrative scenario from the testing of National Software Testing Laboratories (NSTL), and independent testing and evaluation organisation: to load MS Word for Windows on a PS/2 Model 57 with 8MB, with nothing else running takes 7.2 seconds with Windows 3.1 and 9.3 seconds with OS/2 2.0. If you do the same load with an XCOPY in the background, Windows load time jumps to 41.1 seconds, compared with 15.3 seconds for OS/2.

Again, the comparison is really between OS/2 and DOS - Windows simply runs on DOS and inherits its limitations.

Integration

One of the other big differences between the DOS compatibility provided by OS/2 compared to that of Windows is in the area of how easy it is to set up DOS applications to work. There are many ways in which DOS applications integrate much more easily into the OS/2 environment than they do under Windows:

- ***DOS Settings vs PIF files:*** Many have commented that Windows′s method of assigning special settings to a DOS program via a Program Information File (PIF) is cumbersome. This is even more so compared to the consistency of the way the DOS Settings option is implemented in OS/2 2.0. Compare, for example, WordPerfect 5.1 for DOS. Both Windows 3.1 and OS/2 2.0 detect the program′s presence on the disk, and create a unique set of program settings. But with OS/2 2.0, all of these settings are easily found together by clicking on the settings of the program object′s icon. In Windows, clicking on the icon in Program Manager allows you to select its basic properties from the File Properties menu in Program Manager. But the dialog allows you only to change the command invoked, the working directory and icon. To change other options, you need to find the PIF file referenced in File Properties and edit the PIF file with the PIF editor (which by default is also in another program group). The PIF file contains all the other settings such as video mode, EMS/XMS settings, priority and idle detection, but also confusingly has options for startup directory and shortcut key, which are equivalent to (but overridden by) the settings in File Properties in Program Manager. So there are two places to look for settings, and sometimes two places to set the same setting. Furthermore, some important options are missing altogether from the PIF or Program Manager; mouse access in a windowed DOS application, for example, can only be achieved by first loading the DOS mouse driver (which is distinct from the Windows mouse driver - you need two mouse drivers if you want to use the mouse for both Windows and DOS applications) before starting the application. Searching on “mouse” in the Windows help system does not yield this information. OS/2 allows the user to find all customisation for a given program in one settings notebook: command, working directory, icon, EMS, mouse access and so on.
- ***Multiple working combinations:*** The amount of memory available when running a DOS program in Windows 3.1 enhanced mode is at most about 580K, and can be considerably less than 500K in a LAN-connected environment. And the more features that are required (eg 3270 access, country

settings) means less memory for DOS sessions. This can mean that larger DOS programs, including many newer DOS programs or latest versions, do not have enough memory to run. Thus, users are faced with having to run different combinations of applications: DOS program plus network but not with Windows; a Windows program with the network, but not the DOS application, and so on. Some users even have to maintain different CONFIG.SYS and AUTOEXEC.BAT combinations, rebooting between them according to the combination of applications they are running. Many users also have to buy third party utilities such as QEMM, and experiment with various settings, to get the amount of memory they need. Use of such utilities sometimes creates new integration issues, such as getting memory managers, disk cache programs and network drivers to co-exist. In OS/2 2.0, not only can different applications be run at the same time as well as the network, but also any different combinations of settings can be easily maintained in DOS settings, and different configurations run at the same time.

- ***Swap file:*** To allow for better performance in swapping to and from disk in a memory-constrained environment, Windows 3.x offers a permanent swap file. This is a contiguous area of disk, usually not accessible via DOS commands, and therefore preventing other files making any use of that space. Indeed, even if Windows does not need the full amount of space, it cannot be regained for application use until the feature is disabled. The OS/2 SWAPPER.DAT file is designed to shrink as well as grow, so it is more flexible in allowing other files to be created alongside it.

These are a just a few examples of the way in which OS/2′s integration of DOS applications is superior to that of Windows 3.1, and such integration translates into lower running and support costs.

In conclusion, we can certainly say that as a DOS environment, which appears to be the most important consideration even to many Windows users, OS/2 is a better Windows than Windows. Now let′s take a look at the other consideration, compatibility with Windows applications.

Running Windows applications under OS/2

Standard mode support

As far as OS/2 2.0 is concerned, Windows applications are just a special case of DOS applications, which need a special environment (Windows 3.x) to run. The key, therefore, to Windows applications compatibility, is to provide those applications with as similar an environment as possible to what they have under DOS, while taking advantage of the inherent design superiority of OS/2.

First of all, it should be understood that the Windows 3.0 shrinkwrapped, retail package, can be run in a VDM in Windows real mode, (though Windows 3.1 cannot be run in this way because it does not support real mode) and Windows applications started from within this VDM by the Windows Program Manager. It cannot be run in standard or enhanced mode because of the way Windows has implemented the DPMI memory management scheme (it assumes it is a DPMI host and cannot act as a DPMI client - see “DOS extenders” on page 45). Many Windows applications run in real mode quite adequately. In fact, applications written for Windows 2 cannot run in any other mode, and therefore will not run in Windows 3.1. It may be true that many Windows applications have been upgraded for Windows 3, but this forces an upgrade on users who may be quite happy with the function in their current package. It means that the Windows upgrade path is now closed to users of Windows 2 based applications. OS/2 2.0 does not force such a choice, and will run Windows 2 and Windows 3 applications side by side, without having to exit and run in another mode. In fact, some have commented that it is ironic that OS/2 2.0 runs a wider range of Windows applications than Windows 3.1 itself.

Standard mode is needed for many Windows applications (eg Excel 3.0 and 4.0). To accommodate these applications, OS/2 needs to provide additional support. Basically, these applications need to access DPMI services for extended memory support, which is usually supplied by Windows 3.x running in standard mode or above. As we have seen already, OS/2 2.0 offers DPMI services and the ability to access extended memory.

The other requirement is to supply Windows services to Windows applications. This is done in OS/2 2.0 by modifying the Windows kernel and running it in standard mode in a VDM. As part of the joint development and cross-licensing agreement between IBM and Microsoft, IBM has access to both the Windows 3.0 and 3.1 source code. IBM has modified the source to provide a Windows kernel capable of running as a well-behaved DPMI client within an OS/2 VDM (the retail version of Windows 3.0 can only be a DPMI host). This means that OS/2 2.0 includes all the necessary Windows code to run Windows applications. A separate copy of Windows 3.x is not required. The Windows environment in OS/2 is called WIN-OS/2.

Since Windows 3.1 shipped after OS/2 2.0, OS/2 provides a modified Windows 3.0 kernel in the release made generally available in March 1992. IBM has the ability to, and intends to, include Windows 3.1 support in a future update, even though there are very few applications that require this support (see "Windows 3.1" on page 66).

OS/2 therefore supports Windows applications running in standard mode in a VDM. This means that the Windows kernel that runs under OS/2 runs in standard mode, and Windows applications run just as they would under DOS Windows 3.x in standard mode. The use of the VDM design, which provides a self-contained DOS environment, means that the environment is identical, from the application′s point of view, to running under Windows loaded in standard mode, on DOS. This design therefore provides the maximum compatibility with the DOS/Windows environment. In fact, it offers a wider range of compatibility, since Windows 2 applications, which require real mode operation under Windows 3.0 in DOS, can be run alongside Windows 3.0 applications running in standard mode. This combination is not possible at the same time in Windows 3.0, and is not possible at all in Windows 3.1, since Windows real mode has been discontinued in Windows 3.1.

386 Enhanced mode

Windows has another mode - 386 enhanced mode, which requires the 386 processor. OS/2 2.0 does not need to support enhanced mode for Windows applications, since for most Windows applications, there is no difference between standard and enhanced as far as the application itself is concerned. Standard mode provides Windows applications with all the memory management and other functions they need. Enhanced mode adds to standard mode the capability for multi-tasking DOS sessions and demand paging for efficient virtual memory, both of which are provided (in a superior fashion) by OS/2 2.0 itself. Indeed, Microsoft recommends in the Windows 3.0 manual on page 429, that users running only Windows 3.0 applications should run in standard mode, even on 386 systems with 2-3MB of memory, as there is a performance improvement in doing so. Therefore, OS/2 2.0 provides equivalent function to enhanced mode for most users.

There are, however, a small number of Windows applications which require enhanced mode to run. These are not supported in the first release of OS/2 2.0. It is believed that there are only at maximum three or four of these in the market, which represents less than 0.1% of Windows 3.x applications, if we take Microsoft′s figure of 5,000 Windows applications. Such applications require enhanced mode either because they rely on features only available in enhanced mode, or have been coded using the WINMEM32.DLL, a set of routines that provide some 32-bit functions for Windows applications, such as Wolfram Research′s Mathematica and Caere Omnipage Professional. It is unlikely there will ever be many in the latter category of applications, since the WINMEM32.DLL is very difficult to use, and Microsoft themselves warn in Appendix E of the Windows Programmers Reference: "only experienced Windows application programmers with extensive experience writing assembly-level code should attempt to use these functions in an application." That is because even something as basic as memory management using these routines can be very complex, requiring the programmer to create his own assembly language interfaces between the 16 and 32-bit parts of his program (note that such "thunks" are provided by OS/2 2.0 between 16 and 32-bit modules - see "Mixed 16-/32-bit environment" on page 12). Charles Petzold, possibly the most widely respected authority on Windows programming, whose book on the subject is a standard reference work, concluded on this subject that “something is seriously wrong when memory access becomes difficult,” and contrasted the current Windows approach with the ease of 32-bit memory management under OS/2 2.0. At the time of writing, Microsoft had not stated whether WINMEM32.DLL applications would be able to run under a 32-bit Windows environment and have said that it is unlikely that such Intel-specific applications would run unchanged in a future version of Windows/NT running on a RISC processor.

So, although there are a very small minority of Windows applications that will not run under OS/2 2.0, the vast majority will run, and in a mode which allows access to their full function. Indeed, to the Windows application, the environment will appear exactly the same as under DOS/Windows standard mode, but there will be greater overall protection, and the ability to pre-emptively multitask such applications along with DOS, OS/2 and other Windows applications.

Contrast with previous approaches (BCL)

It should be emphasised that IBM′s approach of building Windows code into OS/2 is quite different from previous approaches by Microsoft, which have included a Binary Compatibility Layer (BCL) which attempted to translate Windows calls to PM calls in real time. By providing the “real” Windows code, modified mostly for memory management purposes, the potential for performance degradation is greatly minimised. Those who previously suggested that IBM′s approach is “impossible” may have been thinking of a BCL-type approach, rather than the method actually used in OS/2 2.0.

How they run

Single session or separate session

Windows applications can be started from an icon in the Workplace Shell, just like other applications: OS/2 16 and 32-bit, DOS etc. They are launched into a VDM which is, like other VDMs, protected from the rest of the system. Users have a choice of loading the Windows application into its own VDM (even if other Windows applications are already running in separate VDMs) or being loaded into a VDM which contains other Windows applications. These are termed Single application and Multiple application VDMs (SAVDM/MAVDM) respectively. This is controlled in the settings for the program object: when choosing “WIN-OS/2 window,” you can check (SAVDM) or leave empty (MAVDM) the check box entitled “Separate Session”; or if running in a WIN-OS/2 Full Screen session, simply load the program from the Program Manager in the Full Screen session in the normal way.

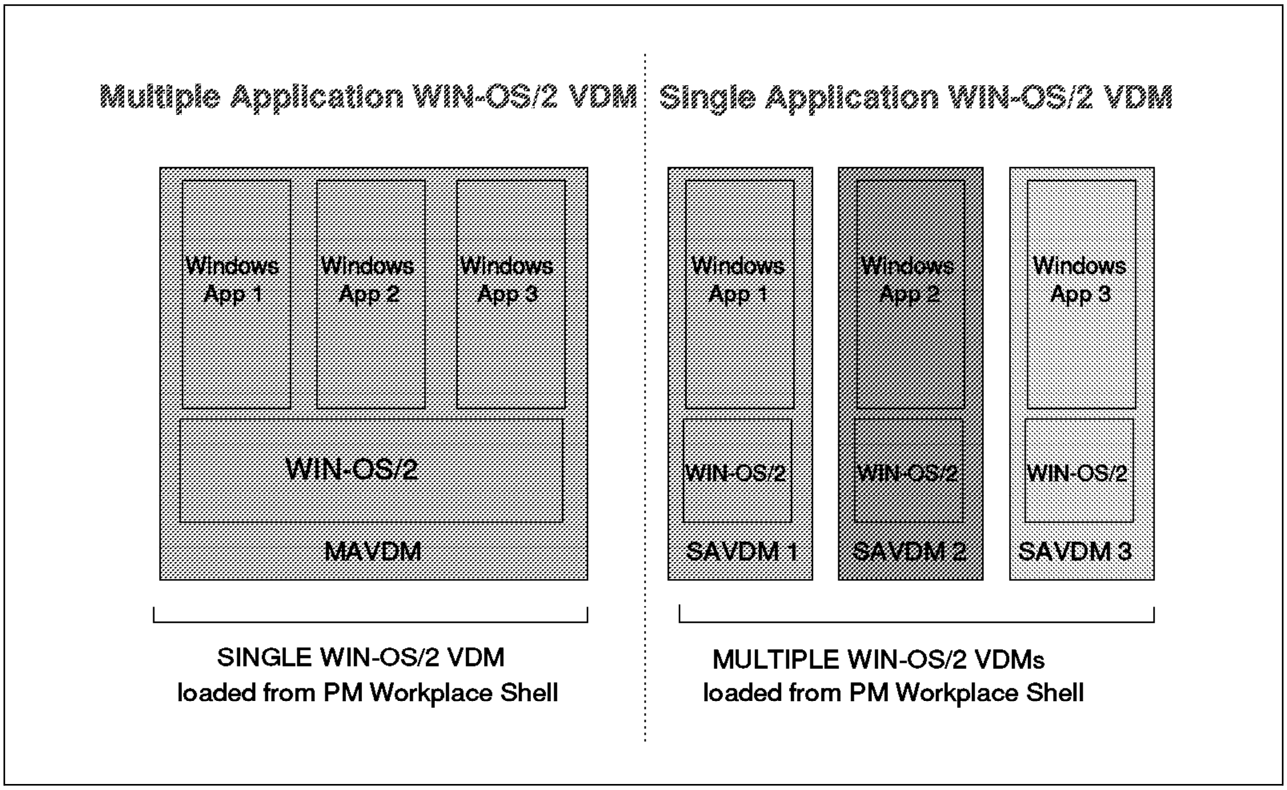


Figure 13. Single and Multiple application Windows groups

Single application Windows sessions load the application executable (eg EXCEL.EXE for Excel 3.0) along with a modified WIN.COM file (WINOS2.COM). This is exactly the same effect as typing “WIN EXCEL” when running DOS/Windows 3.x. The Windows application then runs under the WIN-OS/2 environment, in its own VDM. As far as the application is concerned, this is exactly the same as running under DOS/Windows; the modified Windows kernel provides the same services in a compatible way. Additional Windows applications can be loaded into separate VDMs in the same way.

Multiple application Windows sessions allow the user to run several Windows applications within the same VDM. This is the closest possible fit to the DOS/Windows usage, especially when run in a separate WIN-OS/2 Full Screen session, when it is difficult to tell you are not running under DOS. In this way, users migrating to OS/2 from Windows are able to run under OS/2 but have the general “look and feel” of their DOS-based Windows environment.

Single application VDMs provide the best protected way to run Windows applications under OS/2. Since the application runs in a self-contained Windows environment in its own VDM, it is fully protected from other applications, and the system is protected from it. This means that if the application crashes for any reason, it only affects that VDM, and in fact only that one application. Even other Windows applications running in other VDMs are not affected. This is a significant improvement in reliability over Windows under DOS (both 3.0 and 3.1), in which, since all Windows applications and Windows itself share the same Local Descriptor Table (LDT), a failure in one Windows application may bring down the entire Windows system or corrupt the data areas of other Windows programs.

But even though the SAVDM is fully protected, it can still share data via clipboard or DDE with other Windows applications or PM applications (see “Clipboard/DDE” on page 64).

However, each SAVDM must load its own copy of WIN-OS/2, and will therefore increase the overall working set of OS/2, affecting memory requirements and performance. If memory is limited, MAVDMs provide the ability to run several Windows applications with less resources required. But this is at the cost of losing the additional protection of the separate VDM. If one Windows application crashes within a Multiple application Windows session, it may cause all the applications within that VDM to fail; but the effect is only within that VDM - the other VDMs running DOS or Windows applications within other VDMs are not affected and continue executing. So even here there are benefits running Windows applications under OS/2, for greater reliability from system crashes. MAVDM will also lose the additional WIN-OS/2 benefit of pre-emptive multi-tasking for the Windows applications within that VDM (but not any other WIN-OS/2 sessions). MAVDM is also a requirement where Windows applications need to communicate with each other via shared memory, and where applications want to use OLE between each other (see “OLE” on page 65).

Full Screen or Seamless

In the Limited Availability version of OS/2 2.0 which shipped in December 1991, a WIN-OS/2 VDM ran only in a separate Full Screen session. The user clicked on an icon (eg WordPerfect for Windows) in the Workplace Shell, and a separate Full Screen WIN-OS/2 VDM was launched along with the Windows application. The user could toggle back and forth using Alt-Esc, Ctrl-Esc, or a little OS/2 icon that appeared in the corner of the Windows screen. But this was the only visual clue that you were in OS/2 at all: the screen looked exactly like DOS/Windows 3.0. This Full Screen function is still provided in the product shipped in March 1992 (there is an icon marked “WIN-OS/2 Full Screen” in the Command Prompts Folder).

There are some minor differences between the icons that appear in the WIN-OS/2 Full Screen environment versus DOS/Windows: many of the Windows mini-applications and utilities (eg Notepad, games, PIF editor) have OS/2 equivalents or are no longer relevant; others, such as the clock, are retained for compatibility. OS/2 adds icons for returning to PM from a WIN-OS/2 Full Screen session (since the Windows user wants to have access to either Ctrl-Esc or Alt-Esc key combinations for compatibility with his Windows environment) and for DDE management (see later).

The Windows print manager is retained, and Windows printer device drivers supported. The Windows Control Panel can be used to install Windows printer drivers not supported by OS/2, and to configure other Windows printer drivers, just as in DOS/Windows.

This full screen Windows function fulfilled the commitment made in April to provide Windows compatibility in 1991 (completed in the Limited Availability (LA) product shipped in December 1991). But during the early autumn of 1991, IBM received feedback from many of its beta testers to bring forward the functions it had been discussing for a future version to run Windows applications on the same screen as other DOS and OS/2 applications. The decision was taken to move the GA (General Availability) date to March 1992 to allow this function (and others) to be added. It was first shown in Lee Reiswig′s “OS/2 live” show at Fall Comdex in October 1991.

This latter function has since been called “Seamless Windows.” But the term is not very accurate; it was chosen by the marketplace, not by IBM. It is important, in fact, to note that most of what makes OS/2 2.0 “a better Windows” was already delivered with the LA product in December: protection between applications, pre-emptive rather than co-operative multi-tasking for better and more consistent multi-tasking performance, integration with other DOS and OS/2 PM applications via clipboard and DDE, with no loss of compatibility. Many people still run their Windows applications full screen, even with the GA version, as it does what they need. In fact, full screen has some benefits over Seamless operation (better performance, especially for clipboard and DDE, and more visually compatible with DOS/Windows).

In any case, much of the WIN-OS/2 environment represents a “seamless” migration from the DOS/Windows environment anyway: automatic installation of Windows printer drivers, re-creation of existing Windows program groups and WIN.INI settings.

Also, for most users, the most important aspects of integration are to be able to run all applications from the same user interface, and to allow all applications to share data. These benefits apply to Full

Screen and Seamless operation alike. Windows applications can be loaded from the same folders as DOS and OS/2 applications, from the new Workplace Shell, and data can be shared via clipboard and DDE.

But in the context in which it refers to the opposite of “Full Screen,” “Seamless” operation merely refers to the way the WIN-OS/2 application is displayed on the screen, in its own window on the Workplace Shell desktop as opposed to in a separate Full Screen session (see the diagram below, using Excel 3.0 for Windows):

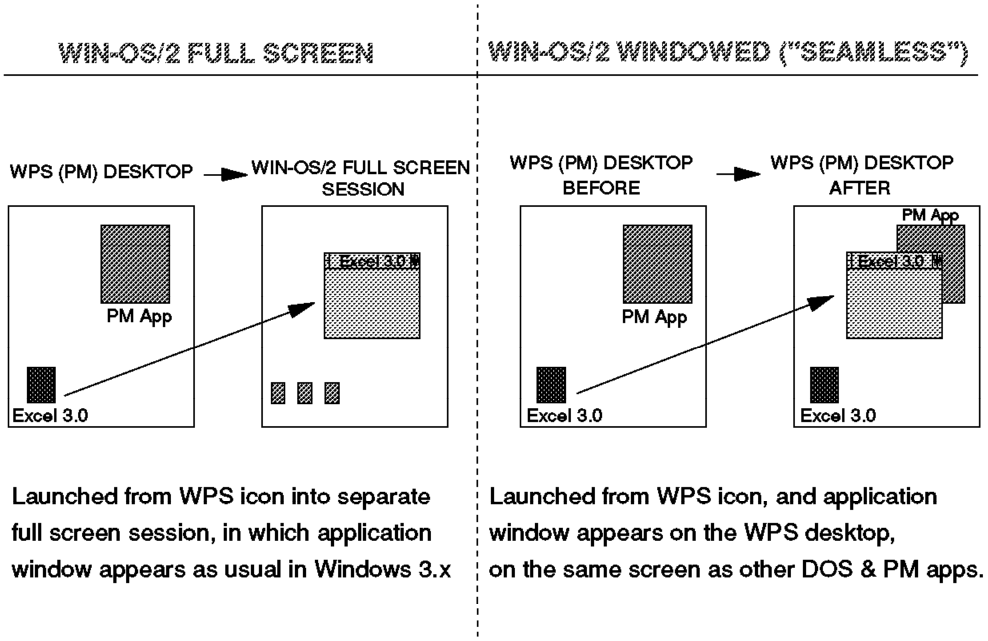

WIN-OS/2 FULL SCREEN
WIN-OS/2 WINDOWED ("SEAMLESS")
WPS (PM) DESKTOP
WIN-OS/2 FULL SCREEN SESSION
WPS (PM) DESKTOP BEFORE
WPS (PM) DESKTOP AFTER
PM App
Excel 3.0
Excel 3.0
PM App
Excel 3.0
Excel 3.0
PM App
Excel 3.0
Excel 3.0
Launched from WPS icon into separate full screen session, in which application window appears as usual in Windows 3.x
Launched from WPS icon, and application window appears on the WPS desktop, on the same screen as other DOS & PM apps.

Figure 14. Windows Full Screen and Seamless Sessions

Note that the fundamental way the WIN-OS/2 application works is the same whether it runs Full Screen or Seamless: it runs in its own protected VDM; it runs with the modified Windows code, it runs in standard mode. But the additional element is the way the screen output is mapped on to the Workplace Shell screen. Also, you can have any combination of SAVDM/MAVDM, or Full screen/Seamless:

- SAVDM Full Screen (WIN-OS/2 Full Screen in DOS Settings)
- SAVDM Seamless (WIN-OS/2 window in DOS Settings - “Separate Session” checked)
- MAVDM Full Screen (WIN-OS/2 Full Screen object in “Command Prompts”)
- MAVDM Seamless (WIN-OS/2 Window - “Separate Session” unchecked; the default for most applications)

How does it work?

Please note that what follows is a simplified explanation. There are no specific considerations for users or programmers in the way the Seamless WIN-OS/2 operation works.

The two key considerations in the design of the “Seamless” function were:

1. Maintain compatibility
2. Retain high performance

This resulted in a more “low level” implementation to avoid the limitations inherent in attempting to map Windows calls to PM. Instead, both the PM and Windows screen device communicate with each other via a Virtual Device Driver (VDD), VWIN, providing synchronisation of access to the video display hardware. Each device driver “owns” its section of the screen. (This was a significant task in the time available, even for VGA; it can easily be understood why Seamless function is only available for VGA in the GA March 92 release, but drivers for above VGA are in development and will be delivered later in 1992). A new entry in WIN.INI, *SDISPLAY.DRV=* , denotes the change. The best way to understand the way it all works is to consider an example. Refer to the numbered diagram below while reading the following explanation:

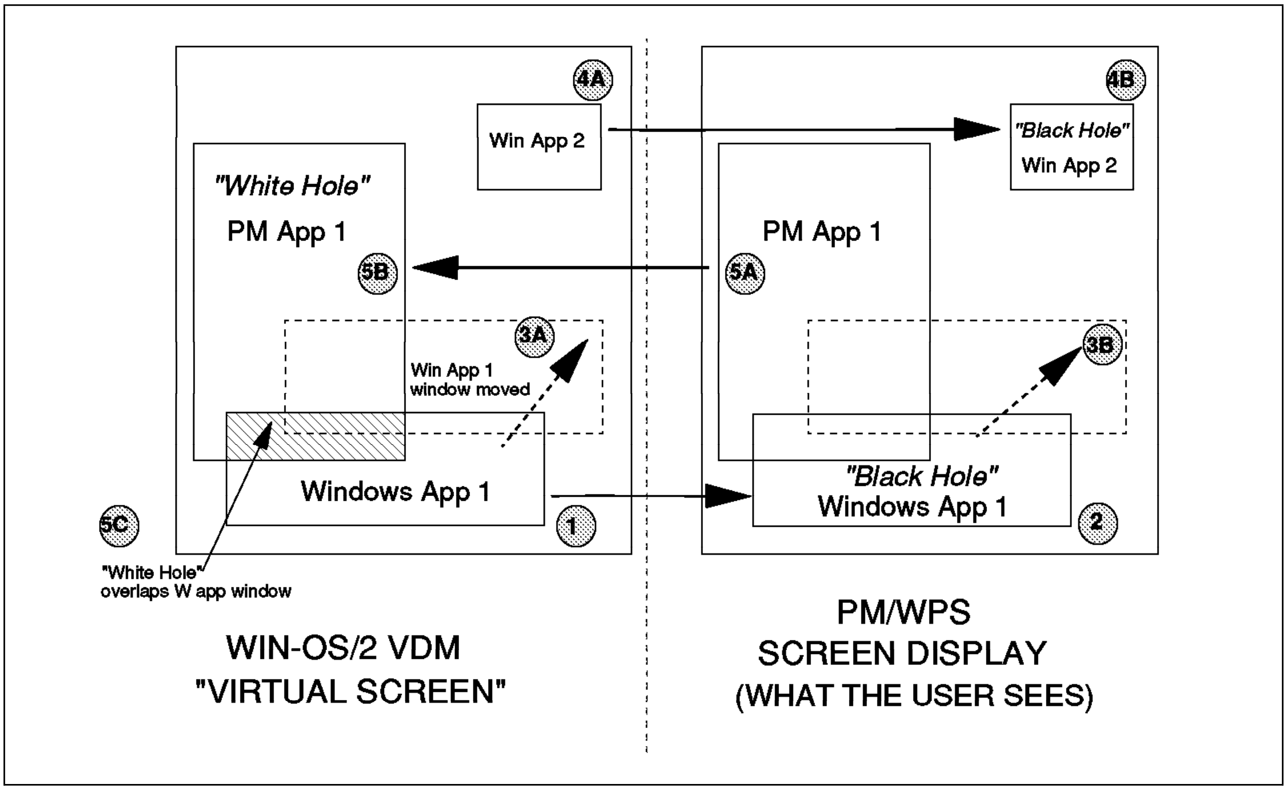


Figure 15. How Seamless Windows works

When a WIN-OS/2 application creates a window, the Windows screen device driver requests the PM screen device driver to open up a “black hole” of the same dimensions (see step 1 in the diagram). The Windows device driver then has control of the video hardware and paints into this space (step 2).The result is that the WIN-OS/2 application (which is completely unaware of this process) creates its main and secondary windows on the Workplace Shell screen alongside (and sometimes overlapping) other OS/2 application windows. Notice this means that the WIN-OS/2 application paints its own frame controls and mouse pointer (eg title bar, minimise/maximise buttons) and even the infamous Windows hourglass!), and these controls will therefore look like Windows controls, not PM ones. If you put Excel for Windows and Excel for OS/2 side by side, you′ll see the difference. Although this creates some minor visual inconsistencies (most users don′t even notice!), it preserves compatibility and performance.

The WIN-OS/2 application window can be sized and moved as normal merely changing the co-ordinates of the “black hole” into which it paints (see steps 3A and 3B). The PM screen device driver repaints any part of the PM screen space previously governed by the WIN-OS/2 window. Other WIN-OS/2 applications (whether launched from the same WIN-OS/2 VDM or not) can create their own main windows (see step 4), or the existing applications can create secondary windows which can extend beyond the original window′s co-ordinates. Corel DRAW! for Windows creates a main window with a tool bar which hangs below the bottom of the window if resized, resulting in a non-rectangular window. OS/2 2.0 handles such exceptions without a problem.

If an OS/2 application window is created, moved or resized to overlap a WIN-OS/2 application′s window (see step 5A) the PM screen device driver will cause a “white hole” to be created in the “virtual” screen space of the WIN-OS/2 session (step 5B). This causes the WIN-OS/2 application to avoid painting the part of the window that has been “covered” by the OS/2 application, and restrict repainting to the portion which is still “visible” under or around the “white hole” (see step 5C). If the OS/2 application window is subsequently closed, moved or resized, the screen drivers communicate again to change the size of the “white hole” (in a similar way that the co-ordinates of the “black hole” are changed in step 3B).

The use of “black” and “white” holes means that WIN-OS/2 and OS/2 (including DOS) application windows can overlap each other normally, without one incorrectly painting over the other. The two screen drivers keep a record of each other′s state without affecting any of the OS/2 or WIN-OS/2 applications. This means there are no significant compatibility or programming considerations. Since the PM screen device driver is always aware of the changes made by WIN-OS/2 applications′ windows, there is no impact to OS/2 applications, and likewise for WIN-OS/2 applications which create, move, resize and destroy windows in the normal way. Performance is good, because communication is at a low level between screen drivers, not an API mapping layer, and compatibility is excellent because no fundamental changes occur to either the Windows GDI or PMWIN (the main window-creating components of WIN-OS/2 and PM respectively). And as far as the user is concerned - it just works, minimising the difference between DOS, Windows and OS/2 applications. In fact, it achieves the aim of “seamlessness” by making applications behave in similar ways, whether DOS, Windows or OS/2 applications. To the user, it appears easy, but the technical accomplishment behind it is considerable.

High resolution support

At General Availability time in March 1992, Seamless drivers were only available in VGA. This means that the whole system (including PM) must run in VGA if Seamless operation is required. The installation program gives users with 8514/A or XGA adapters the choice between high resolution WIN-OS/2 in full screen, or Seamless in VGA. Remember that the choice also affects the resolution PM appears in. This choice will become redundant once Seamless drivers for resolutions higher than VGA appear. Drivers for XGA and SVGA will become available by the end of 1992 (see “OS/2 1992 developments” on page 109).

Since OS/2 2.0 shipped in March, IBM and other vendors have continued work on drivers at resolutions higher than VGA. Drivers for XGA and some SVGA chip sets will be made available in a service pack by the end of 1992, rather than than waiting for a new version of OS/2. It is to be remembered that to enable Seamless operation, both Windows and PM screen drivers need to be modified, for each resolution.

Clipboard/DDE

An important benefit of the Windows environment is the inter-application communications it offers to Windows applications, allowing them to share data. The two most common methods for achieving this are via the clipboard, and Dynamic Data Exchange (DDE).

The Windows clipboard, like PM′s clipboard, usually allows applications to share data on a “once only” basis. Data is “cut” from one application to the clipboard, and “pasted” from the clipboard into another application. Applications that support the clipboard sometimes allow a link to be set up between applications, using the same or similar commands, and that link is usually handled via DDE. DDE is a message-based protocol allowing applications to pass information back and forth, updating items as they choose. For example, an information feed like a stock market ticker, can be linked via DDE to a spreadsheet, and update it with the latest information as the data changes.

Both clipboard and DDE are supported for Windows applications in OS/2 2.0. Since OS/2 already supports both of these functions (OS/2 1.3 provided similar functions for OS/2 16-bit applications), it is mainly a matter of integrating the Windows support within OS/2 so that Windows and OS/2 applications can access the same function.

The way in which this is done in OS/2 2.0 is quite similar for clipboard and DDE. Each requires one protected mode (OS/2) program acting as a "server" for messages between applications. Also, in each VDM running a Windows application, a pair of applications needs to be running: a modified version of the Windows clipboard viewer program for clipboard, and a "ServerAgent" program for DDE. The ServerAgent, which is started when the VDM is started, takes DDE messages from the Windows application and routes them to the DDEServer and receives messages from the DDEServer and passes them to the application running in the VDM.

Clipboard and DDE are also available between applications running in the same VDM. This works in exactly the same way as today under DOS/Windows; the local VDM-based clipboard or ServerAgent programs are not needed unless clipboard and DDE are required outside the VDM. Users can also specify that their clipboard and DDE is local to that VDM only, or "public" to the whole system. If sharing between Windows and OS/2 applications is not required, the clipboard should be kept "private," to improve overall performance.

The result of this is that clipboard and DDE are supported in OS/2 2.0 between Windows applications, and between Windows and OS/2 applications. DOS applications can also participate in clipboard sharing. Thus, there is a consistent way of sharing text and graphical data between DOS, Windows and OS/2 applications.

OLE

Object Linking and Embedding (OLE) is a mechanism created by Microsoft and endorsed by other vendors, to extend the data sharing currently possible in Windows. It is intended to support the creation and editing of "compound documents," where elements may come from a number of different applications, and still be editable by the source application, even when linked in another document or application.

Several ISVs are intending to support OLE, but today only a minority of applications actually support it. Examples include Microsoft Excel version 3.0, and Lotus Notes version 2.0. Windows 3.1 is expected to simplify the enabling of this function, because it includes the necessary support in the product, but applications still need to be rewritten to take advantage of it. Therefore, OLE is more of a potential standard in the Windows world, than a current, widely-used one. In addition, it has some technical limitations which preclude its widespread use in a networked environment in its current form.

Microsoft had previously announced their intention to provide OLE libraries for OS/2 in mid-1991, but no toolkit has yet been shipped. OS/2 2.0 supports OLE between Windows applications in the same Multiple application Windows session, as long as the applications include their own OLE DLLs. An example of such an application is Microsoft Word for Windows version 2.0, which includes its own "mini-applications" for functions like charting and drawing, and links to the main word processing module using OLE. Since there is not yet any equivalent function in OS/2, OLE will currently work between Windows applications only. IBM is currently investigating a superset of current Windows OLE function for a future release of OS/2.

Windows 3.1

In April 1992, Microsoft shipped an updated version of Windows, Windows 3.1. It offers a number of new functions, and a substantially expanded API set, including Multimedia and Pen extensions (although the latter functions require separate add on products). Since OS/2 2.0 had already shipped by the time Windows 3.1 shipped, IBM has no specific support for Windows 3.1 APIs in the first release of OS/2 2.0, nor is it certain how important such support is. IBM′s aim is to run Windows applications, not Windows. Windows 3.1 support is only an issue if it enables new windows applications to be developed which may not run under the current level of WIN-OS/2. Whether developers will build in functions requiring Windows 3.1, will depend on how they view the risk of only targeting part of the Windows installed base (those who have moved up to 3.1). Microsoft promised compatibility between Windows 3.0 and 3.1, and are offering developers the opportunity to ship the extra 3.1 function in DLLs along with the application. If this enables the application to run under Windows 3.0, it should do for WIN-OS/2, by the same logic. If developers follow this path, the need for Windows 3.1 support in OS/2 2.0 may not be as urgent as some suggest. Indeed, there have been very few Windows 3.1-specific applications appearing in the first few months after the shipment of DOS/Windows 3.1. At one time, Microsoft had announced that they would make Windows 3.1 capable of being a DPMI client, and thus run under OS/2 2.0. However, Windows 3.1 does not possess this capability.

Nevertheless, IBM has the Windows 3.1 source code through the joint licensing agreement and will continue to monitor the need for additional levels of Windows support. IBM is fully able to provide such support without help from anyone else, as it has done so far with the development of OS/2 2.0. On April 6th 1992 Microsoft shipped Windows 3.1. On April 7th, IBM showed the Windows 3.1 Program Manager running in a Seamless window under OS/2 2.0 (not the GA version). This was not an announcement of Windows 3.1 support, merely an indication that it could be done. Work is now taking place on updating WIN-OS/2 support (see “OS/2 1992 developments” on page 109) and is in beta test at the time of writing. The main part of the work is extensive compatibility testing with Windows 3.0 applications. The aim is to ensure that the change to Windows 3.1 level of support, does not lead to the kind of compatibility problems that were reported in *PC Week* soon after the release of DOS/Windows 3.1, and referenced in the Windows 3.1 APPS.HLP file. Because of these compatibility problems, and the withdrawal of support for Windows real mode by Microsoft, it is true to say, at the time of writing, that OS/2 2.0 runs a wider range of Windows applications than DOS/Windows 3.1.

A better Windows?

So, in conclusion, the approach taken to running Windows applications under OS/2 2.0 has a number of benefits:

- ***Compatibility***: gives high compatibility with the widest range of Windows applications, since the applications are running under the real Windows code, modified only for compatibility with OS/2.
- ***Performance***: since the applications run under a modified version of Windows itself, it does not suffer from the performance limitations of a Binary Compatibility Layer (BCL) approach, such as was previously attempted (and subsequently abandoned) by Microsoft. Although single tasking scenarios may be up to 20% slower, multi-tasking performance is comparable to that under Windows and in many scenarios better, because of the superior multi-tasking design of OS/2. Indeed, the more the system is loaded, the faster Windows performance will tend to deteriorate in comparison with the same configuration under WIN-OS/2. Even the example quoted earlier from NSTL′s independent benchmarking (see “Performance” on page 55) shows Windows performance less than WIN-OS/2 even with only one background process.
- ***Protection***: since Windows applications can run in separate VDMs, they are better protected from each other, so that errant applications cannot bring down the system. Windows applications themselves are protected from DOS applications more than is possible under DOS/Windows.

And, as we have already seen, OS/2 2.0 represents a superior environment for multiple DOS applications (see “OS/2 versus Windows as a multi-DOS environment” on page 54), since there is more memory, better multi-tasking, and more protection than under any DOS or DOS extender. Therefore, in being a better multi-DOS environment, and running Windows applications with full compatibility and extra protection, OS/2 2.0 is a better Windows environment overall.

Porting Windows applications to OS/2

Of course, although OS/2 can offer excellent compatibility with Windows applications, those applications still remain Windows applications, and they are therefore limited in the extent to which they can integrate with OS/2. For example, full drag and drop functionality and other exploitation of the Workplace Shell, is only possible if you write for that environment, and that means OS/2. OS/2 applications have full access to a 32-bit API, 32-bit memory management, and multi-threading capability, allowing the same application to be much more responsive to the user (for example, retrieving a new file while saving the old one, rather than having to wait with the hourglass showing, as you might in a 16-bit, single-threaded environment like DOS/Windows.) Minor issues of look and feel are also different (as is apparent when the application is run in a window alongside OS/2 applications).

Compatibility with existing Windows applications is important: but it does not mean that users should not be offered the chance of using better, more powerful and more responsive applications by using native OS/2 applications. Some have tried to suggest that having a Windows application is enough, since it will run under OS/2 anyway, and therefore one version will cover both markets. Developers who have followed this “lowest common denominator” approach, are now finding themselves unable to differentiate themselves from the competition, and indeed at a disadvantage in the OS/2 marketplace.

Increasingly, users who migrate to OS/2 2.0, are demanding “native” OS/2 applications, and vendors who can only offer Windows applications running under WIN-OS/2 are finding themselves uncompetitive if they do not supply a real 32-bit OS/2 version. WIN-OS/2 may be a “better Windows,” but it is not as good as real 32-bit OS/2 and the Workplace Shell. Therefore, vendors have considerable incentive to write for OS/2 and not keep only to Windows support. And over 1000 OS/2 32-bit applications are in development: more than 150 shipped within three months of the release of OS/2 version 2.0. Vendors who have committed to OS/2, or who already have products available, include Lotus, Borland, Software Publishing Corp., Novell, Corel Systems, Micrografx, WordPerfect, DeScribe, ZSoft, Oracle, Gupta and Computer Associates. And IBM is also delivering a number of applications (in addition to the systems extensions described in “OS/2 in a connected environment” on page 97) including Personal AS/2, and ImagePlus/2.

Writing for OS/2, and in particular using the OS/2 32 bit API, positions the software developer to take advantage of the developments in the OS/2 environment, including multimedia, distributed computing and the increasing use of object technology (see “Futures” on page 109).

However, possibly the most convincing argument for developers now, is the fact that over one million copies of OS/2 2.0 have been shipped within six months of its first availability, and OS/2 2.0 looks set to continue its success. The one million mark is for many commercial developers, a sign of viability for a platform. Ed Zander, president of SunSoft, Inc., the software subsidiary of Sun Microsystems, was quoted in *PC Week* about Microsoft′s views on porting its applications to SunOS: “Microsoft tells every company that walks in there that it won′t do the ports until the architecture has an installed base of 1 million”[18] Microsoft′s Bill Gates was quoted by *PC Magazine* as saying he would only consider

[18] Quoted in *PC Week*, July 29, 1991 - p111

developing for OS/2 when it shipped more than 2 million units.[19] OS/2 has already passed the one million mark and is heading towards the second. Whether the report in *PC Magazine* has any relevance to any future plans Microsoft may have for OS/2 2.0 applications, is unknown.

Although the Windows and PM APIs are in some respects different, porting tools like the Developers′ Migration Kit (see below) are being developed which aid the migration of Windows applications to OS/2, to allow them to take advantage of OS/2 benefits. These are tools at the developer rather than user level: the Windows applications compatibility solution described above is a user-oriented solution, to gain maximum compatibility with existing shrink-wrap applications and derive some inherent OS/2 benefits like true multi-tasking and reliability; there are also developer tools, which allow applications to be ported from Windows to OS/2 and add OS/2 functionality in the process. Among such tools are those being produced by IBM and Micrografx.

IBM/Micrografx porting tools

In April 1991, IBM announced a joint development and licensing agreement with Micrografx, Inc., a leading vendor of Windows and OS/2 applications. Part of that agreement included IBM′s and Micrografx′s intention to produce a porting toolkit to help move Windows applications to OS/2.

The Developers′ Migration Kit/2, announced in July 1992, allows developers to port applications and drivers from Windows to OS/2 with little or no code changes. There are tools for applications and for device drivers. The applications porting kit, and the device driver kit are based on the Mirrors technology from Micrografx. Developers can maintain a common code base between Windows and OS/2, or OS/2 specific functionality as they port. The latter is recommended to enable better integration with the Workplace Shell (see “Workplace Shell exploitation” on page 73) Drivers ported with the driver tools, such as the HP Paintjet driver, are already part of the OS/2 2.0 base printer driver suite.

This will increase the number of OS/2 applications by leveraging the Windows application base, and make it easier for developers to participate in both opportunities. This in turn will give the user another option to take advantage of his existing investment, but also have a wider choice of real OS/2 applications.

The way in which the porting process takes place is similar to a product previously distributed by Microsoft, the Windows Libraries for OS/2 (WLO). WLO was derived from Micrografx Mirrors technology, but IBM and Micrografx are now using a more mature version of the technology in the Developers′ Migration Kit/2. Furthermore, the philosophy of the Developers′ Migration Kit is somewhat different to that of WLO: the latter retained something of the original BCL approach (see “Contrast with previous approaches (BCL)” on page 59), allowing a Windows application to run on OS/2 by emulating Windows on top of OS/2, providing OS/2 versions of the Windows libraries the application usually links to. Although the porting process is similar, the Developers′ Migration Kit is designed specifically to produce applications for OS/2 2.0 (not both 16- and 32-bit as was WLO′s aim), and has a number of additional tools to help optimise the port for the OS/2 environment. The result is likely to be better performing OS/2 2.0 applications than were possible using the WLO approach.

[19] The report in *PC Magazine*, May 12, 1992 - p32, said : “When CEO Bill Gates was asked what it would take for Microsoft to write for OS/2, he said 2 million copies in the first year - but they′ll sell less than 10% of that”

Better OS/2

OS/2 1.3 has been widely respected and well received. Nevertheless, as well as being a better DOS than DOS, and a better Windows than Windows, OS/2 2.0 is even a better OS/2 than OS/2 1.3. This section will show why.

Compatibility

OS/2 allows applications written for previous releases of OS/2 to work without modification on version 2.0. The design of OS/2 always took this into account, and provides binary compatibility not only for the commercial applications written by ISVs, such as Lotus 1-2-3 for OS/2, but also the “line of business” applications written in-house by various companies to support their key business activities. It is estimated by one independent source that as much as $4 billion of investment has been made in such systems running on OS/2. Compatibility with these applications is a statement of IBM′s intention to protect that investment while migrating forwards to new and better environments.

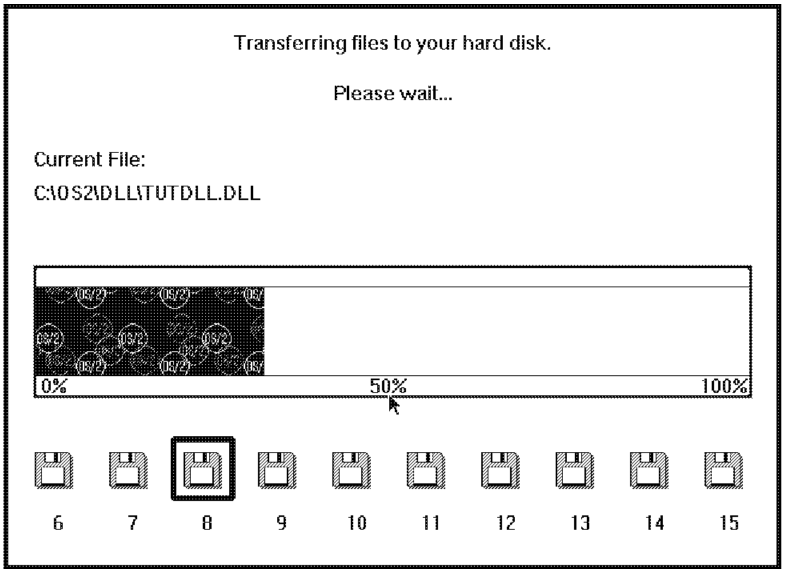
The OS/2 design had to take account of 16-bit applications running in a 32-bit system, and the system itself provides many services to allow 32-bit modules to call existing 16-bit modules, thus allowing for mixed 16/32-bit applications. This means that the mixed model gives great flexibility, both in migrating applications from 16- to 32-bit, and also in allowing 32-bit applications to make the best possible use of existing service routines, window classes etc, developed for previous releases of OS/2. (See “Mixed 16-/32-bit environment” on page 12).

Furthermore, the consistency of the API between 16- and 32-bit, allows relatively easy migration to 32-bit. The PM API in particular was designed with 32-bit in mind, and this makes the 16- to 32-bit conversion relatively straightforward. Conversion can be done in stages if preferred, the mixed 16:32 model allowing a variety of approaches from recompiling and relinking, to full rewrite for critical sections, optimising for 32-bit.

Graphical installation

The user′s first view of the system, at installation time, is very different from 1.3. Instead of being entirely text-based, after the first disks, installation is done in a graphical mode. This is more consistent with the main user interface, and provides better feedback to the user: a progress indicator bar shows how far through the installation the user has got, and pictures of disks appear at the bottom to show how many disks have been used and how many remain.

The Install program is more intelligent; it can figure out your mouse, display, keyboard and country setup, and offer defaults for you to choose. For those who need more detailed control over their system, the selective install which first appeared in 1.3 is also in 2.0, but as a graphical dialog box with check boxes to select the options and buttons to indicate how much each item takes up on disk, as well as a cumulative indication of the total disk space required for the components chosen. Among the options that can be chosen or left are the on-line documentation, REXX, fonts, HPFS, and MVDM/DOS support. These options allow further selection at a more granular level. For example, many people install DOS support but not Windows support, since they have no Windows applications. Even individual utilities and applications can be selected or not; and the size of each component is listed to help you determine whether or not you want to spare the disk space. This can make a difference, as the Tools and Games “applets” (see “Applets” on page 86) take up to 5.7Mb of disk space. But a listing of the individual utilities and games allows you to see what each item takes (eg games: JIGSAW 68K, SOLITAIRE 375K; applets: TERMINAL 1501K, PM CHART 1159K).



Transferring files to your hard disk.

Please wait...

Current File:
C:\OS2\DLL\TUTDLL.DLL

Figure 16. Installation program - progress indicator

The user is given an easier and less error-prone way to control some of the parameters in CONFIG.SYS, by cycling through valid options. Once the product is installed, further customisation is possible via the OS/2 system folder (which contains options similar to the OS/2 1.x Control Panel). Options can be added individually at a later point with the “Selective Install” object in the “System Setup” folder.

For users wanting to install OS/2 over an existing DOS system, OS/2 install offers migration of existing DOS, Windows and OS/2 applications. If the install program detects applications it recognises on the disk, it will offer the user the option to put them in a folder. This way, the system can come up with the user′s existing applications already available. These applications are also set up with an optimal collection of DOS and Windows settings. System Administrators can create their own custom database of applications to migrate in a similar way (see “Migrating applications” on page 43).

When installing over a Windows 3.0 or OS/2 1.3 machine, the program groups are migrated into folders so that the groupings of programs are the same in the new environment. Device drivers from either DOS or OS/2 CONFIG.SYS files can be migrated too, as well as printer definitions from either Windows 3.0 or OS/2 1.3.

For new users, the install can give a tutorial on use of the mouse in order to guide the user through installation.

As well as being simpler and more appealing to look at, installation is faster, due to an enhanced compression/uncompression algorithm for unpacking files from the disks. Install can be done from any drive to any drive, allowing OS/2 2.0 to be installed, in future, from CD-ROM, or any other media that can be accessed as a normal drive letter, including across a LAN if required. In addition, tools and services are available from IBM to assist the automated installation of many machines. Remote installation and systems management is a major area of development for OS/2, but many basic tools and facilities are here today (see “Configuration, Installation, Distribution (CID)” on page 104). Most important of all, is that OS/2 2.0 is already enabled for automated LAN-based installation. See the publication *OS/2 2.0 Technical Compendium, Volume 5: Remote Installation and Maintenance* (GG24-3780-00), for more details.

Such automated installation tools can substantially reduce the amount of time taken to install OS/2 on a number of machines, since installation can take place simultaneously on several machines without operator intervention. The option for automated, LAN-based install, reduces objections to the amount of time taken to install the operating system. Even in a standalone environment, installation is straightforward, and though the number of disks required for an OS/2 installation does mean that a manual install will usually take at least 20 minutes, it is usually one single operation to install the system, with one reboot in the middle. Many people forget that although Windows comes on less diskettes, it does require DOS to be installed first, causing a two-stage installation process, and, for certain requirements, some co-ordination between the DOS and Windows installations is required (for example, if you want to use a mouse in a DOS window inside Windows 3.x - see “Integration” on page 56).

Boot Manager

For users who wish to retain more than one operating system on their disk, OS/2 includes a multiple boot facility (Boot Manager). This allows different operating systems to be placed on separate partitions, in order to select which partition to boot at power-on or reset time. Since OS/2 can support the vast majority of DOS, Windows and OS/2 16-bit applications, it is not anticipated that this will be needed by many people for booting DOS or OS/2 1.3 instead of 2.0, but instead may be used with other systems such as AIX, or as a testing tool for developers or technical support staff.

It works by allocating a 1MB primary disk partition to a Master Boot Block (MBB) which contains code that is always executed first at boot time and handles further access to the hard disk. It displays a menu allowing the user to select the logical drive from which to start the system. Once this logical drive is selected, the operating system loader for the appropriate system is loaded as normal. New versions of FDISK and FDISKPM allow the user to select the logical drive for use. DOS and OS/2 1.3 have to be installed on a primary partition on the first hard disk, but OS/2 2.0 can be booted from any partition on any drive. This gives greater flexibility in planning installation than is possible using DOS, or Windows, which is dependent on DOS for booting.

OS/2.0 - the 32-bit system

There are a number of obvious benefits to users in working with OS/2 2.0. There are also many benefits for developers. Developer benefits can also be relevant to users: anything that helps developers ultimately helps users, since it becomes easier to write better and more powerful applications, and provide the opportunity for applications that could not be produced before.

Why 32-bit OS/2?

First, it is worth examining why it is significant that OS/2 is a 32-bit system, and what benefits this can provide.

Better performance

One of the most obvious benefits of a 32-bit OS/2 program is that it will almost certainly perform faster than a 16-bit equivalent. An OS/2 32-bit application can make use of the full 32-bit instruction set and extended registers of the Intel processor. Some 32-bit compiled programs will also be smaller in size because of the smaller number of CPU instructions needed to complete certain tasks. Low level functions such as string manipulation, data movement and pointer operations will often be twice as fast under 32-bit.

Furthermore, if the application is coded to the flat memory model, there are none of the overheads of segmented programming, such as the need to reload segment registers every time a different 64KB of memory needs to be accessed. This is slow and also an inconvenience and wasted code for the developer.

The system itself provides more efficient paging than both the swapping mechanism of OS/2 1.3, and the memory paging used in the 16-bit Windows system (even on a 386). Applications that take advantage of the smaller memory granularity (memory can be swapped out to disk in 4KB pages) by intelligent organisation of code and collaboration with the paging mechanism by allocating memory in page size units, can show a significant performance improvement.

The performance improvements that are possible depend on the changes made: they range between 5% and up to around 40%. An example of the improvements possible is in the REXX interpreter, part of the OS/2 base system. Performance has been dramatically improved since it was changed to run in 32-bit mode, as the graph below shows:

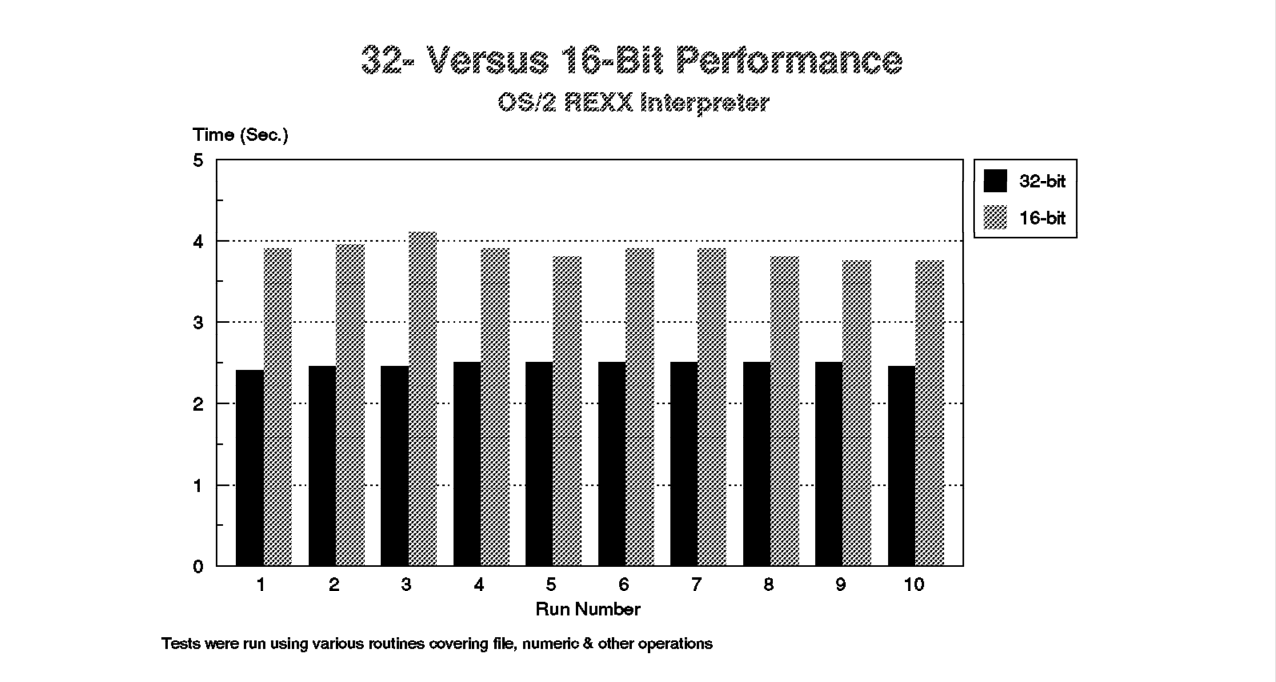


Figure 17. Comparison of 32- and 16-bit performance - REXX

Even more encouraging is how such benefits can be achieved with relatively little additional development time. since OS/2 was designed with 32-bit in mind, migration from 16-bit has been made as straightforward as possible (see "Migration 16- to 32-bit" on page 74).

The difference between 16-bit and 32-bit can be significant, especially when it is comparing Windows and OS/2 capabilities at the same time. DeScribe is a graphical word processing application which exists for 16-bit OS/2 (OS/2 1.x) and for 16-bit Windows (Windows 3.x). The company has, in the past, demonstrated beta test versions of its 32-bit OS/2 release (which shipped soon after OS/2 2.0′s March availability), comparing it with its own 16-bit Windows version. Comparisons made by Describe, Inc., show that the 32-bit OS/2 version can be anything between 30 and 300% faster than the 16-bit Windows version, to complete the same operation. In addition, the OS/2 32-bit version implements features like multi-threading which cannot be offered in the Windows version (since Windows 3.x cannot provide multi-threading). This makes it possible, for example, to save one file and then start loading another while the previous file is being saved to disk, or looking at one document while another is repaginating. This results in even higher ***perceived*** performance to the user, because the application is more responsive, and less time is spent looking at an hourglass icon.

More sophisticated applications

Applications that may need space and better ability to handle large data objects (eg DTP, CAD/CAM, financial modelling, multimedia, database) will be easier to develop in a 32-bit system. Indeed, some may not be possible or feasible in a 16-bit system. In a segmented environment, programmers have to develop their own algorithms to use multiple segments to implement a single logical structure, which is complex and may make it difficult to achieve performance objectives. This is another issue that may help some applications move from the Unix environment to OS/2, since they have not been feasible on an Intel platform till now. Also, OS/2′s support for overlapped I/O enables support for more sophisticated applications, like multimedia, where it allows better synchronisation between audio and video.

Workplace Shell exploitation

Windows and DOS applications will load from the Workplace Shell, but not take full advantage of the drag and drop capabilities and other methods of participating in the shell. It is possible to provide compatibility, but not full integration, since Windows applications remain Windows applications (see “Porting Windows applications to OS/2” on page 67).

Some of the examples of what can be achieved by integrating the application with the Workplace Shell include:

- drag and drop printing, and file loading
- drag and drop installation
- consistency of user interface with the rest of shell - ease of learning applications
- registration of application objects, so they can appear as templates and have their own file associations
- interaction with other Workplace Shell objects
- more powerful user navigation: context menus, settings controls, views etc
- on line documentation, including integration with the Master Help Index if required
- shutdown notification (eg start up with same collection of views and objects next time - pick up from where you left off)

Although the benefits of Workplace Shell integration do not specifically arise from the fact that the system is 32-bit, they can only be achieved today by coding to the OS/2 32-bit API. 16-bit Windows can provide very little of this function today.

Portability

Applications written for OS/2 2.0 will be much easier to port to other 32-bit systems like AIX than applications written for 16-bit systems, as dependencies on processor specifics (such as the Intel segmented model) are eliminated. Ports from other environments (Unix etc) will be easier for the same reason. This will give the user an even wider choice of applications.

Applications written for OS/2 2.0 will also be better prepared for systems as yet unreleased such as a future portable version of OS/2.

Simpler programming model

In OS/2 2.0, there is no need to take into account the requirements of the Intel segmented model, calling for code to be broken down into 64KB segments, and also no need for segment manipulation when passing control between threads or calling DLLs, for example, as is currently the case in 16-bit systems like OS/2 1.3 and Windows. Instead, memory is allocated with a 32-bit pointer that maps into an address space of 512MB, with no 64KB limit on the size of individual segments. Memory can therefore be allocated in logical units dictated by the requirements of the application rather than the constraints of the segmented memory model. This leads not only to better performance (reduced segment reloading), but reduced application development time (less code needed to do the same job, since segment handling code is no longer necessary).

32-bit OS/2 coding is also simpler because there is only one memory model, instead of the many (small, medium, huge) under the segmented application model (represented by OS/2 1.3, DOS, Windows).

The result for the user is that powerful and sophisticated applications will appear quicker, and some will be feasible for the first time, because they are easier to develop. Also more programming time can be spent on providing better function or usability rather than managing memory segments, which should also benefit the user.

Exploit hardware investment

32-bit programs allow companies to take advantage of the investment in 32-bit hardware systems that they have been purchasing for several years. Currently, 32-bit hardware is, in most cases, running 16-bit software. 16-bit software does not take advantage of the features of the hardware that have been optimised for 32-bit operation. 386 machines have simply been used as “go faster” 286 or 8086 machines, until a 32-bit system has become available to take advantage of it. As we have seen, the benefits of 32-bit are not just in performance, but in broadening the scope of what can be done. Waiting any longer for 32-bit alternative systems to become available, simply increases the length of time that the investment is left unexploited.

All in all, 32-bit equals a richer set of applications, which are faster, and allow more choice and more function for users; in this way, developer benefits become user benefits.

Migration 16- to 32-bit

Best of all, migration from 16- to 32-bit OS/2 applications can often be accomplished relatively easily. PM was designed with 32-bit use in mind, and the design of OS/2 2.0 has taken into account the need, not only to run OS/2 1.3 applications unmodified, but also to make it straightforward to change the 16-bit OS/2 application to exploit 32-bit. In some cases, little more than a few function call changes are required. In many cases, changes will be needed to change from a segmented 16-bit memory management to the flat memory model of 2.0. How much work is involved here depends on how many dependencies the existing 16-bit application has on segment manipulation. What certainly can be said is that movement from 16- to 32-bit OS/2 will be no more difficult than changes to other 32-bit APIs (such as a future 32-bit Windows API), when they appear. Windows 3.x is based on an inherently 16-bit segmented model; unlike OS/2, it did not have the benefit of being designed with 32-bit migration in mind.

OS/2 - a 32-bit API - TODAY

Some arguments against OS/2 rest on claims that OS/2 is not a full 32-bit system, or that other DOS-based alternatives provide a 32-bit system today. Such claims are based on misconception. The essence of whether a system is 32-bit or not, does not lie in how its internals are coded, but in the benefits it offers to users and programmers. In fact, even to developers, the key issue is not what the system looks like internally, but what kind of programming interface is offered. And, in this respect, OS/2 already offers a full 32-bit API, which has been designed to offer the maximum ease of migration from 16-bit (see above). The OS/2 32-bit API enables developers to escape from the limitations of 16-bit segmented memory management to a flat memory model, but also offers multi-threading, advanced graphics support, and powerful interprocess communicatons. The OS/2 32-bit API is also enabled to accommodate extensions like multimedia and pen-based computing. It therefore opens up the future path not only for multimedia and pen, but also portability to RISC, and for increasing use of object-oriented technology, distributed networking and other developments.

In contrast, Microsoft offers today only a 16-bit API for Windows 3.x. The only 32-bit function currently available is the sparse set of functions in the WINMEM32.DLL, which even Microsoft itself warns, requires advanced Windows and assembly level programming skills. Applications coded in this way may not be easily portable to other processors, if extensive use is made of Intel assembly language. Since WINMEM32.DLL is a set of eight 32-bit function calls grafted on to a 16-bit DOS extender, the 32-bit code cannot make calls to 16-bit Windows or DOS functions, and therefore the WINMEM32 program has to create its own interfaces between the 16- and 32-bit code segments, with address and parameter translation, ie the application must implement its own thunks, and they must be implemented in assembler. This is in contrast with OS/2 2.0, which is designed to allow easy mixed model programming, and provides thunk controls to move between 16- and 32-bit modules. This makes developing 32-bit applications much easier, and much easier to migrate from 16-bit.

Microsoft has announced that when Windows/NT ships, it will implement a 32 bit API, referred to as Win32. Publicly available specifications of the Win32 API show it to be very similar in principle to the OS/2 32-bit API, with few extra features. But since Windows/NT is not scheduled to ship before the first half of 1993 at the earliest, Microsoft has announced a subset of the Win32 API, Win32s, will be available for developers working on Windows 3.1. This is understood to offer flat memory model programming (although details of the thunking that will be required are as yet unavailable), but will not offer advanced features like multi-threading and advanced graphics, which Microsoft says will not be available until Windows/NT appears. Industry commentators have pointed out that the result of this is that developers will have to consider three different Windows APIs:

- Win16 (Windows 3.x)
- Win32s (Windows 3.1)
- Win32 (Windows/NT)

and have to consider how they can design programs across three APIs, which can meet the stated Microsoft goal of scalability across the various future Windows offerings.

On the other hand, OS/2 is available ***today***, and the 32-bit API has been available to developers for several years now, offering all the features promised across the three separate future Windows offerings, in one single API. And the target platform has sold over a million units, several months before any alternative is due to appear.

Internally, OS/2 is a hybrid 16-/32-bit system. The majority of the system code (including most of the kernel, VDDs and the Workplace Shell) is 32-bit. Some other parts are obviously 16-bit as they aim to provide compatibility with older 16-bit software: for example, WIN-OS/2 is a modification of 16-bit Windows code, and is therefore as 16-bit under OS/2 as it is under DOS! Also, OS/2 maintains 16-bit code where it has been designed to accommodate 16-bit modules and DLLs, offering support for mixed 16- and 32-bit code that is not apparent in Microsoft′s stated aims for Windows/NT. Device drivers and parts of the file system are also still 16-bit. Other 16-bit parts would gain little benefit from being adapted to 32-bit (such as the command line utilities like TREE and SORT). In general, OS/2′s mixture of 16-bit and 32-bit code is well suited to the range of today′s applications where 16-bit calls predominate. It remains to be seen how well Windows/NT will be optimised for the 16-bit applications which will still be the majority (especially as it will not be able to run any of the OS/2 32-bit applications which will comprise the majority of 32-bit applications when NT finally ships). As 32-bit applications grow in popularity, OS/2′s mixture will become more 32-bit.

But it must be stressed that the internals of the system are irrelevant to both users and programmers. Programmers care about a 32-bit API: OS/2 offers a shipping product with a 32-bit API; Microsoft only currently supplies a beta version of Windows/NT. Users care about compatibility and performance, and OS/2 provides both of these within its design. Where 32-bit can offer potential benefits (eg performance or future portability), the system components either have been moved to 32-bit or will be over time (for example, the 32-bit PM graphics engine and screen device drivers, which will be delivered by the end of 1992 as an update pack). On the other hand, though Microsoft claims that Windows 3.1 has significant 32-bit code internally (though only when running in 386 enhanced mode), it does not offer any 32-bit API nor multi-threading, so the benefits of 32-bit are only partly realised.

One of the key issues in the “32-bit” debate is portability to and from other processor families such as RISC. But such portability for OS/2 or Windows requires two things: the operating system kernel and subsystems, and the applications. If the API is not 32-bit, there is little point in having a portable kernel, since the applications cannot move without the API. OS/2 has a mature 32-bit API ***TODAY***, and it has been available for over 4 years (since the original Microsoft OS/2 2.0 SDK). This API was designed to accommodate the future directions of OS/2, including portability to RISC, a promise we made together with Microsoft in 1989. Although Windows 3.1 contains some 32-bit modules internally in some of its components (eg FastDisk) some of these are so Intel-specific that moving them to RISC will be difficult, if they are attempted at all. So once again, the issue of the internals of a system is an irrelevance, and moreover a distraction from the real issue: the key factor in a 32-bit design is balancing the aims of portability and performance, but most of all, ensuring that applications can be delivered to exploit the system via the 32-bit API. Today, OS/2 delivers such an API, with features that are only promised for the future by Microsoft.

Workplace Shell

The new user interface for OS/2 2.0 is one of the more obvious ways in which the new OS/2 is better than before. But because it is a new interface, some aspects need some discussion as to why they are new and what benefits they bring.

Why another user interface?

The new look of OS/2 2.0 sets it apart visually not only from previous versions of OS/2, but also from other GUI environments on Intel-based PCs, such as Windows. To some, a new look presents opportunity; for others, who do not like change, the difference can seem a problem. It is worth examining some of the reasons why OS/2 2.0 presents a new look and feel, rather than just continuing with the OS/2 1.x look.

- ***Move to the future***

  OS/2 is not only a system that protects past investments in DOS, Windows and OS/2 1.x, but is also a platform for the future. It aims to provide the kind of object-based interface that will be the norm in tomorrow′s object-oriented systems, but can deliver this function today, since 386-based platforms have the power to support it. This is the new look defined in IBM′s Common User Access (CUA) 1991 guidelines for the "workplace" model. But CUA is not just a set of rules - it is based on a ***vision*** of how computers can really become more useful and useable. Ask your IBM contact about the "CUA Vision" disk-based presentation and video (see "CUA Vision materials" on page 135), to see where the workplace model is heading in future, and what sort of applications can be developed.

- ***Need to attract new users***

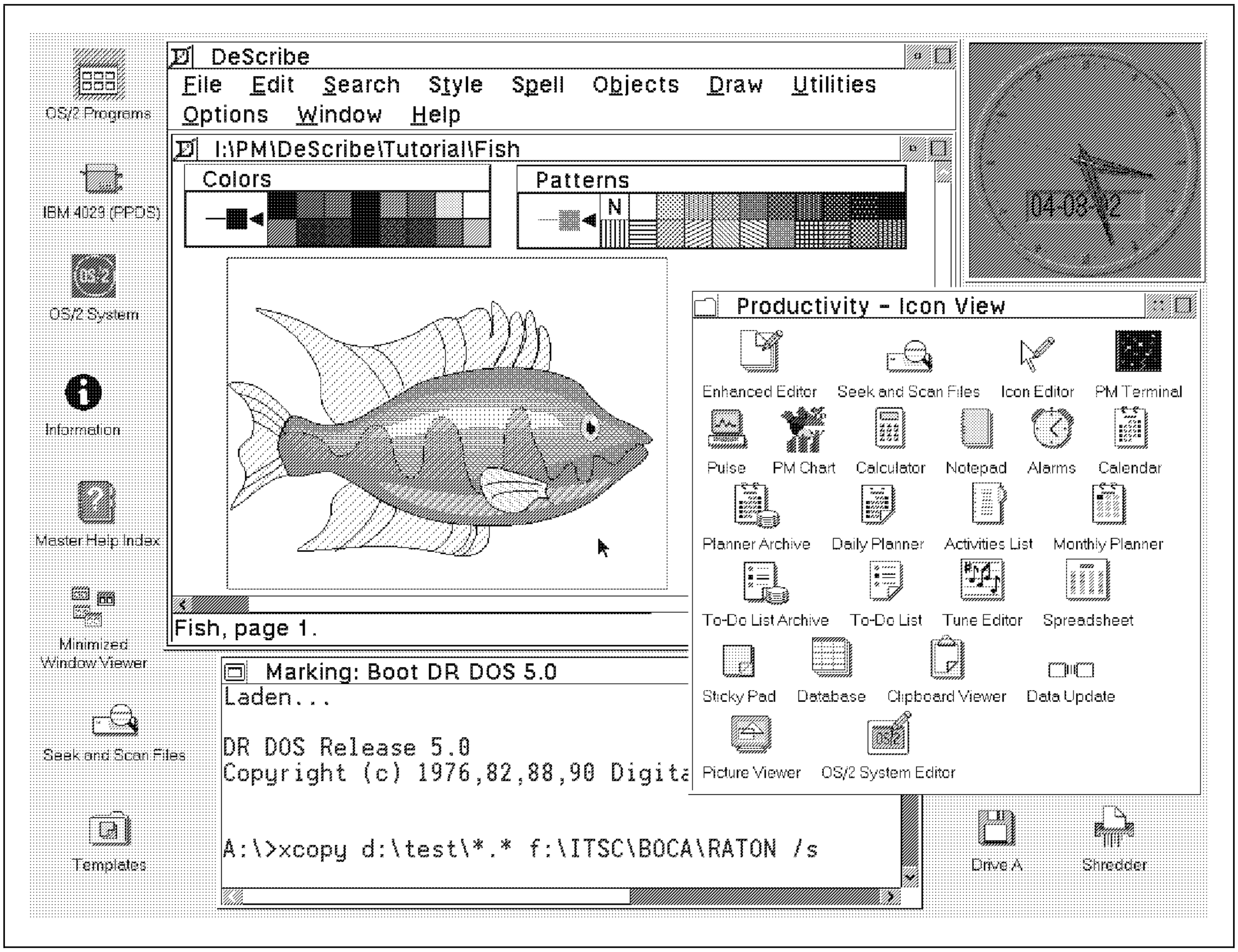
  The PC marketplace is beginning to slow in growth. Vendors end up selling more hardware and software to the same people, and IS departments find it difficult to broaden the base of computer usage. That is because apart from the pioneers and early adopters, PCs are still too difficult to use for too many people. Current users would like to do more with their existing systems, but are constrained by the difficulties of learning so many different applications, each with their own unique way of working. And the fact that we are finding neither new uses nor new users for PCs, means that the return on our investment is limited, and the benefits are not being fully realised.

- ***Focus on INFORMATION, not the computer***

  Too much of today′s use of PCs requires the user to know a lot about the way the computer handles the data. Even GUIs like Windows and OS/2 1.x are basically a graphical representation of the same old computer-oriented way of working. Instead of worrying about different and inconsistent "managers" (File Manager, Program Manager, Desktop Manager), with some handling files, others programs etc, the user can focus on the information he wants to work with, and let the system worry about file and programs. You can have icons representing a report, and have the system associate that with a given application (such as a word processor) so that clicking on the icon loads the application. In this way, programs become tools to achieve the real desired result - working with **information**.

- ***2nd generation GUI***

  OS/2 2.0′s user interface, and the design principles behind it, is the result of more than five years of analysis, prototyping and testing in IBM′s usability laboratories. The results of the tests indicate that an information-oriented user interface like the Workplace Shell is more productive and easier to pick up for first time users. Once the basic principles are learned, even experienced users of old GUIs like Windows, find they prefer the Workplace Shell, since more complex tasks can be accomplished easier and it is more flexible. It is an interface that builds on the achievements of "first generation" GUIs like OS/2 1.x and Windows, and moves users on to the "second generation."



```
Laden...

DR DOS Release 5.0
Copyright (c) 1976,82,88,90 Digita

A:\>xcopy d:\test\*.* f:\ITSC\BOCA\RATON /s
```

Figure 18. OS/2 Workplace Shell

- ***Harmonise different user interfaces***

  People who have been using PCs for some time will have acquired knowledge of many different types of user interface: DOS applications have very few standards in user interface terms, and even Windows adds another dimension and another set of standards to learn. The Workplace Shell does not eliminate these differences (it must retain them for compatibility), but does offer a user interface layer above the individual differences, to provide a level of consistency and integration. For example, all applications, whether DOS, Windows or OS/2, can:

  – be launched from icons on a common desktop
  – appear in windows which can be sized, moved and hidden
  – share data via a consistent set of commands (Mark/Edit/Copy/Paste)
  – retrieved from the background, or closed via a common Window List

  The migration towards future object-based systems such as those based on the Taligent venture (see "Object-oriented environments" on page 113), provides the potential for convergence of ideas from the Workplace Shell, Motif, and the Apple Macintosh desktop.

- ***New, but evolutionary***

  This progress can be made now without compromising compatibility with the past. The old ways of working with the PC can be kept alongside new ways:

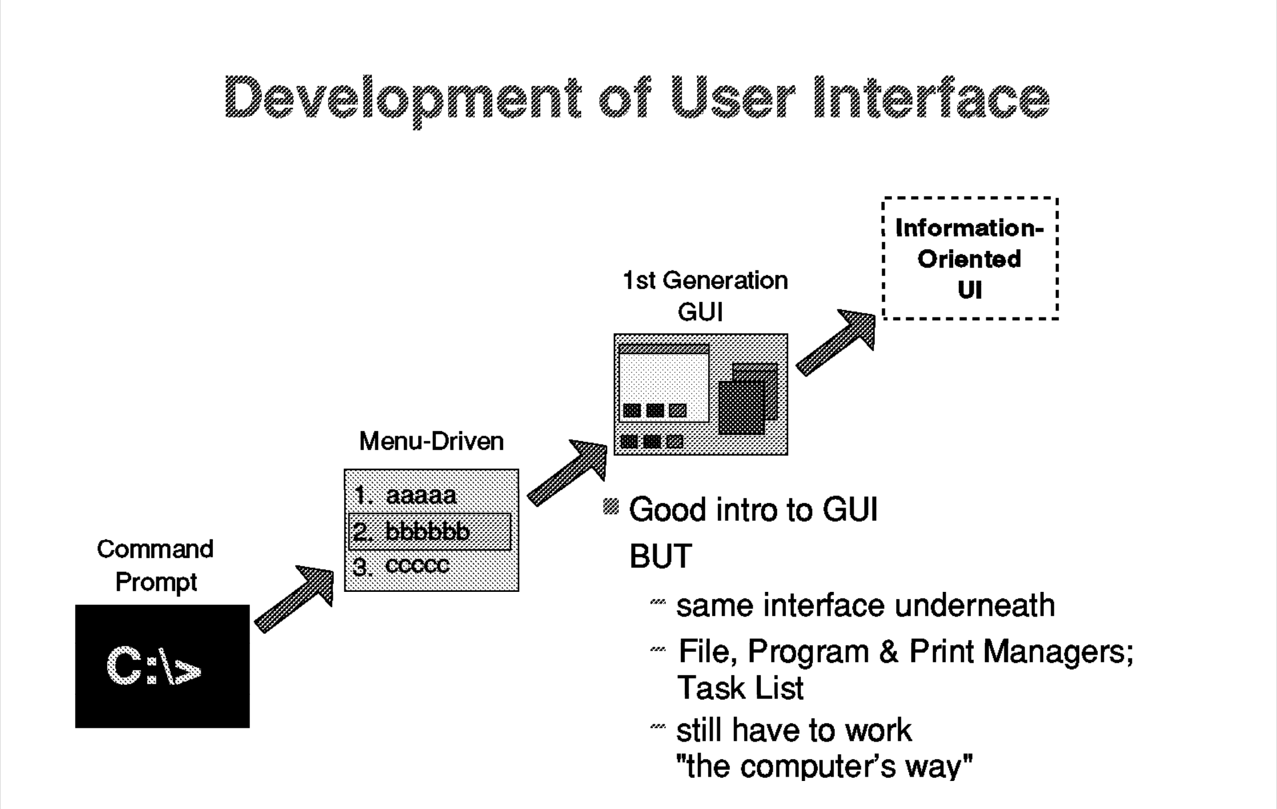
– the C:\> command prompt can be retained as an icon or even a menu item from the desktop context menu.
– you can use folders just like Windows 3.x or OS/2 1.x groups, and use the icons within them just like programs, thus treating the Workplace Shell like the graphical program loader that Windows is for many users.

This allows the user to take what he knows and apply it immediately, while learning new skills that will make his work more productive. This mix and match of new and old knowledge makes evolution towards the 2nd generation GUI much easier.

The key is to provide compatibility, but not to let the past bar your way to the future.

An INFORMATION-oriented user interface

The first generation GUIs like Windows and OS/2 1.x have been a good introduction to the benefits of GUI for many people, but they are also a constraining factor on the progress of the man-machine interface. This is because the user interface design relies heavily on computer-oriented concepts like files, programs and directories. The user interface (UI) model is built around various “managers”: File, Print, Program and Desktop. To navigate through the system, the user must know about the difference between a file and a program, about the physical layout and organisation of data on the disk, and to understand some of the constraints this imposes (for example, you cannot put files in a program group, only programs, but programs can be loaded from the File Manager). There were other usability constraints which related more to the implementation of the user interface, rather than the design itself. In the Windows 3.0 and OS/2 1.3 File Managers, only one directory tree window could be opened, even on a system with multiple drives. The Print Manager only interacted directly with the File Manager via drag and drop, but other applications either could not, or implemented their own drag and drop protocol.



Development of User Interface

- Good intro to GUI

BUT

- same interface underneath
- File, Program & Print Managers; Task List
- still have to work "the computer's way"

Figure 19. Development of user interface

Moreover, this style of user interface imposes an application-oriented way of working, or an "action-object" paradigm: a user wanting to create a report will load his word processor, then look for the file containing the report. The ideal for the user would be to encourage an ***information***-oriented way of working, or one that uses the "object-action" paradigm, as this is the way we work more naturally when away from the computer. Moreover, it is important not to forget the computer is an item of ***information*** technology. It is the information that the user wants to work with, that is why he uses the computer in the first place.

The Workplace Shell does not make the user look for the word processor to create a report, but rather allows him to click on an icon representing the report, and automatically load the application (the word processor) associated with it. This use of the object-action paradigm, and the focus on information, is why the Workplace Shell is sometimes called an ***object-oriented user interface (OOUI)***. It means that you create a letter by dragging a new copy from a "template" (see "Templates" on page 82), and regard the application as a tool to work with the information, rather than a program which you must "feed" with data.

The new shell is designed to provide a more task-oriented, not process-oriented way of working, allowing the user to focus on what they want, not how to do it. The new shell will also reduce the amount of system-specific knowledge needed, by being more analogous to the manual way of performing tasks (using the physical desktop analogy).

Notice, however, that the new object-oriented way does not preclude the old ways of working. If you need the C:\> prompt, it is available; if you want folders full of program icons, which you click on to load, and then go through the "File Open" menu to find your data, that can be done too. You can mix and match the action-object and object-action techniques as you wish. But even with older DOS programs, some of the benefits of an OOUI (such as clicking on an icon representing a report, and having the system load the appropriate application), can be realised very easily without a radical change to working style. In fact, even if the Workplace Shell is used as a graphical program loader, there are features in it (such as workareas) which make it a better one than Windows (see "Multi-tasking and the user interface" on page 29).

Workplace Shell components

Not only do the design principles of the Workplace Shell differ from the older GUI models, but it is also visually different, and it includes a number of new features that did not appear in previous versions of OS/2.

User interface elements

In this section we will look at some of the user interface elements that contribute to the new look and feel.

Desktop

If you are familiar with OS/2 1.x or Windows 3.x, the first thing you will notice about OS/2 2.0 is that there is no obvious Desktop Manager, Task List or Groups. Instead, the screen represents your desktop, and everything on the desktop is an object - files, devices, programs etc. This means the desktop is the background of activity, and items can be placed on the desktop or in folders - in fact, anywhere you want; there is no restriction on where icons are placed as there is in Windows or OS/2 1.x. This means you can have either a "tidy" or a "messy" desktop according to the way you work. Although the desktop is meant to give you a visual association with a familiar idea (your own desk), the Workplace Shell desktop is not meant to be exactly the same as your real desktop; it offers a bit more than your real desktop. After all, if the computer only did the same things as you could do manually, why use a computer at all?

Objects and folders

The items on the desktop are objects: files, programs, devices (such as disk drives, the shredder and printers). Objects reside in folders or on the desktop (which is actually just the highest level folder). Folders are used instead of group windows, and are more powerful and flexible. Folders can contain any object, including other folders (again, just like the physical desktop), and can group items according to a given project or activity (features like workareas take this even further - see "Multi-tasking and the user interface" on page 29) You can also have more than one view of a folder, allowing you to look at the same information from different perspectives ; a simple example is in the Drives object, which lists the files by drive and directory; this allows multiple simultaneous views - Tree, Details, Icon views - of the same directory.

Each object is visually represented by an icon. You work with objects by ***direct manipulation*** - ie by pointing at it with the mouse (see below). Notice that sometimes only the terminology is different: what OS/2 2.0 calls a "Program object," Windows and OS/2 1.x might have called a "Program reference" or "Program icon."

Direct manipulation

Direct Manipulation is the act of working with an object by pointing at it with the mouse, double clicking to load it or open it, or dragging it somewhere else. Since most older user interface models worked by an action-object paradigm, experienced computer users are used to a certain level of indirection in using a computer (ie you go through a menu to invoke a command to work on an object). But the Workplace Shell allows users to work with objects more naturally, as they would on a real desktop - directly. So, although experienced users may be tempted to think of direct manipulation as a gimmick, it is in fact a much more natural, consistent, and, in most cases, effective way of working with the system.

A specific example of direct manipulation is drag and drop. This is where one icon is "picked up" with the mouse and moved somewhere else, often to another icon or folder. Note that this use implies that objects can be programmed to understand what it means to have something dropped on them, or for them to be dropped on something. In fact, there are APIs supplied for programmers to set the behaviour of their own icons (eg for selection, dragging and dropping).

Use of these APIs allows greater consistency between the shell and applications, and also greater integration. Developers can now register their objects (ie applications) with the system and have them used as an integrated part of the system. This all results in a more powerful and consistent interface for the user.

Although Windows 3.x offers some direct manipulation facilities and some APIs, its use in the system is currently limited mainly to the File Manager. Interaction between different parts of the system is therefore more limited than in OS/2 2.0, compromising the benefits of consistency.

Descriptive names

Objects and folders can be given any descriptive name. For example, the user may have a folder called "Annual Report to the Shareholders," which may contain an icon called "Shareholder Report Draft 14 dated 6th March." This may be a document created using a Word Processing program. Another icon may represent a spreadsheet containing the latest financial results. Notice that icons can represent data files, and to edit them, the user can simply click on the object to invoke the application that created that file, and edit it further. Even though files are physically stored by their old 8.3 file names (at least using the FAT file system), they can be represented on the desktop or in a folder by the more descriptive long name. It is easier to see that "Letter to Joe" would be more meaningful to any user (even the creator of the file!) than LJB0905.TXT. This feature is implemented at the file system level, so that even data files relating to old DOS applications can be given long names in this way - applications do not need to be rewritten.

The use of descriptive names is in keeping with the principle of allowing the user to focus on the information, not the computer′s way of working. In this way, OS/2 offers more fexibility than Windows, in which the computer-oriented way of working still shows through in the retention of 8.3 file names, despite the graphical veneer on the surface.

Associations

Associations can be created between files and programs, on a file-by-file or wild card basis. This means that simply by double clicking on a data file, it can be loaded into the associated application. OS/2 applications can also create their own file types and associations which are created automatically when the program is installed (DeScribe and Excel for OS/2 are just two of the applications that do this already). Associations can also allow a data object to be dropped on a program icon and have the program loaded with the data file. Multiple associations can be created per object.

Shadows

Since object icons can be placed anywhere on the desktop or in folders, this flexibility encourages different arrangements of programs, data and devices into "projects." This may mean that the user wishes to place a given object in more than one folder. Instead of physically copying an object from one folder to another, which can waste disk space and also cause maintenance problems if the object is subsequently changed, the Workplace Shell allows the user to set up a "shadow" of that object. (Apple Macintosh users will be familiar with the idea which they call an alias.) The shadow is linked to the original object such that changes in the original are rippled through to the shadows.

Shadows allow different uses of the same object, and allows work to be organised without regard to the physical location of data (they are particularly useful for working with data or devices located on a server). In this respect the combination of folders and shadows is much more powerful than the use of group folders and specially created directories in Windows or OS/2 1.x.

Templates

Templates are a means of easily creating new objects. Most objects can be defined from an existing type, or template. The system provides templates for the common objects such as data files, program objects, folders, printers and so on. For example, program objects are created by dragging a program template from the Templates folder, and entering the information about the program executable and working directory.

But users and programmers may define their own templates. Programmers can create new file types (eg a Customer Invoice) and define templates via their own applications. Users can create templates themselves. For example, a "memo pad" template can be created, to use to create new memos, each of which may be automatically associated with a chosen word processor, and would contain a department logo, and headings for the "To:," "From:" and "Subject" parts of the memo. Or a chart template could be created, so that each new chart created inherited the default characteristics: a logo and given fonts and colours. Since many business processes are repetitive, templates match many users′ work very well. Any kind of information that has the same basic structure in each instance, and only the specifics beneath the "headings" change, is suitable for creating templates. Templates provide a consistent way of creating new objects, whether data, program, folder or device.

Pop-Up menus

Clicking with Mouse Button 2 on an object′s icon, or on an open folder, displays a menu of options to work on the object. These pop-up menus display only the actions appropriate to the current state of that object. This reduces screen clutter and increases ease of use by eliminating redundant or inapplicable options. This menu can be customised by the user (for example, to change the application loaded from the “Open” menu, or to add user-defined commands).

The use of Mouse Button 2 is one example of the UI innovations that has received some criticism in the computer press, though it is not clear why. Indeed, it is somewhat ironic that when Borland′s Quattro Pro for Windows was reviewed in beta test, one reviewer praised its′ use of Mouse Button 2 as a “Property Inspector.” In fact, the Property Inspector provides very similar functions for a spreadsheet or graph element (otherwise known as an object!) as the Workplace Shell′s pop-up menus do for object icons. While it is likely that Borland′s UI designers came up with the idea independently, it is strange that what is a virtue from one vendor can be criticised in another. It is an illustration of how perception, rather than objective evaluation, is a danger when considering user interface design. (At least there is now the potential for some consistency in the use of Mouse Button 2 to work on an individual object - previously it was not used at all under Windows or OS/2.)

Visual Clues

It is important, for the user′s comfort and satisfaction, that clues are supplied to aid learning and provide a context for what the user is doing. A major principle in the Workplace Shell is to provide visual feedback where possible, to keep the user informed about what is happening. These are often ignored by the casual reviewer, but they contribute to the usability of the system. Among the examples of such visual clues are:

- change of mouse pointer between clock icon, normal pointer, I-beam for editing, according to the context
- half-toned icon during copy of object (as opposed to normal tone for a move operation)
- a line drawn between a shadow and its original during the shadow operation
- a box drawn round an object which would be the target of a drag-drop operation
- shading behind the icon of an object that is in use
- “no-entry” sign on an icon when it cannot be dropped on a given object
- highlight of the object or objects (by darker shading) which are currently selected (and for which the current pop-up menu is valid)

These visual clues are more widespread and more subtle than in less sophisticated GUIs such as Windows, where it is sometimes difficult to determine the context of an operation or an object, because of the lack of visual clues. (Consider, for example, how the new user reacts to an icon in the Program Manager and another, for the minimised application, at the bottom of the screen, with no apparent difference between them, yet they behave differently.)

Consistency

One of the most important aspects of the Workplace Shell is its consistency. Not only does it provide a consistent means of loading programs from different origins (DOS, Windows, OS/2) and make them work together, but it also provides a standardised interface to different tasks, through the use of drag and drop. For example, OS/2 allows drag and drop to be used for copying, deleting and printing a file, where DOS uses three different commands (COPY, ERASE, PRINT), each with their own set of parameters. As stated before, it also provides a consistent interface (Mark/Edit/Copy/Paste) to sharing data between applications, whether DOS, Windows or OS/2 applications. And drag and drop is applied to many other actions throughout the shell, such as setting colours and fonts. This consistency is reinforced when using programs that integrate with the Workplace Shell and use the same manipulation techniques (see “Workplace Shell exploitation” on page 73). The consistency of the Workplace Shell (where drag and drop are pervasive) can be contrasted with the rather superficial use of such techniques in Windows 3.1. Consistency is not achieved where drag and drop can only be done from the File Manager, and not throughout the shell.

Consistency means that not only is the environment easier to use, but new actions are learned quicker, often by experimentation. In fact, the key to judging a user interface is not how easy it is to do something in an hour or a day (after all, no user interface is completely intuitive), but what extra can be achieved after the basics have been learned, and how easy it is to do more.

Flexibility

The user has complete freedom to set the look and behaviour of the shell. Colours, fonts and even background images can be set for the desktop, and for each folder individually. Icons and descriptive text can be set for each object, and even the behaviour of objects and their windows (such as whether a window is hidden or minimised to the desktop) and the use of mouse buttons can be customised, where appropriate on an object-by-object basis. This flexibility is important, because users rarely agree on what is preferable. The Workplace Shell reinforces the truth that the PC is a ***Personal*** computer.

Controls

As well as providing a different way of working, the new look and feel of the Workplace Shell is established by a number of new user interface controls, including the following:

Container described above as folders, used to logically group objects on the desktop. It can provide multiple views of the objects (icon, text, tree, details).

Notebook an easy way to navigate through a complex dialog. It looks like the paper notebook you may use at your desk. It supersedes separate dialog boxes by providing multiple pages, selectable by tabs. The notebook control is used to allow the user to tailor the settings of each object.

Slider allows the user to select a quantity from a range of possible values, by using a control very similar in appearance to that found on many electronic devices - a “sliding” button.

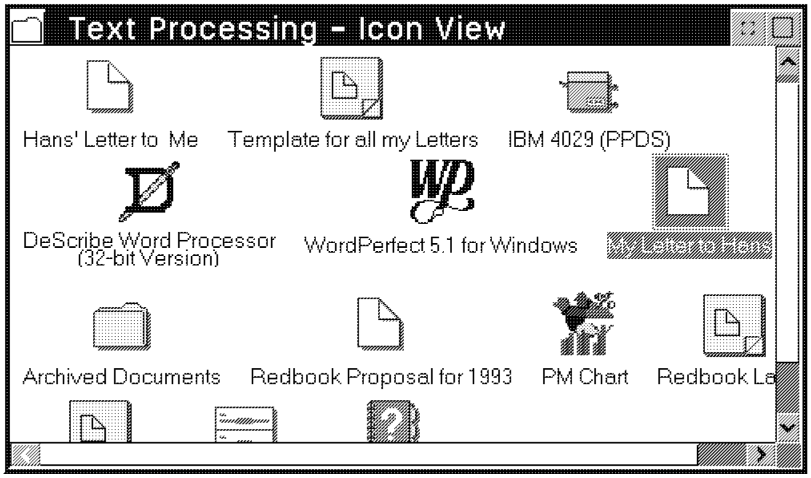

Text Processing - Icon View
Hans' Letter to Me
Template for all my Letters
IBM 4029 (PPDS)
DeScribe Word Processor (32-bit Version)
WordPerfect 5.1 for Windows
My Letter to Hans
Archived Documents
Redbook Proposal for 1993
PM Chart
Redbook La

Figure 20. Container

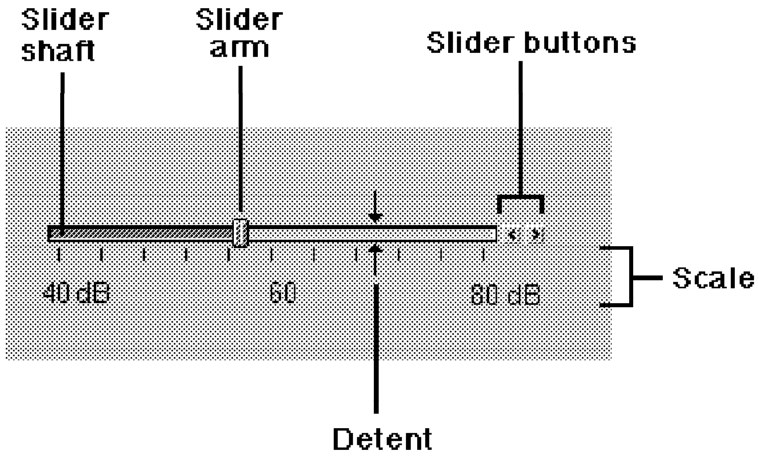
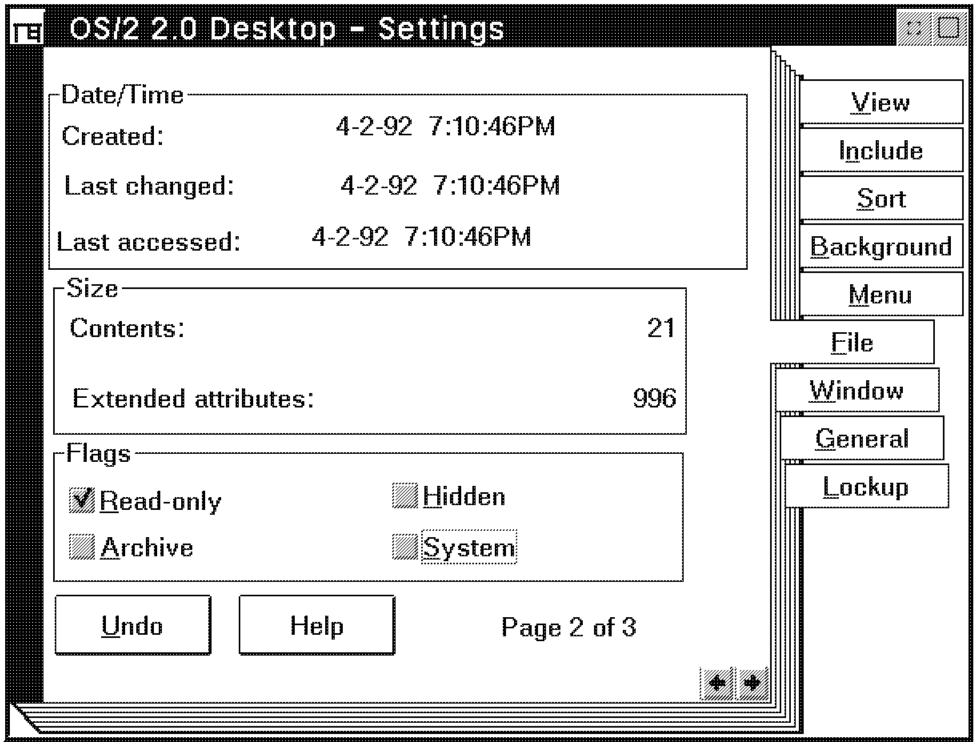


Figure 21. Slider



OS/2 2.0 Desktop - Settings

Date/Time

Created:	4-2-92 7:10:46PM
Last changed:	4-2-92 7:10:46PM
Last accessed:	4-2-92 7:10:46PM

Size

Contents:	21
Extended attributes:	996

Flags

☑ Read-only ▨ Hidden
▨ Archive ▨ System

Page 2 of 3

Figure 22. Notebook

There are also standard dialogs for open and save file, and font selection. All of these controls have programming interfaces, so that developers can use them in their own applications, to allow a common look and feel between applications and the shell.

Applets

OS/2 2.0 also ships with a number of mini-applications and utilities, ("Applets") which give some basic functionality to get started immediately, as well as acting as a learning aid, particularly for manipulating objects with the mouse. (They also provide a bit of fun!). These range from productivity applications like an editor, simple spreadsheet, calendar and card file, to a charting program and terminal emulator. There is also a utility that allows the user to search through the disks for files matching a file specification or even a given item of text. Of course, there are also a number of games including an OS/2 version of Solitaire that allows you to cheat if the game is not going in your favour!

Applets are not intended as full function applications, but as a means of getting productive use of the system even without installing extra software. However, many of them are powerful enough in their own right to serve the occasional user of, for example with PM Chart, a charting package, without needing to invest in more software. They help to make the system appealing to both the first time and the less experienced user.

Extra facilities

There are a number of extra features contained in the shell to help overall productivity, or improve ease of use:

System Setup This object allows the system to be easily configured and changed to suit the individual. Not only does it contain options for installing new features or adding drivers, and for setting colours and fonts, but also for modifying some of the default behaviour of the system (eg whether windows are hidden or minimised to the desktop; whether clicking on an icon creates a new instance or retrieves a running instance of a program; whether to prompt when deleting an object, and so on).

Drives This object replaces the File Manager from previous releases. It provides similar function to the OS/2 1.3 File Manager. It has a multi-threaded design to give good performance even in large directories. It is no longer restricted by the Multiple Document Interface (MDI)[20] design of the OS/2 1.3 File Manager (which is still evident in the Windows 3.1 File Manager), but instead creates a series of modeless[21] windows, which can be moved wherever the user wishes. Drives offers all of the function commonly used by the File Manager in either Windows 3.x or OS/2 1.3, with the addition of greater flexibility in certain operations. For example, in keeping with the rest of the shell, it offers different simultaneous views of any drive or directory, either icon view, tree view, or details view (the latter lists file size, creation date, and other details). Since the windows created by the Drives object are just like any others in the shell, files can be dragged between the desktop and different directories, and to the printer or any folder.

[20] MDI means that multiple child windows are contained within the bounds of the parent window, and no child window can be sized beyond the bounds of the parent. The Windows 3.1 File Manager and the behaviour of the directory windows within it, provide an example of MDI.

[21] Modeless (as opposed to modal) refers to the behaviour of child windows in a GUI. Modal windows have to be closed or the dialog within them completed before the parent window can be accessed. They tend to enforce a particular pattern of progressing through several windows (an example is in the use of error dialogs, which tend to be modal.) Modeless means that the user may move from the child window back to the parent without closing intermediate windows or dialogs. An example is the "Find" dialog in the OS/2 System Editor.

Printers

The ease with which OS/2 handles printing (with drag and drop) is one of its strongest features, but equally impressive is the flexibility of printer setup and customisation of settings. Different print objects may be set up, not only to represent each physical printer, but also to represent a particular combination of settings: for example, you can set up the same printer to print portrait Times Roman (represented by one icon, which you can give a descriptive name such as “Portrait - for letters”) and create another icon to represent the same printer running in Landscape mode with a small font for those wide spreadsheet reports (to which you give another name). Both these printer objects can be kept on the desktop or in a given folder, and print output can be directed to either according to the results required.

Convenience features

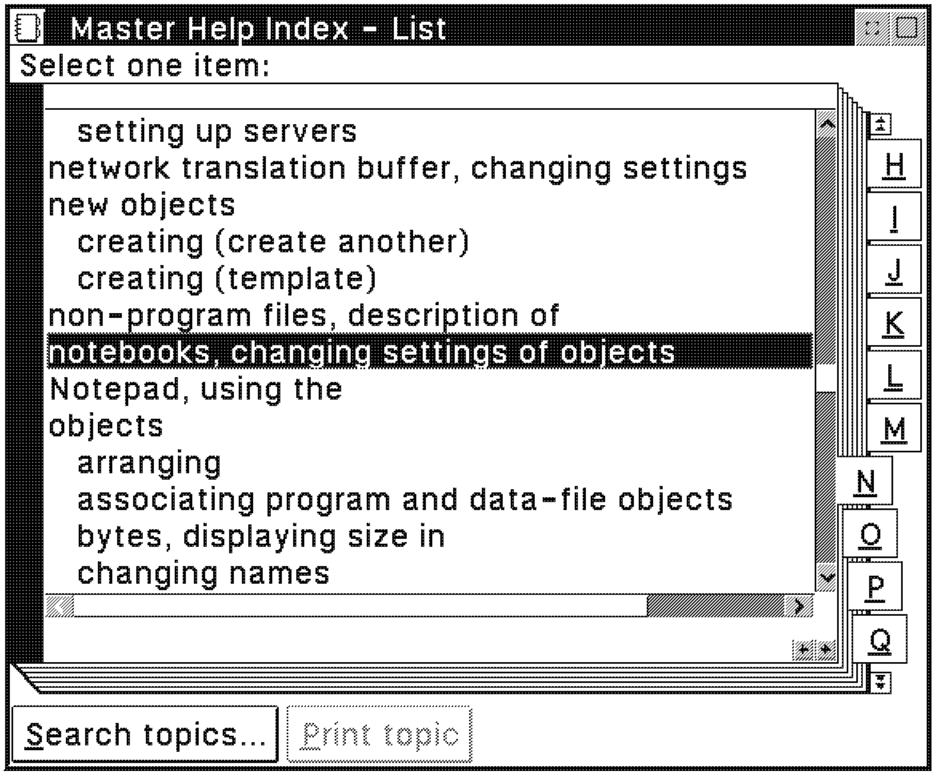
The shell also includes several features for greater user convenience. The user can set his own background (“Wallpaper”) for the desktop, or any folder, as well as a keylock security feature (blanks the screen and locks up the keyboard and mouse after a user-specified period). The desktop layout (positions of folders, applications loaded etc), can be autosaved at shutdown and restored when next starting the system. The workarea feature (see “Multi-tasking and the user interface” on page 29) makes it possible to do this on a folder by folder basis. This means that different projects (or even different users sharing the same PC) can be kept logically separate. All of these features help make the system easier and more enjoyable to use.

Tutorial

The default action after installation is to start the tutorial, so that the first time user is taken through the key elements of using the Workplace Shell. The tutorial is highly recommended, even for an experienced user of DOS Windows, or OS/2 1.x, to familiarise oneself with the differences from the previous environment. The tutorial can be revisited at any time; it usually resides in the Information folder.

Online Help

The tutorial is just one of the items of information that the OS/2 user has available. As well as context-sensitive help being always available through the F1 key, there is also a Master Help Index, which gives a “how to” reference guide to using the system. The Command Reference from version 1.3 is also retained, and there is in addition, a glossary of terms. The ability of the shell to keep multiple modeless windows open, allows the help text to remain on screen while an operation is performed, to guide the user through the process.



Master Help Index - List

Select one item:

- setting up servers
- network translation buffer, changing settings
- new objects
  - creating (create another)
  - creating (template)
- non-program files, description of
- notebooks, changing settings of objects
- Notepad, using the
- objects
  - arranging
  - associating program and data-file objects
  - bytes, displaying size in
  - changing names

Figure 23. Master Help Index

The Master Help Index and Glossary use a variety of techniques to present information to the user: text in different colours and fonts, hyperlinks and even pictures. The Help facility also includes an indexing and searching system, to help users find the help they need. All of these facilities are open to the OS/2 developer as well, using the Information Presentation Facility (IPF). This allows online help to be created very simply from text files containing a tag language. New functions in OS/2 2.0 include the ability to predefine the help window size and its position relative to the parent window; multiple help pages (windows), called viewports; support for multiple fonts, and easy setup of tables of information; “tear off” help pages, allowing a user to retain the help on screen as he follows an example; and hypergraphics - the ability to click on a graphic and be linked to text or more graphics.

All of these facilities are available to application programmers, so that on-line help for applications can be created to have a consistent look and feel with the rest of the shell. Some developers are creating on-line guides for their applications, and by placing them in the Information folder on the OS/2 desktop, creating an “on-line bookcase.”

In fact, the IPF facility can be used even more widely than just online help for a program; in IBM it is used as a general information delivery tool, for online documents and reference guides - sometimes it can be easier to use the computer to search for information and then browse or print it, especially if (as with most reference information) there are a lot of cross-references.

One of the help items added after feedback from the OS/2 beta test program was the “Start Here” object. This was in response to customers who asked that the shell provide the user with a visual focal point, in case, looking at a desktop with no open windows and no menus, he became confused and did not know what to do next. The Start Here object allows a first time user (or an occasional user) to find a focal point to go and retrieve information about common operations. It is not a replacement for the help system, but an easy starting point to find out more.

Adobe Type Manager (ATM)

One of the most important aspects of a graphical environment is, of course, that it continue to handle text well, and provide the benefits of different fonts and typefaces. OS/2 2.0 continues the innovation of OS/2 1.3, in including the Adobe Type Manager (ATM) as an integrated part of PM. ATM provides scalable font technology, to display and print high quality type on a variety of screens and printers (not just PostScript printers, but laser printers, inkjets and even dot matrix printers). ATM is closely related to Adobe′s PostScript printer language, and both use the same Type 1 font format. This means that companies with an investment in PostScript on printers or on other platforms (such as Macintosh, AIX, VAX/VMS and IBM System/370) can be assured of a compatible use of fonts in OS/2. ATM is a widely used font standard, with over 12,000 fonts available and over $4 billion invested. It is under consideration as an ISO standard. Indeed, for SAA systems, Type 1 is the standard font format, and included in OS/2 2.0 are 13 Type 1 fonts from 4 families (Times Roman, Helvetica, Courier and Symbol) that will be implemented as “core fonts” across all SAA platforms. Any of the 12,000 fonts available in Type 1 format can be installed and used with OS/2 applications, via the Font Palette.

ATM allows OS/2 applications to render fonts in any size on the screen and printer, retaining the high quality of the typeface. It allows WYSIWYG (What You See Is What You Get) between screen and printer, which is important not only for Desktop Publishing, but even for word processing and spreadsheets. ATM is also available for DOS/Windows 3.x as a separate product, ATM for Windows, but OS/2 includes this utility at no extra charge in OS/2 2.0, so that users running Windows applications may take advantage of Type 1 fonts in their OS/2 and in their WIN-OS/2 sessions. However, in OS/2 it is not a separate utility, but an integral part of the PM system. OS/2 applications can use Type 1 fonts without any changes to the applications.

Windows 3.0 had no in-built scalable font technology, and ATM for Windows was a popular product in that environment. However, in Windows 3.1, Microsoft introduced their own proprietary font technology, TrueType. This is not available on other platforms, except for Apple System 7, and according to Microsoft has 600 fonts available (compared to Type 1′s 12,000 or more). OS/2 has the ability to include support for other font formats such as TrueType, through an open font interface which is available to font providers; such support would usually be provided by the owner of the technology (in this case Microsoft). This would allow TrueType support to be provided in OS/2 in an integrated way, like ATM, if Microsoft wishes to expand the platforms on which TrueType is available. OS/2′s aim is to provide open font support, not be tied to a proprietary font technology.

In summary, the benefits of using ATM and Type 1 in OS/2 include:

- better quality screens and printouts
- WYSIWYG between screen and printer
- improved output even on less expensive printers
- wide choice of fonts
- investment protection in Type 1 fonts and PostScript
- a compatible font technology between OS/2 and WIN-OS/2
- integrated font technology available to OS/2 applications without change
- font portability to other platforms (Macintosh, VAX/VMS, AIX, S/370)

LAN-independent shell

As well as being a powerful and easy interface for working with data held locally, the Workplace Shell has a degree of “LAN-awareness” built in. Although it does not include requester code in the base system, the Workplace Shell is able to recognise the presence of network requester software such as the OS/2 LAN Requester for OS/2 LAN Server, or the NetWare requester for OS/2. It provides an API which other network providers can write to, to integrate more fully with the Workplace Shell. In this respect, it is not tied to any single LAN system, but can provide a graphical view of data held on different networks - it is “LAN-independent.”

OS/2 has provided access to multiple LAN systems at the same time (eg NetWare and OS/2 LAN Server) since OS/2 1.3: not only does it handle the concurrent protocols required (eg NETBIOS, IPX etc) but now, in OS/2 2.0, can offer a graphical view of network resources (files, printers etc) consistent with the rest of the shell. By providing the same constructs for remote as for local data (folders, icons, printer and file objects), data can be moved between one folder and another, without the need to know the precise location of the data or which server software is running. This makes access in multi-vendor networks much easier.

Among the features it provides are:

- ability to login/logout to network servers through a graphical dialog
  - this includes presenting an appropriate login dialog (if necessary) before a network object can be accessed; this means that, for objects that have specific login requirements, prompts for login are displayed, even if the main login is completed.
  - an extra item on context menus for network objects is Login or Logout
  - provides consistent login (though NOT a single login) between NetWare and OS/2 LAN Server - the difference between *LOGIN* (NetWare) and *LOGON* (OS/2 LAN Server) are minimised
- browse available servers and resources (subject to logon viewing permission)
  - A “Network” folder contains an icon for each requester installed (for example, one for NetWare and one for OS/2 LAN Server); clicking on the icon provides a view of the servers available

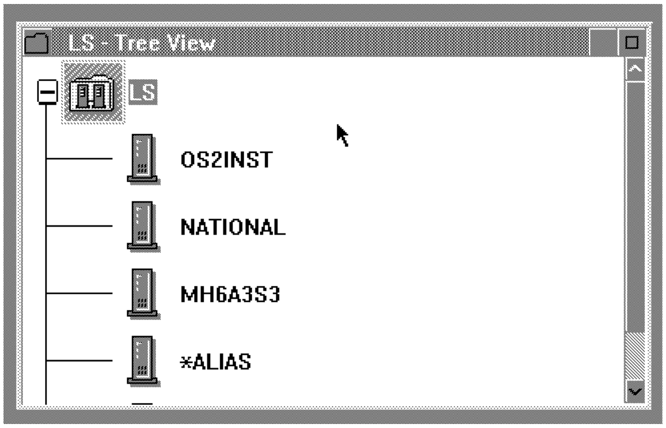


Figure 24. LAN Server Tree view

  - shared directories and printers are given similar icons to standalone objects, but have a mini-network icon to distinguish if necessary, as shown in the folder below. This folder is from a NetWare server, but it is impossible to distinguish between NetWare and LAN Server - that′s what is meant by LAN-independent:

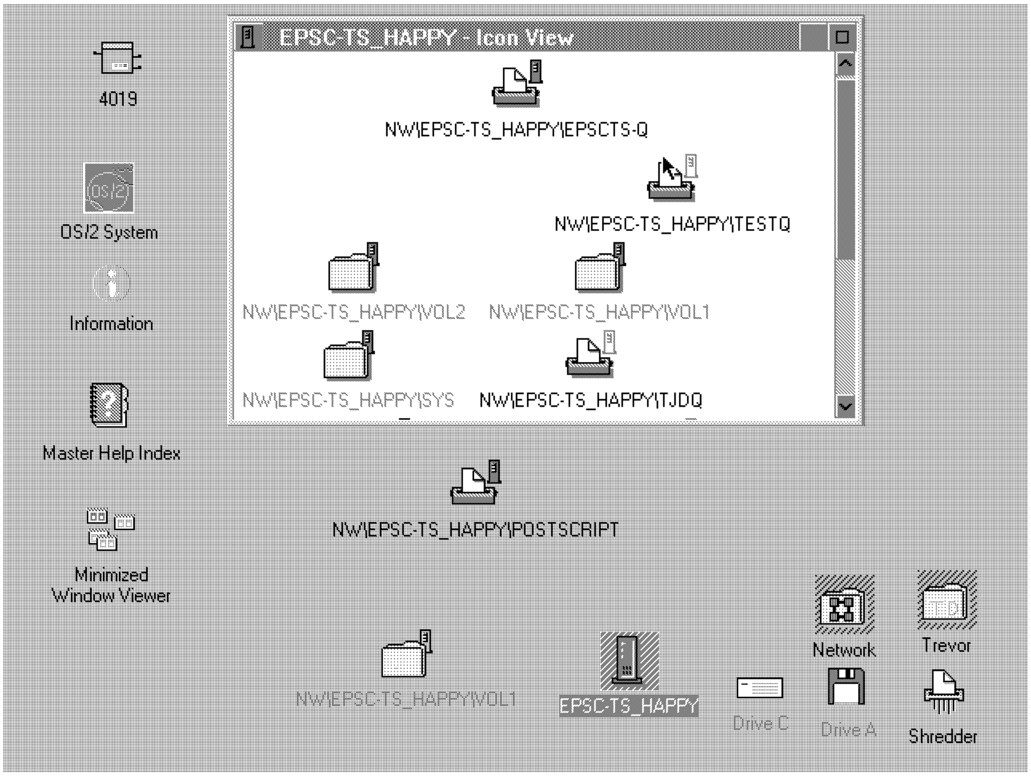


Figure 25. Folder with network resources

- resources can be moved onto the desktop, or any folder, for easy and convenient access
  - servers, shared disks, files or printers can be shadowed into any folder including the desktop (see “Shadows” on page 82)
  - this is not just for the current session, but the icons representing the remote objects will be retained on the desktop at next boot, and the appropriate login dialog presented when the icon is next used
- seamless access to network folders, files and printers
  - remote disk resources can be opened (if appropriate access privilege in force) to show folders and files (behaves just like the Drives object for local data). Programs and data can be used and copied or shadowed from this network disk (without having first to assign the network directory a drive letter such as X:).
  - shared printers are easy to set up (prompts to install matching local device driver if not already installed, and sets up rest of configuration)
  - printer object can be opened to show queued jobs and job status (just like local printer), but user can only manipulate (hold, release, cancel) his own jobs; administrator can manipulate all jobs
  - network printer can be the default (no need for local printer to be defined)

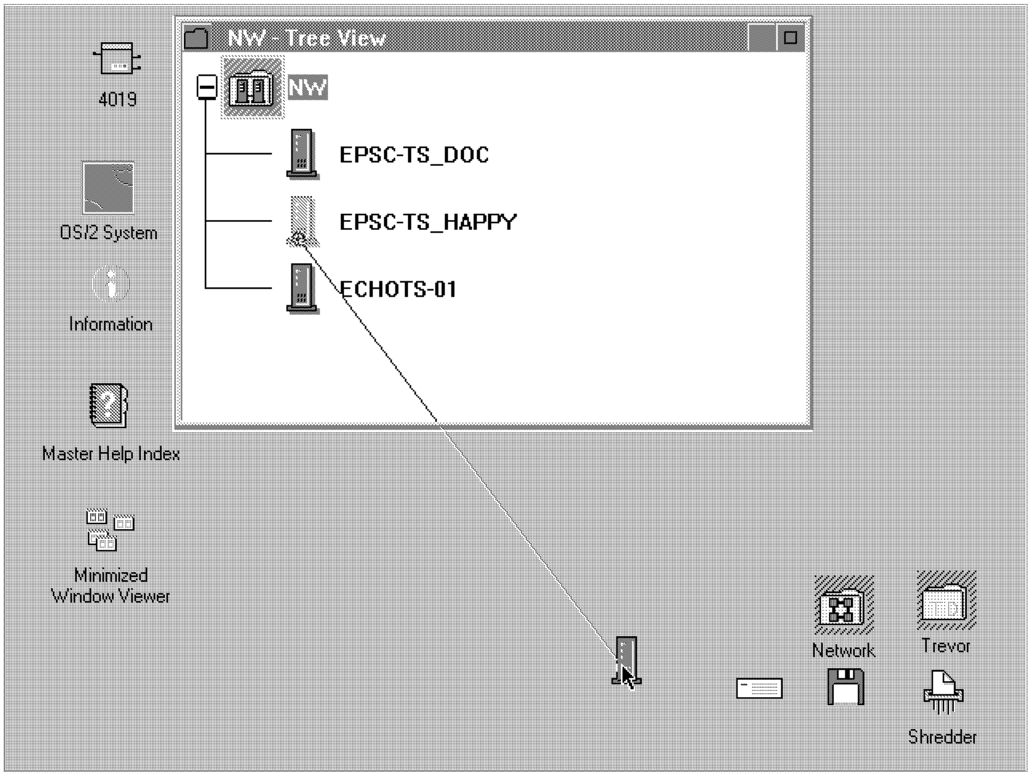


Figure 26. Dragging a shadow of a network resource to the desktop

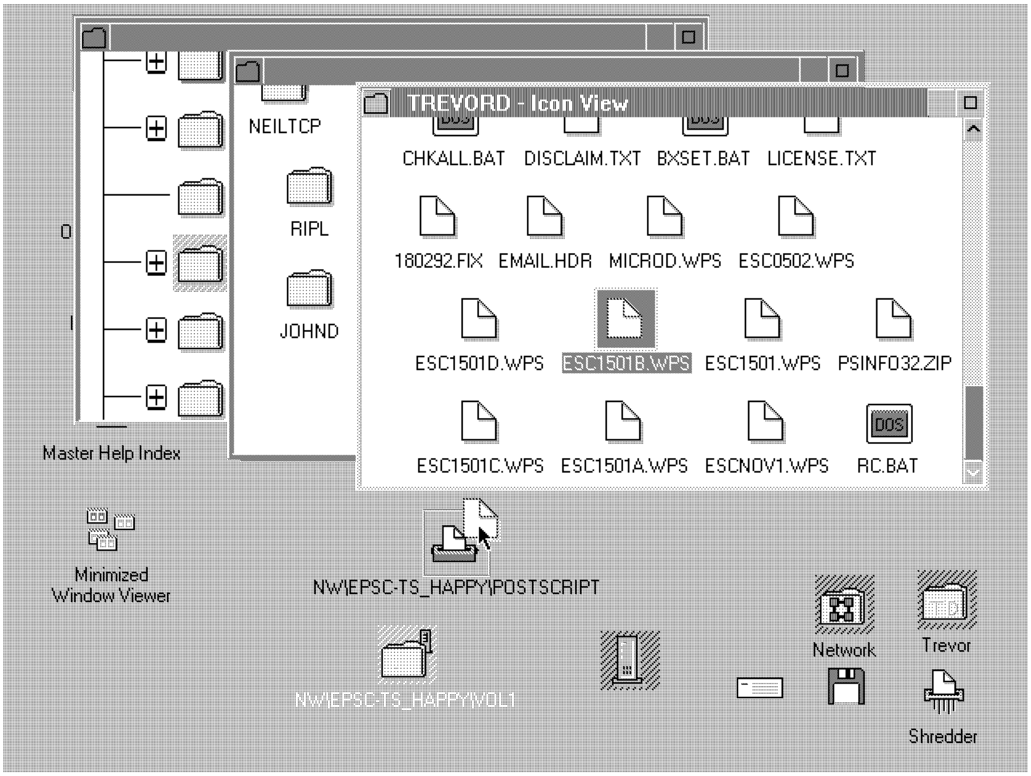


Figure 27. Drag and drop printing on the network

Disk and printer resources set up for use by an administrator will appear as drive objects and printer objects, like local resources. Thus, once a user has logged on, he can see a P: drive and an X: drive, if these have been set up for him by the LAN administrator. In addition, the user can assign his own drive letters and logical ports to network objects that he has access rights to (similar to doing a NET ALIAS command in OS/2 LAN Server).

But the key point of the LAN-independent shell, is that all access is graphical, through the Workplace Shell. There is no need for the command prompt or for character-based menus. This helps to bridge the gap between the LAN and local resources, and provide a more consistent and seamless access to both.

System Object Model (SOM)

One of the most significant elements of the Workplace Shell is something that users never see - the System Object Model (SOM). SOM is an object model - a way of defining objects to the system. It is the foundation of the object-oriented design of the Workplace Shell. In fact, the Workplace Shell itself is built on SOM: Workplace Shell objects (folders, the shredder, the clock and so on) are SOM objects. This means that the Workplace Shell is object-oriented, not only in its user interface, but also in the way it is built. It is built on an object foundation - quite literally, built with objects.

The fact that the Workplace Shell is built on objects, means that it can take advantage of the benefits of object-oriented design: it is extendable. Suitably written programs can add their own user defined objects or evolve them from the base classes provided with the system. The result can be that the distinction between application objects and system objects becomes blurred - the applications blend with the rest of the shell to provide a seamless set of services to the user. This is the starting point for evolution towards a full object-oriented environment, where distinctions between applications and the system cease to exist, and objects can be combined in different ways to accomplish user tasks.

The way in which such objects are built in OS/2 2.0 is via SOM. SOM tools are provided with the OS/2 2.0 developers′ toolkit. SOM is not a programming language, but a system for defining and manipulating object class libraries. It provides a set of APIs and a run-time library to allow object-oriented programs to be written to interface with the Workplace Shell. In fact, SOM has a wider scope even than the Workplace Shell objects you see on the desktop. It is a means of implementing object-oriented constructs, and is designed to be system and language-independent. This is an important feature, because most object-oriented development environments today are language specific (eg you can use either C++, or Smalltalk, but not mix both), and therefore it is difficult to share or subclass objects or classes from other environments.

Note that you do not need to develop SOM objects to get a high degree of integration with the Workplace Shell. Many features such as drag and drop are provided in standard PM APIs.

The significance of SOM, rather, is in providing an object-based layer on today′s generation of operating system. This allows many of the benefits of object-oriented programming to be realised on a platform that also provides compatibility with older applications. As such, SOM provides the architectural foundation today, for OS/2 to move in future towards increasing object-oriented content. The Taligent joint venture (see “Object-oriented environments” on page 113) is another potential source of object technology. As object-oriented development becomes increasingly important throughout the 90s, OS/2 and SOM will evolve to meet the need to use such techniques in developing OS/2 applications.

Moving from a previous GUI

The design of the Workplace Shell has been tested for several years in IBM′s usability laboratories, and the beta test of OS/2 2.0 provided more information on how easy it is to learn and use. In general, people who had little experience of computers found the Workplace Shell easy to pick up and use, once the basic principles had been explained. For DOS users who have not been exposed to GUI before, and for Macintosh users, the transition is not very difficult either. IBM′s tests, and the evidence of the OS/2 2.0 beta test, showed that the main category of users who encountered difficulties, were those who had used a previous PC-based GUI, such as Windows 3.x or OS/2 1.x.

During the beta test, as a direct result of customer feedback, some changes were made to improve the migration for this latter class of user. These included the ability to predefine the look and feel of the shell to be more like that of Windows 3.x or OS/2 1.x, and individual features such as the choice of minimising windows to icons on the desktop, rather than "hiding" the window, which is the default Workplace Shell behaviour.

In fact, though the default look and feel can appear different on first view, most items are very familiar to the Windows or OS/2 1.x user once the connection is made between the old style and the new:

Table 7. Windows 3.x and OS/2 1.3 vs. Workplace Shell

Windows 3.x - OS/2 1.3	**OS/2 2.0 Workplace Shell**
Groups	Folders (offers more function - see "Objects and folders" on page 81)
Desktop Manager/Program Manager	None. Instead, a random "messy desk" assortment of objects (files, programs or devices). This can be made as "messy" or as "tidy" as you wish.
Icons	Objects. This is an important distinction. An icon now represents the object (file, program or device) and not a running program. The main consequence of this is in the way minimised windows are treated (usually they are hidden, but you can choose to represent them as icons on the desktop - this can be set on a per-object basis)
Print Manager	Now separate print objects for each printer (see "Extra facilities" on page 86)
File Manager	Now a "Drives" object for each drive (see "Extra facilities" on page 86)
Control Panel	Separate objects for colour, fonts, mouse etc.
Task List	Window List (also accessible via Ctrl-Esc)
Menus	Pop-up menus, see "Pop-Up menus" on page 83
Minimise button	This "hides" the window rather than places an icon on the desktop, but the behaviour can be customised to iconise the window if required.
Adding Programs	Use Templates, see "Templates" on page 82
Mouse usage	Two button mouse usage. In general, button 1 selects and button 2 drags. Use the Tutorial to learn more about the mouse. Mouse button behaviour can be changed to suit the user

But in nearly every case, not only does the Workplace Shell equivalent provide the Windows function, but it also offers more flexibility. For example, folders offer far more function than the simple group windows in OS/2 1.x or Windows 3.x (see "Objects and folders" on page 81 for more information):

- allows data files, programs and devices to be grouped together
- allows long descriptive names
- colours, fonts and icons can be easily changed on a per-folder basis
- the workarea property can be set to enable groups of applications to be opened and closed together

Furthermore, much of the new function provides a powerful incentive to learn because of the many benefits, for example:

Templates A consistent way of creating new objects, whether programs, data files or folders. In Windows 3.x these would take a different procedure for each item. (See “Templates” on page 82)

Hiding windows Although this behaviour takes some getting used to for the Windows or OS/2 1.x user, the experience of the beta test program suggests that most people never wish to go back to the old way, in which the desktop is cluttered with minimised icons. Hiding reduces clutter from the screen, and the Window list (see below) is a much more effective and consistent way of retrieving open windows. Tests show that minimising a running program to an icon is confusing for users who have not used a computer before, as there are often two copies of the icon, one to represent the program reference, and the other the running program, and the user does not know how to differentiate between them.

Window List This is a more powerful tool than the Task List, as groups of applications can be resurfaced, closed or tiled in one step (see “Multi-tasking and the user interface” on page 29)

Mouse Buttons Since Mouse Button 2 is unused in Windows, there is no comparison, except that there is a happy coincidence of use between IBM and Borland in their use of the Mouse Button 2 to produce a pop-up menu on a given object (see “Pop-Up menus” on page 83)

Flexible Desktop Objects and windows can be placed where the user wishes, and the action of the desktop is modeless - there is no need to close certain windows before others can be opened. There are therefore no artificial restrictions in the user′s movement around the desktop.

But, even if the new function does not attract, it is not necessary to learn all of the new ways at once. OS/2 provides all the old function too: there is an object for the DOS or OS/2 command prompt, for those users addicted to the C:\> prompt; and Windows users can select a WIN-OS/2 Full Screen session if the familiarity of the Windows environment is required (see “Full Screen or Seamless” on page 61).

But the most important difference from Windows is the Workplace Shell′s flexibility: you can put icons where you choose, not where the Program Manager or File Manager allow you; you can change the look or behaviour of almost any aspect of the system if the default does not suit you; and you can use drag and drop across most of the system, not just the File Manager.

Benefits of the Workplace Shell

But all of the function and power of the Workplace Shell would be useless unless it provided tangible benefits over other GUIs. This section lists some of the advantages.

Easy to learn

No system is totally intuitive, but the Workplace Shell offers a number of aids to quick and effective learning. The tutorial gives most of the basic elements, and can be revisited at any time. There are productivity applications and games to get productive quickly, and have fun while learning. And the online help system is comprehensive as a reference, or as immediate help when required. The “Start Here” icon gives a focal point if you ever get completely lost!

Flexible

Almost anything can be changed if the default does not suit. You can change colours, fonts and background images, not just for the system as a whole, but for each folder. The behaviour of mouse and keyboard, of how windows work, and most other system defaults are all configurable. And the workarea principle can make each project have a different combination of icons and folders, to suit the job you are doing.

Personal

The Workplace Shell returns to the principle that the PC is a ***Personal*** computer. Its′ motto could be "have it your way." A secretary′s desktop can look different from a manager′s desktop, and each different from an engineer′s. You can have as many or as few icons as you wish. You can change the look and feel to behave more like OS/2 1.x or Windows 1.x.

Simple

Though the Workplace Shell is a powerful environment, it does not overwhelm the user. You can start by using what you know and are familiar with (including the C:\> prompt if you wish!), and move on to learn new function as you need it. Much of the new function and tasks can be learned by experimentation, as the consistency of the shell, and the visual feedback (eg "no-entry" signs where icons cannot be dropped) allows users to teach themselves.

Compatible

The Workplace Shell offers the old ways of working like the DOS prompt and the Windows Full Screen. It takes nothing away. But it also provides a standard way of running applications without compromising any compatibility. It does not enforce a way of working with older applications that is foreign to the user. You can either close applications in the consistent OS/2 way by double clicking on the system icon, or use the application commands (such as / Quit for Lotus 1-2-3).

Consistent

The Workplace Shell provides the same way of working whether your resources are local or on the LAN. It allows you to use techniques like drag and drop consistently across the system, where you would need three different commands in DOS, and would be restricted to doing things from the File Manager only in Windows.

Information-oriented

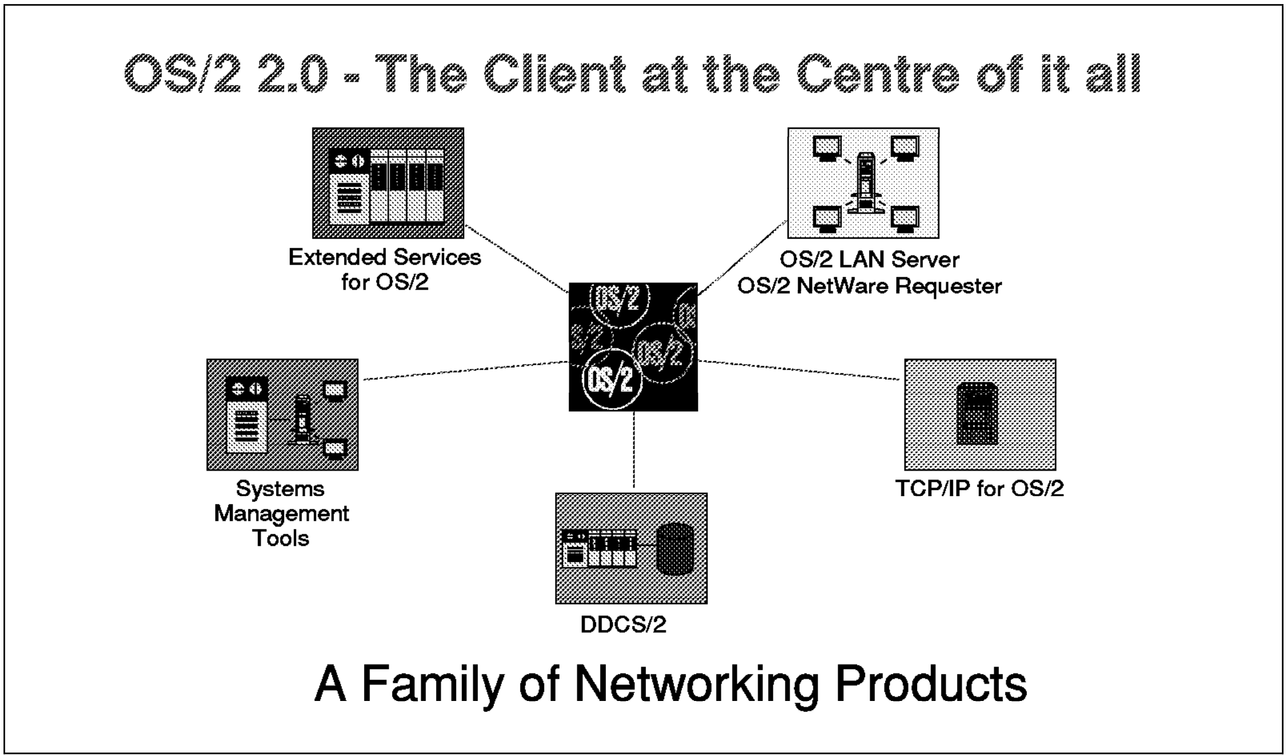
The Workplace Shell allows users to work in a more natural way, by focusing on the information they need to work with. There is less need to worry about the computer′s housekeeping, like files and directories. All in all, you work the way ***you*** want to work, not the way the computer forces you.

Integrating

Applications are loaded in the same way, data objects are treated in the same way, whether they come from a DOS, Windows or OS/2 program. The Workplace Shell integrates local and remote resources. The whole purpose is to present as seamless an interface as possible to the variety of tools, application and data that users need to do their job. in this respect, it is a ***unifying*** interface, and reinforces the key aim of OS/2 2.0 as the integrating platform.

OS/2 in a connected environment

OS/2 is the critical component in IBM′s vision of the complete, managed, client-server system. It is the key element which allows the PC to become the focal point of information processing. Instead of in the past, where key data processing was performed at a remote host, and information provided to users from the central system, the OS/2 vision moves the PC to the centre, and places the user in control of the information. This vision lies at the heart not only of IBM′s user interface design (see "Workplace Shell" on page 77), but also of its strategy to make OS/2 the base for an integrated family of networking extensions, that allows OS/2 to be a completely network-aware system. This makes OS/2 the ***Integrating Platform***, not only for local productivity applications, but for the other components of the enterprise system. It is no longer realistic to view the PC in isolation from the rest of the corporate network.



OS/2 2.0 - The Client at the Centre of it all

A Family of Networking Products

Figure 28. The OS/2 family of networking extensions

The systems extensions to OS/2 complete the picture of the full function OS/2, designed to address the needs of the PC platform of the 90s, as discussed in the section "The changing PC environment" on page 1 . They add to the wide application choice, ease of use, investment protection and reliability of the base system, by providing extensive connectivity, management tools and exploitation of the existing corporate network. This section expands on the strengths of OS/2 2.0 as an easy-to-use productivity platform, and shows how, as a base platform, and with its extensions, OS/2 is enabled to act as the "super client" of today, and the future.

OS/2 for client-server

OS/2 is not only today′s server platform of choice (acknowledged by both IBM and Microsoft), but the best available client. These are some of the reasons why:

- ***Consistent Platform for both Client AND Server***: It is clear that if the same platform can be used for both the client and the server, then the benefits of consistency will lead to a much more manageable platform, with only one operating system to support. OS/2 is the only operating system available that is consistent across client and server.
- ***Multiple Concurrent Network Protocols***: OS/2′s great strength in communications is acknowledged across the industry. In particular, OS/2′s multi-tasking design allows it to handle multiple communications protocols (eg NETBIOS, 3270, IPX, TCP/IP) with ease. This ease of integration is in contrast with the problems of attempting such concurrency under DOS or Windows, such as lack of memory, poor multi-tasking performance and the instability of DOS-based multi-tasking. In a world where heterogeneous or mixed-vendor connectivity is becoming the norm (a recent report estimated that 70% of LANs run more than one protocol), OS/2 is today′s only reliable client choice. At the Networld trade show in early 1992, IBM demonstrated a single OS/2 client machine with one Token Ring card, connected to a “wall of servers,” which included three LAN servers, one running NetWare, another Banyan Vines, and another OS/2 LAN Server; a RISC System/6000 running AIX, an AS/400 and an ES/9000. One client was connected simultaneously to all these shared resources - it was, of course, an OS/2 2.0 client.
- ***Enabled for LAN-based install***: OS/2 can be installed from a server to multiple clients, allowing faster and more controlled installation, with the added benefits of greater automation. Future tools will be available from IBM to ease the system administrator′s job still further (see “Graphical installation” on page 69 for more information on installation).
- ***LAN-independent shell***: OS/2′s Workplace Shell provides a graphical view of resources, whether on the local machine or on the LAN. Remote drives can be set on the LAN from products including NetWare, OS/2 LAN Server, Banyan VINES, or TCP/IP for OS/2, and also on host systems using Virtual Disk facilities in products like PC Support/400 and Workstation LAN File Services/VM. The Workplace Shell gives seamless access to different server environments, making multiple connectivity easier to implement, and provides consistent access to network files and printers (see “LAN-independent shell” on page 90)
- ***Easier to Manage***: OS/2 provides the ability to allow administrative tasks (such as collection of local configuration data and performance measurement) to be run on the client while preventing excessive impact on local performance and usability. Such tasks are becoming increasingly important in a highly distributed PC client-server environment, and particularly where PCs are being rolled out into mission-critical usage in remote locations where on-site support cannot be provided. Remote diagnosis and support is therefore critical in such environments. But background monitoring and data collection cannot be achieved easily on a single-tasking platform like DOS. And the limitations of environments like Windows, that attempt to graft a multi-tasking layer on top of single-tasking DOS, are revealed when such systems management functions are attempted. Even if such tasks could be run in the background on Windows (not always the case), the lack of true multi-tasking would cause the background “probe” to be intrusive to the user′s foreground activity. Only a pre-emptive multi-tasking environment like OS/2 can offer these benefits.
- ***Supports both productivity and line-of-business applications***: Today′s client-server environments are moving beyond the simple file and print sharing of the first LANs, towards applications like Lotus Notes, and SQL database applications, that can deliver competitive edge through enabling workgroup communication. In this environment, it is important to deliver not only the support for in-house “line-of-business” applications, but leverage the investment in client productivity applications. OS/2 2.0 is the only platform that delivers industrial strength reliability ***AND*** wide application compatibility.

- ***Reliability***: This is perhaps the most critical issue of all. Many of the reported problems with Windows 3.0 occurred in networking environments. It is clear that although Windows 3.1 may have improved reliability in some areas, it has not changed any of the architectural deficiencies that cause its limitations in networking; most of the problems stem from the fact that Windows continues to run on DOS (see "Reliability and protection" on page 39 and "Reliability" on page 54). The issue can be summed up thus: what use is it having a fault-tolerant server if you cannot rely on your client?

Forrester Research′s May 1992 report on the growth of the "super client" (see "The changing PC environment" on page 1) identifies OS/2 2.0 as being a leading candidate to satisfy the demand for a highly protected, network-aware, true multi-tasking client platform, for use in "line-of-business" applications. According to Forrester, the "super client" role is beyond the scope of either Windows 3.1 or Windows/NT, and predicts for the latter "product delays...and bloated hardware requirements." That independent analysts should make such statements is not surprising, given the difference in protection between OS/2 2.0 and Windows 3.1, and the difference in hardware requirements and availability of OS/2 2.0 and Windows/NT. OS/2 2.0 is the platform that meets the exacting requirements of the modern client platform (see "The changing PC environment" on page 1) in an acceptable configuration, and is available today, not promised for the future.

The OS/2 family of networking extensions

OS/2 2.0 is part of a family of products and systems extensions, which are designed to work together. In other environments, customers have to buy third party software (if it is available) and hope it will all work together; with IBM these extensions are tested together and integrated. Here are some of the systems extensions from IBM which complement the OS/2 base system:

Extended Services for OS/2

Extended Services for OS/2 is a separate product which provides communications and database functions. It includes Communications Manager, which offers a wide range of connectivity and protocols (all of which can be active at the same time); and also provides Database Manager, a powerful client-server SQL relational database, part of the SAA family of relational databases that includes DB2 and SQL/DS. Extended Services for OS/2 release 1.0 works with both the OS/2 Version 1.3.1 16-bit base and OS/2 Version 2.0 32-bit base. This will be an advantage in mixed 286 and 386 environments. Extended Services is supported on a selected range of IBM-compatible PCs, not just PS/2s.

In keeping with the intention to offer modular options to customers, Extended Services comes in two forms, which differ only in their database function: Extended Services for OS/2 delivers Communications Manager and Database Manager in a single package, providing an "all-in-one" complete connectivity solution; Extended Services with Database Server for OS/2 adds the ability to create databases on a server, and offers cost-effective client functions for DOS, Windows and OS/2 clients.

Extended Services is a key component for OS/2′s participation in the SAA standards, particularly for communications protocols (APPC, CPI-C, and APPN) and relational database (SQL, DRDA). It is one of the key building blocks for the SAA co-operative processing applications of the future, both by third party applications vendors and by customers themselves.

DDCS/2

SAA Distributed Database Connection Services/2 (DDCS/2) is a complement to Extended Services. It offers host database connectivity to an OS/2 client and, working with Database Manager, allows DOS, Windows and OS/2 clients to access host databases conforming to the Distributed Relational Database Architecture (DRDA), which includes not only IBM′s DB2, SQL/DS and OS/400, but potentially third party database products too. DDCS/2 widens the scope of the OS/2 client, and is part of the wider SAA distributed database direction.

OS/2 LAN Server

OS/2 LAN Server version 2.0 is a powerful platform for providing LAN services to DOS, Windows and OS/2 clients.

Entry and Advanced levels are available. Entry provides an economical base system for both the 16-bit and 32-bit bases while Advanced adds features like a high performance 386 file system (HPFS386) and additional error recovery, on the 16-bit base. OS/2 LAN Server 3.0, which at the time of writing had entered beta test, will offer, among other new features, the ability to run Advanced level function like HPFS386 on the OS/2 32-bit base. Other new features in LAN Server 3.0 include:

- high level of NetWare and OS/2 LAN Requester client co-existence, on the client desktop, and when integrating NetWare resources into an OS/2 LAN Server domain as externally defined resources
- peer Services, which allows a OS/2 requester to share resources with one peer
- redirected Install, delivering unattended install within the CID process
- 802.2 Virtual Device Driver (VDD), enables DOS 802.2 applications to share an adapter with other DOS and OS/2 applications (this function is also included in NTS/2 - see “IBM Network Transport Services/2” on page 102)
- OS/2 TCP/IP coexistence via support of NETBIOS over TCP/IP, allowing support of a greater range of protocols
- improved disk fault tolerance, mirroring without the need for rebooting the server, and mirroring of the boot drive
- support for Apple Macintosh clients via an add-on product

Like Extended Services, OS/2 LAN Server is supported on a range of selected IBM-compatible equipment as well as PS/2s.

OS/2 LAN Server is a robust, scalable solution, from the small LAN, to the large and complex. It makes life easy for the user, the LAN administrator, and the systems manager:

- *for the user* - a single view of all resources available, and automatic allocation of resources. Combined with the Workplace Shell, this becomes even easier.
- *for the administrator* - the Domain feature makes management of large LANs much easier, and allows location-independent resource naming, making changes easier to implement.
- *for the systems manager* - performance and management tools, including integration with the SAA host-based NetView.

OS/2 LAN Server provides excellent performance. In tests run by LANQuest Labs, an independent benchmarking company (report dated June 1992), OS/2 LAN Server Version 2.0 had the best overall performance compared to Novell NetWare 3.11 and Microsoft LAN Manager.

In the future, IBM plans to enhance OS/2 LAN Server towards full distributed function for the LAN environment, including common distributed services, common developer infrastructure, and open industry standards.

OS/2 LAN Server and Microsoft LAN Manager

In 1989, IBM and Microsoft made a commitment to work towards greater commonality between Microsoft LAN Manager and OS/2 LAN Server. The original scope of work has now been completed, and commonality has been achieved at the API and functional level. This means that applications can be written to run on both platforms. OS/2 LAN Server clients and Microsoft′s LAN Manager clients can coexist and interoperate on the same LAN. Both clients can logon and access resources at either or both servers. OS/2 LAN Server and LAN Manager utilise a common underlying security system as well, which includes common user domains, access control and server local security. OS/2 LAN Server′s support for selected non-IBM equipment means that customers can now buy LAN Server on both IBM and OEM machines, and also achieve consistency that way.

Microsoft has not, at the time of writing, committed support in Microsoft LAN Manager for OS/2 2.0, either as a server or as a client, although they have access to the OS/2 2.0 code. Nevertheless, customers using LAN Manager 2.0 may obtain OS/2 2.0 client access to the server, using IBM LAN Enabler version 2.0 (see “LAN Enabler” on page 102).

Although the current release of OS/2 LAN Server contains code licenced from Microsoft, future OS/2 LAN Server plans (see above) have no dependency on Microsoft, whose declared future plans for LAN Manager place little emphasis on OS/2.

NetWare from IBM

IBM, as an open vendor, provides its customers with the two major options for server software: OS/2 LAN Server and NetWare from IBM. OS/2 LAN Server is an excellent platform in an environment requiring IBM host connectivity today, and complements the use of Extended Services, but many customers have NetWare-based LANs, for application, historical and functional reasons. They therefore require co-existence and interoperability between the two standards. NetWare expands IBM′s ability to offer better solutions in a mixed networking environment, where support of diverse clients (Unix, Macintosh) are required in addition to DOS, Windows and OS/2.

OS/2 2.0 is an excellent client for NetWare. OS/2 1.3 plus the OS/2 NetWare Requester offered access to services on the various NetWare platforms. For OS/2 2.0, the NetWare Workstation Kit for OS/2 Version 2.0 includes the requester code and utilities required for a fully functional NetWare client. A separate product, NetWare Services for OS/2, adds to the requester code, a MAP utility for network management.

IBM offers co-existence between LAN Server and NetWare. With the OS/2 LAN Requester and the NetWare OS/2 requester installed, the same OS/2 workstation can log on to both types of server. This allows customers to use both products according to immediate requirements and installed base. Greater interoperability is planned. One key element in achieving this is the commitment by Novell to move NetWare to the 32-bit OS/2 base. This will combine the strengths of NetWare with the power of 32-bit OS/2 as a server platform. Novell′s plans amount to a key endorsement of OS/2 as both a server and client platform. IBM will continue to offer interoperability with NetWare systems as we incorporate more distributed services into the OS/2 LAN Server.

LAN Enabler

IBM LAN Enabler version 2.0 offers the OS/2 requester, LAN Support Program and DOS LAN Requester (DLR), identical in function to that provided with OS/2 LAN Server 2.0, in a separate product. The package includes requesters for DOS, Windows, OS/2 1.3 and OS/2 2.0, as well as protocol support, NDIS-compliant network adapter drivers, LAN API support, and a VDD for NETBIOS applications.

This product will allow 286 and 386 PCs to connect to servers including OS/2 LAN Server 2.0 and Microsoft LAN Manager 2.0, and other compatible servers, without having to buy a separate copy of LAN Server, or Extended Services for OS/2. It also allows OS/2 2.0 client access to NETBIOS, 802.2 and NDIS applications without buying LAN Server or Extended Services.

This enhances the function of OS/2 2.0 as a client to different networks by providing the necessary function in a package separate from the server, at an economical price.

IBM Network Transport Services/2

IBM Network Transport Services/2 (NTS/2) provides networking support on an OS/2 2.0 base, without requiring OS/2 LAN Server or Extended Services for OS/2. It provides the LAN adapter and protocol support (LAPS) to support networking applications on an OS/2 2.0 machine, and to enable automated installation of OS/2 and other CID[22] -enabled software across a LAN.

NTS/2 is a combination of:

1. Network Driver Interface Specification (NDIS) compliant transport protocol and network adapter software
2. OS/2 2.0 support for DOS programs requiring NETBIOS and IEEE 802.2 APIs, by providing a VDD/PDD combination for NETBIOS and 802.2. This allows DOS applications using a Token Ring adapter, to share that adapter with other DOS and OS/2 applications running on the same machine. For example, this allows DOS and Windows 3270 emulation programs using Token Ring to share the adapter with OS/2 LAN Requester.
3. Configuration Installation Distribution (CID) enabling software, including a LAN CID utility (LCU) to manage the automated installation process.

Note that the LAPS function was also released earlier in the LAN Enabler/2 v2.0 product. NTS/2 adds to LAN Enabler/2, the CID support to allow automated installation of OS/2 in a LAN environment. In this respect, NTS/2 is more akin to the DOS-based LAN Support Program, with the addition of CID support (ie it provides the network transports and APIs to enable LAN support), whereas LAN Enabler/2 can be thought of as related to the OS/2 LAN Requester (since it provides command line and menu interfaces to manage LAN-based resources).

Open systems connectivity

Although OS/2 is the SAA client platform, it is also vital that it embraces the open systems world. IBM′s strategy is to enable OS/2 and AIX in particular, to interoperate as much as possible. TCP/IP for OS/2 is one product that provides significant networking co-existence with UNIX-based systems, providing support for many open standards such as TCP/IP, NFS, TELNET and X-Windows. This allows OS/2 to extend its client capability into the open systems arena. IBM′s strategy is to link the OS/2 and open systems worlds closer together through common systems extensions, connectivity and management tools (see “AIX interoperability” on page 116)

[22] CID stands for Configuration, Installation, Distribution - see “Configuration, Installation, Distribution (CID)” on page 104

The combination of the OS/2 2.0 base, and TCP/IP for OS/2, means that the OS/2 2.0 user can today view, *on the same screen*, DOS, Windows, and OS/2 applications as well as Unix applications through X terminal or TELNET. This widens still further the range of application support OS/2 2.0 can provide.

Systems management

Connecting PCs together is only one part of the process of integrating the PC into the enterprise system. The more PCs are installed in a company, the harder it is to control their installation, maintenance, problem solving and performance tuning. IBM has proven strength in handling complex, distributed networks, and therefore understands the importance of providing tools to help manage these tasks. To date, systems management for OS/2 has been provided by two means:

- by functions integrated directly into OS/2 and extensions like Extended Services (such as the First Failure Support Technology - FFST/2)
- by separate standalone products which address a specific and distinct aspect of the overall task of managing OS/2 systems

In the latter category, IBM has already produced a variety of products, based on OS/2, to assist various aspects of systems management. The table below lists a few of them:

Table 8. OS/2 systems management tools

SPM/2	System Performance Monitor/2 (SPM/2) tracks key aspects of a system′s use: CPU, disk etc, to identify performance problems. This can be used in both server and client environments.
NetView Distribution Manager/2	an installation management tool, working in conjunction with the host-based NetView, that includes a LAN Download Utility to install or upgrade software via a LAN server to LAN-connected PCs.
SAA Delivery Manager	allows host-connected OS/2 machines to receive software updates from the host
DCAF	Distributed Console Access Facility (DCAF) allows remote diagnosis and management, by receiving screens from and controlling the keyboard of a remote OS/2 machine.

IBM is continuing to expand on this range of management tools, and to integrate them more closely. IBM′s long term goal is the provision of an architected management platform which adheres to indistry standards from organisations such as ISO, X/Open and OSF.

IBM′s strategy for systems management is based on the belief that in establishing the true cost of a PC system, the initial hardware and software cost is only a small proportion. The extra costs, what many independent analysys call “cost of ownership,” relate to issues like integration, support and systems management; or, as it was stated in “The changing PC environment” on page 1, the costs of maintaining the systems should not exceed the benefits it provides. With the growth in the size and complexity of LANs and client/server systems, planning and managing these systems has become increasingly difficult, time-consuming and expensive. In a report in April 1992, Gartner Group said that “Costs surrounding product updates represent the largest component of PC software life-cycle costs.” Customers are therefore investing a growing amount of resources into distributed systems management (DSM) and are demanding that they be simplified and automated.

On October 6th, 1992, IBM responded to customer demands for solutions to DSM problems with 2 major announcements:

- the IBM LAN NetView family of DSM products
- a generalized solution for configuration, installation, and distribution (CID) of OS/2, its subsystems, and application software.

IBM LAN NetView family

The IBM LAN NetView family of products is an OS/2-based platform for performing DSM functions in enterprise-wide, multivendor local area networks (LANs). The LAN NetView platform provides the ability to monitor and control information processing resources, including hardware and software, throughout an enterprise. Information is collected from monitored resources (managed systems) and sent to other resources (managing systems). The offerings allow the management of OS/2 2.0, DOS 5.0, and DOS 5.0 with Windows 3.1 clients.

The new platform allows the LAN administrator at a workstation to access different systems management applications using the same screen. It provides enhanced automation and greater integration among products to streamline such management tasks as detecting hardware or software failures on the network - which often require the combined efforts of users and trained personnel at both remote and central locations.

The set of LAN NetView distributed system management applications being developed by IBM include:

- LAN NetView Manage, the platform product that provides system services for LAN NetView management applications.
- LAN NetView Start, LAN NetView Monitor and LAN NetView Fix, three management applications designed to run on that platform. They provide, respectively, configuration, systems performance and fault-management services.
- LAN NetView Tie, which provides a LAN-to-NetView link for customers who prefer to divide management of system resources between LAN- and host-based products, or who prefer to centralize all management on the host processor.

IBM is encouraging vendors to provide agents for their system resources to allow LAN NetView products to manage them, and is encouraging vendors to provide additional system management applications on this platform. Among the vendors who have declared their support are Novell, Microcom, Computer Associates and Ungermann-Bass. Customers may also choose to implement their own LAN NetView compliant distributed system management applications.

To encourage the development of third-party systems management applications, LAN NetView provides a set of common application programming interfaces (APIs) that have been accepted by key industry organizations that further standards, such as the Open Software Foundation (OSF) and X/Open, and accepted by IBM′s SystemView. One of these interfaces, named the X/Open Management Protocol (XMP), allows developers to create applications without having to know the underlying management and transport protocols, such as CMIP and SNMP. SystemView has adopted XMP as its Common Management Interface Protocol (CMIP), and the LAN NetView products are among the first management products in the industry to incorporate elements of the OSF′s Distributed Management Environment (DME) technology.

Configuration, Installation, Distribution (CID)

IBM also recognises the need to provide a generalised solution to the problem of automated installation and configuration of multiple PCs.

On October 6, IBM announced its CID strategy. CID is not a single product, but a collection of processes with supporting products. It automates the processes of configuration, installation and distribution in the DSM environment and dramatically improves productivity for system administrators. This allows both standalone LAN and host-attached LAN customers to realize dramatic resource savings by using the CID processes.

Though there are several aspects to CID enablement, its primary objective is to provide unattended, remote installation of OS/2, its subsystems, and application software on LAN attached workstations.

Customers with both large and small networks will be able to remotely install and manage software such as OS/2 or OS/2 applications from a single location, without requiring the involvement of the desktop workstation user. The CID process can apply to installing not only OS/2 system code, but also supported application code. Each system requires a physical LAN connection, but is not reliant upon a file server like OS/2 LAN Server or Novell NetWare for the distribution mechanism.

The installation process is performed by an individual who initiates the installation process at their workstation and performs other tasks such as removing/replacing diskettes when prompted and rebooting the system when prompted. This individual would not be required to know the responses to detailed product installation questions. The answers to the product installation questions are kept in response files which contain specific responses to each installation question. The response files are unique to each product. Each product provides either a utility or a model which the system administrator can use to generate the response files. A procedure file which may be tailored is set up by a system administrator and stored on the code server. This procedure file contains the commands which initiate product installations for other CID enabled products such as IBM LAN Server 3.0.

CID provides the ability to individually tailor PCs or groups of PCs. Previous solutions allowed for cloning, where software is distributed by duplicating the files across the network. These duplicated files did not always fit individual workstation environments and sometimes required extensive changes from the local user. With CID technology, individualized support is now enabled for OS/2 products and applications, so that users can automatically receive new software already configured for their computers. DOS and Windows users will continue to be supported in the short term by cloning, with integrated support for DOS clients to be provided by mid-1993.

Among the products that are being delivered to implement the CID process are:

- LAN NetView Start, which allows administrators to manage and plan the configuration and installation of OS/2 software across the network.
- Network Transport Services (NTS/2), which provides NetBIOS connectivity and helps execute instructions generated by Start about installing software on the workstation.
- NetView Distribution Manager/2 (NetView DM/2), which will now provide expanded software distribution and installation capabilities, including change management and recovery, across standalone or interconnected LANs from a single site on the LAN.
- Enhancements to NetView Distribution Manager (NetView DM) for an MVS host, which, in conjunction with Netview DM/2, will now allow centrally controlled distribution and installation, as needed, of system software, user and vendor software, software changes and data files.
- Software Profile Management Facility, which works with NetView DM to store descriptions, or ″profiles,″ of groups of users with similar characteristics, simplifying software installation and updates for a large number of workstations.

All system and application software to be installed using this method must be CID enabled. OS/2 Version 2.0, OS/2 LAN Server 3.0, NTS/2 v1.0 and OS/2 Extended Services Database Manager are all CID enabled. Enabling consists of allowing installation to be redirected to a network drive, and the ability to provide user input through a response file.

In addition to CID enabling our own products, IBM is encouraging vendors to provide CID enabled applications. IBM also announced agreements with independent software vendors (ISVs) who intend to enable selected products to take advantage of CID. To date, a total of 164 vendors have committed their products to become CID-enabling, including Computer Associates, Describe, Lotus, Micrografx, Novell, Software Publishing Corp., Symantec and WordPerfect.

Commitment to open standards

One of the most important aspects of IBM′s systems management strategy, including OS/2 systems management, is IBM′s intention to esnure interoperability with a broad range of other vendors′ systems management products. That is why the IBM systems management products aim to support industry standards such as CMIP and SNMP, and are based on the OSI Management Framework standards implemented by Hewlett Packard′s Open View Network Management Server Product version 3.1. Thus the IBM tools will be able to take advantage of a programming interface consistent with the Open Software Foundation Distributed Management Environment (OSF/DME), as well as being part of SystemView, IBM′s strategy for enterprise-wide systems management.

OS/2 is the base platform for all of these management extensions, because it is only OS/2 that is extendable in this way, and can provide the support for these management functions (see "OS/2 for client-server" on page 98). It is further illustration that OS/2 is the true client of choice, as well as the server.

Migration from existing connectivity products

IBM recognises that customers who wish to move towards the "super client" platform, still have signficant investments in older connectivity products. OS/2 2.0, as the integrating platform, provides a wide degree of support for DOS-based connectivity products, as well as supporting other existing networking products.

Networking on OS/2

Not only does OS/2 provide support for the two most popular network server products, NetWare and OS/2 LAN Server, but it is also a platform that runs a large range of the other major LAN software products, either as client, or server, or both. Many of these products had support for OS/2 1.3, but new versions are being produced for OS/2 2.0 from vendors such as Banyan (who are developing a requester for their VINES network system) and Digital (who are planning support for their Pathworks product). This makes OS/2 not only the client of choice in these networks, but also, for many of them, the base server platform.

DOS requester products for some of these products can be supported, running in a VDM, though this often (as in the case of IBM DOS LAN Requester for OS/2 LAN Server) requires booting the real version of DOS in a VDM (see "Virtual Machine Boot (VMB)" on page 45). Although DOS network drivers can often be run in a VDM, they can only provide services to that individual VDM, since there is usually no Virtual Device Driver for the protocol stack or the Media Access Control driver (eg 802.2). Obviously, this restriction is usual for OS/2 2.0 (see "DOS device drivers" on page 40), and in most cases causes no loss of function where a device is dedicated to a single application (as it usually is for fax cards and scanners). However, it tends to defeat the purpose of networks to have the resources available to only one application, and therefore OS/2-based network requesters and drivers are recommended whenever they are available (many vendors already have or are working on OS/2 2.0 requester code for their server product; contact your vendor for details).

DOS communications applications under OS/2

Because of OS/2′s extensive compatibility with existing DOS and Windows applications, it can even run DOS-based communications applications. This includes not only popular asynchronous communications products such as ProComm Plus, but also IBM′s own DOS 3270 and 5250 products, Personal Communications/3270 and DOS PC Support/400. The communications aspects are straightforward, as OS/2 is designed to handle multiple concurrent protocols and even background communications with more reliability than is possible under DOS extender environments such as

DOS/Windows 3.x (see “Virtual Device Drivers (VDDs)” on page 21 for more discussion of how it does this).

However, both Personal Communications/3270 and DOS PC Support/400 require some customisation before they can be made to run satisfactorily under OS/2 2.0: the first requires tailoring the DOS Settings, the second requires booting the “real” DOS in a VDM to support special functions like shared folders (which uses a block device driver) - see Appendices A and B in *OS/2 Technical Compendium - Volume 2: DOS and Windows Environment* (GG24-3731-00) for more details.

There are, even with these settings, some restrictions when running these applications:

- File transfer must be executed from the same VDM as the emulator runs in (you may need to enable session switching to the VDM DOS prompt via the DOS Setting KBD_CTRL_BYPASS)
- Limitations in PC Support function (only Basic DOS, not Extended DOS support; other DOS, Windows or OS/2 applications cannot access PC Support/400 facilities)
- Sharing of Token-Ring Adapter (if using Token Ring rather than DFT or Twinax) with other applications is not possible without an 802.2 VDD. This restriction disappears if the 802.2 VDD from NTS/2 is installed (see “DOS device drivers” on page 40 and “IBM Network Transport Services/2” on page 102).

Because of these restrictions, it is recommended to use OS/2 versions of these emulators and communications functions, which appear in Extended Services for OS/2, because of the higher function, easier installation and access of resources across DOS, Windows and OS/2 applications. Nevertheless, the support of most functions for the DOS applications under OS/2 enables, for many users, a more than adequate migration. This is in contrast to some of the press reports concering Windows/NT, which warn that compatibility with existing DOS-based connectivity programs and device drivers is likely to be extremely limited, forcing users to consider buying new connectivity software if it is available by the time Windows/NT ships.[23] This is, of course, one of the many compatibility issues that have to be encountered when a new platform is introduced.

[23] See *PC Week* July 27, 1992 - page 1. The lack of support for DOS device drivers is confirmed in the Microsoft document ″Microsoft Windows NT Operating System″.

Futures

It is of course important, as well as examining the strengths of OS/2 2.0, the current release, to look at the future of OS/2 and some of the areas in which it will develop.

OS/2 will continue to evolve to satisfy new requirements and bring forward new technologies. With OS/2 2.0 we have designed the system not just for 1992, but as a platform for many developments throughout the 90s.

The OS/2 32-bit architecture (OS/2 2.x) represents the base for the immediate future. OS/2 2.0 has the functions that customers need today, and has been designed to carry through to the next major steps, into the world of object-oriented technology and distributed systems. This section examines some of the areas in which OS/2 will be enhanced to meet the needs not only of today′s “super client” (see “The changing PC environment” on page 1 and “OS/2 for client-server” on page 98), but of the long term vision of the “window on the world.” The OS/2 direction is towards a single client system, encompassing existing investments and harnessing new technologies like object and multimedia, and a scalable, robust server platform exploiting the increasing power of the hardware, and distributed networking support based on open standards.

OS/2 1992 developments

By the middle of 1992, IBM had delivered:

- OS/2 2.0 base system
- Extended Services for OS/2
- OS/2 LAN Server 2.0

By the end of 1992 and into early 1993, IBM intends to consolidate on the early success of OS/2 2.0, by delivering updates to address the following issues:

- improved device driver support
- enhanced “Seamless Windows” capability
- performance and memory enhancements
- enhanced WIN-OS/2 support
- extensions for multimedia and pen

The current hybrid 16-/32-bit PM graphics engine will be replaced by more 32-bit technology. This will allow at the same time 32-bit screen device drivers to be delivered for VGA, SVGA and XGA. It is intended that these drivers will not only improve screen performance, but also deliver high resolution “Seamless Windows” capability for resolutions above VGA (see “High resolution support” on page 64). IBM plans to update WIN-OS/2 function to include Windows 3.1 features. Work is continuing to reduce the memory requirements of the system, and to improve performance in low-end configurations. Such work will produce performance benefits in all configurations.

Further multimedia extensions (MMPM/2) releases will ship in 1993, adding video and image capabilities to the audio functions shipped in the first release, which became available in June 1992. OS/2 Pen extensions and a developers′ toolkit are planned for late 1992 or early 1993.

During the same period, IBM will produce device adaptation kits to ensure that more drivers for disks, displays and printers will appear from third party vendors. Some of these may be available via IBM and public bulletin boards. If you have a query about a specific device, you are advised to contact the vendor in question first, to understand their development timeframes and proposed distribution methods.

Many of the individual changes described above will be made available in a service pack before the end of 1992. Some selective fixes are already being made available through bulletin boards and via Compuserve. Other function, such as the WIN-OS/2 3.1 support, is currently in beta test.

These are just the enhancements in the base product. Also planned are a series of enhancements to the systems extensions, including a new release of OS/2 LAN Server, enhancements to Communications Manager and the LAN Enabler, as well as further developments in automated configuration, installation and distribution. These will take place between the end of 1992 and early 1993.

32-bit system extensions (communications, database, LAN)

OS/2 2.0 is the base for Extended Services for OS/2, which provides a powerful set of communications and database functions, and for OS/2 LAN Server version 2.0. The first releases of Extended Services for OS/2 and LAN Server (versions 1.0 and 2.0 respectively) on the OS/2 2.0 base are 16-bit only, in order to provide the same system on both OS/2 2.0 and 1.x. This will allow customers to standardise on common systems extensions across mixed 1.x and 2.0 (16- and 32-bit) bases.

However, future releases of both Extended Services for OS/2 and LAN Server will be full 32-bit products, taking advantage of the 32-bit environment to allow even better performance, concurrency and throughput. In addition, Novell will provide a 32-bit version of NetWare on the OS/2 2.0 base, fulfilling the commitment made in the IBM-Novell February 1991 joint announcement.

Using these 32-bit extensions will allows customers to fully exploit 32-bit performance and throughput across the entire system.

Multimedia

As applications become more sophisticated, multimedia technology embedded in mainstream applications as well as in separate applications will become increasingly important.

OS/2 is already an excellent platform for multimedia, as products such as IBM′s Audio Visual Connection (AVC), M-Motion and the ActionMedia II Developer′s Toolkit have shown. But the 32-bit OS/2 environment allows easier development of more sophisticated multimedia applications, and will give the potential to include multimedia features in many other applications, thus making its use more widespread. In particular, the multi-tasking and overlapped I/O capabilities of OS/2 will allow the kind of applications to be developed that are not feasible under environments like Windows, that are based on DOS.

Multimedia Presentation Manager/2 (MMPM/2) is a series of extensions to OS/2 enabling it to handle multiple media (audio, images and video) in a consistent way. Current DOS, Windows and OS/2 multimedia applications (IBM examples include Storyboard Live!, AVC and Linkway) all include their own multimedia device support and use their own unique data formats. MMPM/2 provides a level of standardisation by offering:

- a hardware-independent layer, the Media Control Interface (MCI), which allows new functions, devices and data formats to be added in a modular way
- a common data specification (RIFF) agreed by both IBM and Microsoft, and common between the OS/2 and Windows multimedia extensions

MMPM/2 provides data streaming and synchronisation services to enable easy management of and transfer of data between devices. MMPM/2 also presents a series of "applets" which control and play devices like MIDI devices, digital audio, and Compact Disc. The Multimedia Presentation Manager Toolkit/2 (MMPMTK/2) is also available to help programmers write OS/2 multimedia applications. It includes language bindings for C and MASM, sample programs and documentation. The MCI has both a procedural message interface for C and MASM, and an interactive string interface. The latter allows easy incorporation of multimedia function into a variety of applications (including productivity applications like spreadsheets and word processors).

The multimedia extensions for Windows will run under the WIN-OS/2 environment, but users and developers of multimedia may prefer to use the OS/2 facilities, which provide as wide a range of device support, data compatibility through formats like RIFF, but greater function, performance, and device utilisation. For example, OS/2′s support for sharing of media devices between multimedia applications is superior to that possible under Windows.

Overall, OS/2 provides a better multimedia environment that Windows, because of:

- Better encapsulation of system resources by OS/2. Under Windows 3.x the programmer needs to be aware of directly controlling the system resources.
- Better stability of OS/2, especially in the area of multi-tasking which provides improved coexistence of applications with less coding effort. For example, yield points required under Windows 3.x are not necessary with OS/2.
- Better graphics, such as Bezier functions, on OS/2.
- Better performance, through use of the flat memory model and pre-emptive multi-tasking.
- more capacity for large data objects (image, sound) through the flat memory model

The combination of better multi-tasking and the support for threads leads to better synchronisation of the multiple data streams needed for multimedia. An example is synchronising digital audio with analog video for a firing cannon or a talking head. OS/2 "fast threads" are an enhancement specifically added to improve the performance of threads for multimedia applications. In Windows 3.x, synchronisation is left entirely to the application. Developers have to write it themselves, thus increasing development time and cost. OS/2 controls multiple accesses to resources more effectively than Windows, so that audio and video can run without noticeable interruption while loading other applications. Such seamless integration of multimedia into the normal business desktop is more difficult to achieve in an environment like Windows, that does not provide pre-emptive multi-tasking.

Multimedia applications frequently need to deal with large memory objects for bit maps, audio streams, or even streams of bit maps. These objects are much easier to manipulate in the flat 32-bit memory model provided by OS/2 2.0.

All of these features will make OS/2 2.0 the multimedia platform of choice.

IBM and Apple announced, in July 1991, the two companies′ intention to collaborate on multimedia standards. In October 1991, IBM and Apple announced the formation of an independent, joint venture company, Kaleida, to develop multimedia specifications and technologies. This brings two of the leaders in the multimedia industry together, and increases the possibility of a broad-based industry standard.

Pen-based computing

Pen-based computing is likely to be an important means of expanding the use of computers in the next few years. IBM has explored the use of Go Corporation′s PenPoint operating system for dedicated pen-based portable systems. IBM believes that some new applications may require the use of technology specifically designed to exploit the pen interface. Nevertheless, applications for existing systems may also benefit from pen-based extensions, particularly on desktop machines, where the pen may prove, in some circumstances, to be a better input tool than those currently available. IBM is exploring these options through technology which we have demonstrated publicly, using pen extensions for OS/2. In future, as the technology develops, the tablet and desktop systems will both offer OS/2 and Pen extensions. It shows once again how OS/2 is the base for most of IBM′s innovative new technology in software.

Systems management

Some companies in the desktop marketplace operate only in the PC arena, and their strategies display little understanding of the difficulties of configuring, maintaining and managing large volumes of PCs in a highly connected environment. As companies′ use of PCs grows, and their importance in the corporate network increases, systems management of these distributed PCs becomes of fundamental importance. The business platform of the 90s will need to be a manageable platform, else the costs of keeping the system running will endanger the benefits.

There is already a range of products available for OS/2 to support the systems manager, including NetView Distribution Manager/2 and LAN Automated Distribution/2 to help automated installation, SPM/2 for performance and resource management, and DCAF for remote diagnosis (see “Systems management” on page 103). OS/2 will also become a full participant in IBM′s SystemView strategy and more products will be delivered to support the process of fault reporting, remote diagnosis, asset management, software delivery and maintenance. It is easier to provide this support on a true multi-tasking platform, which can accommodate administrative processes or threads capturing management data, while processing its normal local applications. The limitations of DOS (and therefore of Windows) confine the function that can be provided to such systems.

Presentation Manager futures

PM is the strategic SAA API for GUI development. IBM is fully committed to its continued support and future enhancements. It has not been announced yet how and in what timeframe the following will be delivered in product form, but they represent some of the areas for future development:

32-bit implementation

It is planned that the internals of PM will be migrated to a full 32-bit implementation. (In OS/2 2.0, a full 32-bit API is already provided - future changes simply affect the way PM works internally). This will give improved performance (it is anticipated that this change may improve performance for many applications running on the Workplace Shell desktop). The first stage of this is being provided in the Service Pack available by the end of 1992, with the 32 bit PM Graphics Engine. Later work will move the PMWIN subsystem to a full 32 bit implementation. In addition, parts of PM are currently written in 16-bit Assembler, and are being rewritten in C for greater portability in preparation for a move to a portable version of OS/2.

Continuing object-oriented direction

PM already exhibits some characteristics of an object-oriented environment, especially in programming for the Workplace Shell and the System Object model (SOM) - see “System Object Model (SOM)” on page 93 . This direction will continue, not only in the method of programming, but also in the user interface, driven by the increasing object-orientation of SAA Common User Access (CUA). SOM itself will evolve to offer increasing object content to OS/2.

Distributed PM

This technology will allow an OS/2 PM application to be distributed across a network to other OS/2 PM systems, to X Terminals or workstations running the X Window system, or to DOS/Windows machines running an X server. This means that 32-bit OS/2 PM applications will be made available to 286 based DOS/Windows or OS/2 1.x PCs, and X terminals or workstations connected to the OS/2 PM application via a network using the X11 protocol. In essence, the OS/2 PM application acts, in X terminology, as an X client. X server function (where an application running on a UNIX RISC machine can appear in a window on a PC running OS/2) has already been delivered with TCP/IP for OS/2 v1.2. These developments allow OS/2 to integrate more fully into the open systems world as well as providing a wider user base for OS/2 PM applications.

Object-oriented environments

In looking ahead to the future, a key challenge to the IT industry, customer and commercial developers alike, is the ability to deliver software solutions in a more timely and cost effective manner. Object-oriented (OO) technologies have proven to make software development and maintenance easier, faster, less prone to error, and therefore less expensive. Object-oriented programming is now being endorsed by the software industry as a more productive programming approach, encouraging greater re-use of code. It is also more suitable for object-oriented user interfaces, like OS/2 2.0′s Workplace Shell.

One of IBM′s goals for Personal Systems operating systems is to provide an object-oriented development and operational environment that is customisable, allowing developers and users to take incremental advantage of new technologies, while protecting their investments in existing applications.

We are well on our way to achieving this with OS/2 2.0. OS/2 itself already exhibits many aspects of object-oriented programming, particularly in the System Object model (SOM), the object model that underpins the Workplace Shell (see “System Object Model (SOM)” on page 93). Furthermore, the Workplace Shell itself has many characteristics of an object-oriented user interface (see “An INFORMATION-oriented user interface” on page 79)

IBM also intends to encourage object-oriented development for OS/2 2.0. OS/2 will continue to be enhanced with extensions to improve support for object-oriented programming languages, as well as enhancing SOM. SOM is a technology for packaging software objects. It allows objects to interoperate without requiring that they be written in the same language or compiled together. SOM is the key element which allows applications to integrate with the desktop facilities of the Workplace Shell. It supports objects written in C today. C++ and other languages will be supported in the future. IBM intends to extend SOM, to provide application frameworks to further increase programmer productivity, and to provide tools to assist end users with visual programming (assembly of objects). IBM will enhance SOM to be compliant with the Object Management Group′s (OMG′s) Common Request Broker Architecture (CORBA). OMG is the leading object technology consortium that has published CORBA. Adherence to this standard will provide management for distributed heterogeneous networks comprised of multivendor operating platforms.

Customers can start to build applications with object-oriented tools today under OS/2, not only with C using SOM, but also with products like Enfin/2 and Digitalk′s Smalltalk/V PM . Furthermore, Borland have announced an agreement with IBM under which Borland will supply their C++ object-oriented development tools for OS/2 2.0, thus helping programmers who want to use C++ for object-oriented development under OS/2 2.0. Other object-oriented development tools vendors are also endorsing the OS/2 environment by supplying their tools for OS/2 2.0.

IBM has made significant investments of its own in object-oriented programming, of which SOM and the Workplace Shell are clear examples. Through alliances, IBM is supplementing its own technology with that of the leaders in the object-oriented industry, in order to provide for the future development of OS/2.

The agreement between IBM and Apple, announced in October 1991, included both companies′ intention to form a joint venture company, Taligent, to develop a next-generation operating environment based entirely on object-oriented (OO) technology. IBM and Apple had independently come to the conclusion that object-oriented technology is key to solving the ever-increasing application development challenge. The two companies had been involved in developing object-oriented technology in-house for some time. One of the key objectives in forming Taligent was to bring the benefits of this technology to customers sooner than either partner could have achieved alone.

IBM will licence Taligent′s system, as will Apple and other hardware system manufacturers. IBM and Apple are both contributing technology to the Taligent venture. Among Apple′s contributions was their “Pink” technology and among IBM′s offerings was SOM.

Taligent′s technology will be used separately by both Apple and IBM as the core of new products expected in the mid- to late-90s. These products will operate in parallel to, and complement the evolution of, OS/2 and AIX. Over time IBM intends to utilize a subset of Taligent′s object services and frameworks to benefit OS/2 application development and enable future compatibility with Taligent′s environment.

Taligent′s system software environment will emerge in the mid-90s, targeted at specific market segments that can readily leverage this technology. Once the Taligent system is available, IBM intends to create an OS/2 “personality” as a vehicle for protecting application investments. This will provide a path for customers who want to take advantage of the Taligent system.

Therefore, in the mid to late 90s, OS/2 and the new object-oriented environment will co-exist, and customers will be able to choose between an OS/2 system or a full object-oriented environment. But even then, not all customers may need or want to adopt a full object-oriented environment, and will stay with OS/2 as OS/2′s object-oriented content increases. OS/2 will remain the mainstream IBM Personal Systems operating system, and will continue to be developed and enhanced through the 90s. Since OS/2 itself will become increasingly more object-oriented, and its applications will run in the new environment, evolution towards a full object-oriented environment can be made gradually and seamlessly over time. IBM believes there will continue to be many customer and hardware requirements that demand OS/2, after the availability of Taligent.

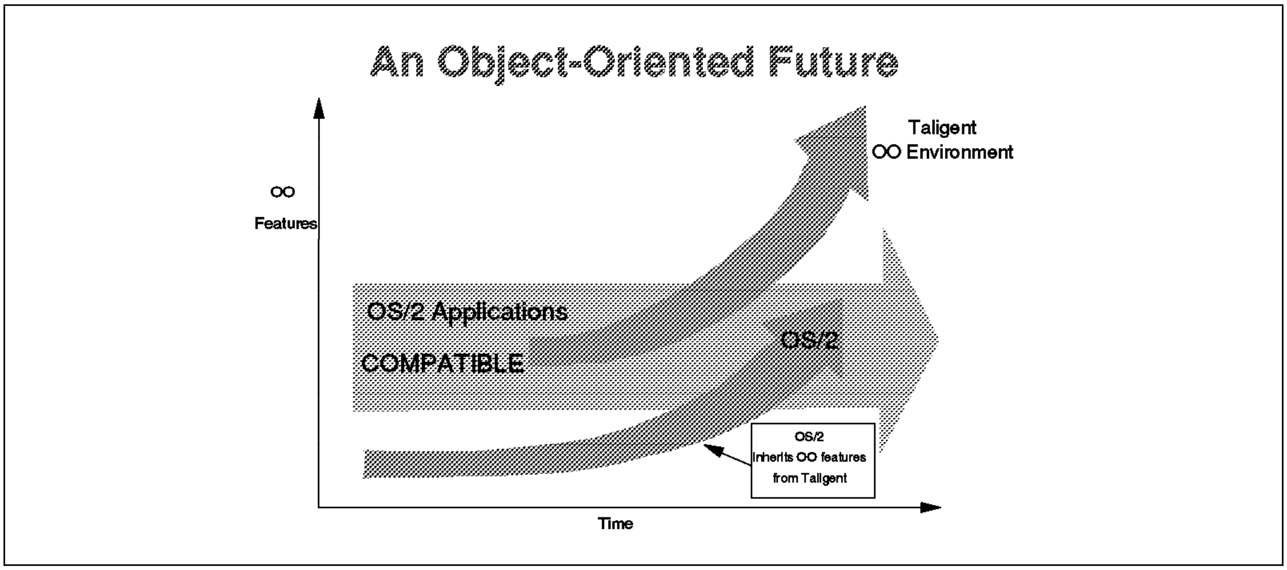


Figure 29. OS/2 - an object-oriented future

This approach is possible because OS/2 2.0 already contains many object-oriented features, such as the System Object Model, and the Workplace Shell user interface. Because of OS/2′s object-oriented content, and because OS/2 applications will continue to run in the new system, OS/2 is the platform best suited to evolve towards a fully object-oriented environment.

Some have tried to suggest that IBM′s direction towards a future object-oriented environment in some way threatens OS/2. The opposite is the case: the IBM-Apple announcement endorses and strengthens the OS/2 strategy. Since OS/2 already displays significant object-oriented content, and since OS/2 applications will be fully supported in the new object-oriented environment, it follows that using, programming for and deploying OS/2 is an excellent way to secure an object-oriented direction. If the future is object-oriented, OS/2 is the way to get there.

Distributed computing

IBM recognises the increasing importance of open networking, linking to mixed vendor environments, both clients and servers. In May 1990 the Open Software Foundation (OSF) announced the Distributed Computing Environment (DCE), a group of technologies aimed at simplifying the work for users and application developers in these complex computing environments. In endorsing DCE, IBM subsequently announced not only support for DCE on AIX but also the intent to extend the Systems Application Architecture (SAA) to incorporate key elements of DCE. As one of the SAA systems, OS/2 will be a platform for the delivery of the key elements of DCE.

For those not familiar with DCE, it consists of a number of key technologies. Of these, six are of particular interest for distributed environments including OS/2, and can be summarised as follows:

Remote Procedure Call (RPC)
The basic notion is that procedures called by an application may actually be run on a computer somewhere else in the network. The RPC mechanism takes care of the communications details so that writing distributed applications approaches the simplicity of applications on a single machine.

Distributed Naming Service
This provides a single naming model throughout the distributed environment. Resources such as servers, files, disks, or print queues are identified by name independent of the physical location in the network. Full X.500 support is provided.

Time Service
Many distributed applications need a single time reference to properly determine event sequencing and duration. The time service provides a mechanism for synchronising each computer in the network to a recognised time standard.

Security Service
Provides the network with authentication, authorisation, and user account management. Authentication validates the identity of a user or service to prevent fraudulent requests. Authorisation is the process of determining whether an authenticated user should have access to a resource. These facilities are made available through a secure communications capability provided by the RPC.

Threads Service
A facility to support concurrent programming, much like threads in OS/2. This will be used by other DCE components to implement their services.

Distributed File System
Joins the file systems of the nodes in the network through a consistent interface that makes global file access as easy as local file access. The Distributed File System should provide users with a uniform name space, file location transparency, and high availability.

IBM′s intention is that OS/2 LAN Server will evolve towards a full distributed LAN system, providing DCE services as described above. OS/2 LAN Server will incorporate architectures and industry standards which fully meet customer requirements and will include technologies such as:

- Distributed Relational Database Architecture (DRDA)
- Open Software Foundation (OSF) Distributed Management Environment (DME)
- OSF′s Distributed Computing Environment (DCE)
- Transarc/ENCINA Online Transaction Processing

OS/2 is a key part of IBM′s plans for the delivery of key elements of distributed computing. IBM has already shown technology demonstrations of OS/2 participating in a distributed environment along with AIX. Distributed computing support will make OS/2 an even more open platform, as both a client and a server.

AIX interoperability

IBM is committed to providing greater co-existence and interoperability between SAA and AIX systems. OS/2 will participate in this strategy. This does not involve merging function, but retaining the unique strengths of the two systems. The aim is to make it easier for customers to build mixed networks combining the best of both OS/2 and AIX, RISC and Intel hardware, according to their applications.

Part of this strategy involves putting compatible key systems components, such as relational database, on both platforms. As well as providing a 32-bit database on OS/2 in future, IBM will also provide a compatible 32-bit database for AIX, and support client operations between AIX and OS/2 databases. Another element of the strategy is to improve co-existence by providing networking support. Products like TCP/IP for OS/2, already exist to fulfil this requirement, and IBM intends to strengthen OS/2-AIX inter-networking in future through DCE, DME and other technologies.

Future Windows compatibility

OS/2 2.0 supports Windows 3.0 applications by including modified Windows code under the cross-licencing agreement with Microsoft. The March 1992 release of OS/2 2.0 does not contain support for Windows 3.1, since at the time when OS/2 2.0 shipped, Windows 3.1 was not available. It would be impossible to plan support for a product that had not shipped by the time OS/2 2.0 was available. IBM plans, and is able, to support Windows 3.1 at a later point, if it proves necessary (see "OS/2 1992 developments" on page 109 and "Windows 3.1" on page 66). In fact, on April 7th, 1992, the day after Microsoft shipped Windows 3.1, IBM demonstrated the Windows 3.1 Program Manager running under OS/2 2.0.

It is important to understand that OS/2 2.0 supports Windows ***applications***, not Windows. Windows support in OS/2 is only relevant for the Windows applications that OS/2 customers wish to use. IBM continues to licence the DOS/Windows source code, and has rights to all source code in development until September 1993, in order to continue to provide support for 16-bit Windows applications. Microsoft has not made its future plans clear on 16-bit Windows application support, but it may be difficult for Microsoft to persuade its customers to accept future versions of 16-bit Windows that do not support today′s 16-bit Windows applications. IBM considers that it has the ability to provide support for today′s generation of Windows 16-bit applications, as long as is necessary.

There are, as yet, no 32-bit Windows applications available, because Microsoft does not plan to ship its first 32-bit Windows platform until some time in 1993. However, there are a growing number of OS/2 applications: hundreds are already shipping, and over 1000 have been announced for shipment by the end of 1992 or early 1993. OS/2 will therefore have the largest number of 32-bit applications for some time. Therefore, the trend in the marketplace towards real 32-bit applications, and the continuing success of OS/2, may make Windows support a moot point. The issue may be rather what levels of OS/2 32-bit compatibility can Microsoft build into their future products.

Since many of the applications currently available or under development for 16-bit Windows 3.x will also be available for 32-bit OS/2, future Windows compatibility may cease to be an issue, since running native OS/2 versions of applications will be superior to running the Windows versions even in WIN-OS/2 (see "Porting Windows applications to OS/2" on page 67). Support for Windows applications is intended mainly as a migration to OS/2 2.0, but investment in Windows applications should not be considered a long-term strategy for customers planning an OS/2 future. Many vendors now understand the need to provide real OS/2 versions of their software for competitiveness′ sake, and are not relying on running their Windows version under OS/2 to address the OS/2 opportunity properly. Furthermore, the evidence suggests (see "Windows applications" on page 50) that Windows applications have not made anything like the same impact on the market as Windows itself. IBM will continue to monitor the continued need for supporting Windows applications, and will exercise its skills and intellectual rights to do what its customers require.

Portable version of OS/2

The joint IBM-Microsoft statement at Comdex in November 1989, indicated that both companies were committed to producing a version of OS/2 that could run on non-Intel processors such as RISC, and include US Department of Defense security at the C2 level and symmetric multiprocessing. IBM remains committed to this direction.

The most important part of the work in portability lies in ensuring that ***applications*** are easily portable to any new platform. The other element of portability is to provide a portable operating system kernel (the low level code that interfaces with the processor). In fact, the importance of the latter has often been exaggerated in relation to the former: there is no point in having a portable kernel if you cannot move your applications easily.

IBM has already made great progress in the first step with the 32-bit OS/2 API, which eliminates many of the dependencies of the 16-bit Intel-based architecture. This provides the potential of a 32-bit code base which is ready to take advantage of a future move of the kernel to other processors. The kernel may change underneath, but the API must be established and preserved. It is also clear in the design of OS/2 2.0 how IBM is already preparing many of the key components of OS/2 (such as the subsystems like PM and the Workplace Shell, and the API set) for easier portability and to accommodate future directions. In contrast, although Microsoft has publicised its portable kernel, the other elements required for portability, specifically a full 32-bit API with multi-threading, interprocess communications and advanced graphics, are not yet delivered. And it is unclear from their public disclosures about Win32s (see "OS/2 - a 32-bit API - TODAY" on page 75) how to make a straightforward migration to take advantage of RISC, without changes to the code base. The Windows API today remains 16-bit, and 32-bit exploitation, including the full benefits of portability, remain a future promise.

As for the choice of kernel, IBM has a variety of options, through licencing agreements. IBM is working on a range of technologies, and will announce its direction when it has fully evaluated the available technology in the light of customer requirements. Suffice it to say that although the "NT" in Microsoft′s future system, Windows/NT, stands for "New Technology," the idea of a portable micro-kernel is not new, and other examples already exist in a mature form, including the OSF′s micro-kernel based on the Carnegie-Mellon Mach technology. IBM has chosen not to integrate Windows/NT code into its future offerings. But NT is by no means the only way of achieving the objectives stated in November 1989.

As far as customers and application developers are concerned, if portability is a key issue, 32-bit application development under OS/2, rather than migration from 16-bit DOS and its extensions, is the clearer path.

Conclusion

OS/2 is therefore not only taking advantage of today′s requirements, and the wide range of DOS, Windows and OS/2 applications that already exist, but provides an immediate future for powerful 32-bit developments, including a 32-bit API and the potential to integrate with the new OS/2 Workplace Shell user interface. The requirements of the workstation of the late 1990s for an object-oriented platform supporting multimedia, DCE and systems management, can only be served by a robust, architected 32-bit platform. That platform is OS/2.

Appendices

Appendix A. Comparison tables

DOS environments

The following notes refer to the table on the next page.

Notes:

1. See “Comparison with memory usage under DOS” on page 36, “Amount of memory” on page 18 and “Memory” on page 54.
2. See “MVDM memory management” on page 20 and “Expanded and Extended Memory” on page 38. Also see “Standard mode” on page 48 for Windows 3.x′s EMS support.
3. The phrase “none/switch” means that no individual DOS application can overcommit memory, but the real mode portion can be moved to disk to make room for another DOS application. However, extended or EMS memory allocated by the application is not switched to disk. OS/2 1.3 can swap the DOS application to disk when running protected mode applications. Windows 3.x 386 enhanced mode can overcommit up to four times the physical memory on the machine. OS/2 2.0 is limited only by the amount of available disk space.
4. Although the default is to swap through the file system, Windows 3.x 386 enhanced mode does allow swap space to be pre-allocated, and gain improved performance by avoiding the DOS file system. Because this disk space is pre-allocated, none of it is available to be shared dynamically for any other use. OS/2 2.0 implements access to the swap space via the file system for both FAT and the HPFS. The OS/2 2.0 implementation provides the flexibility of a dynamically sized swap file combined with good performance.
5. OS/2 2.0 and Windows 3.1 provide windowing of DOS applications on the Workplace Shell desktop in all text and VGA graphics modes, while Windows 3.0 only supports windowing of text and CGA modes. According to Microsoft, Windows/NT will not support windowed VGA graphics.
6. Although Windows 3.x does include a print spooler, and printing concurrently from DOS applications is not supported. Windows permits only one DOS application to print and requires that other DOS applications be suspended if they attempt to print concurrently. Use of a DOS print spooler (loaded prior to Windows 3.x.) is not a viable solution, since printing concurrently from multiple DOS sessions causes all the output to be jumbled together on the same page. Windows does warn of a device conflict in use of the printer in the latter case and offers a choice on how to proceed, but whatever the choice made, the same incorrect output results. OS/2 2.0 provides correct spooling of printer output from concurrent DOS applications.
7. Some DOS applications are extremely timing sensitive, mostly communications applications using high data rates. These are supported in all systems when the application is in the foreground. However, in DOS, OS/2 1.3., Windows 3.0 real mode, and Windows 3.x standard mode, the application is suspended while in the background and so cannot satisfy any timing constraint. In 386 enhanced mode, Windows 3.x provides the option of setting exclusive mode, so a timing sensitive application can be run in the foreground without interference due to time slicing. However, all other applications are suspended while this DOS application is running in this mode. In some cases the need for exclusive mode can be avoided by manually adjusting the relative priorities. The ability to dynamically manage priorities permits OS/2 2.0 to multi-task timing critical applications both in the foreground and background. Detailed configuration of idle time detection is also possible in the DOS Settings (see “DOS Settings” on page 41).
8. Although Windows 3.1 has addressed some of Windows′ limitations in terms of stability, it is still based on DOS, and therefore still prone to problems caused by DOS-based TSRs which take the system out of protected mode into real mode (see “Reliability and protection” on page 39 and “Reliability” on page 54).

Table 9. Comparison of DOS environments

			Windows 3.0 on IBM DOS 5.0			Win 3.1 Enh Mode with DOS 5.0	
	IBM DOS 5.0	**OS/2 1.3**	**Real**	**Standard**	**Enhanced**		**OS/2 2.0**
Conventional Memory - with EMS & Mouse ***(1)***	622 KB 601 KB	529 KB	558 KB	571 KB	569 KB	577 KB	633 KB
Memory w/ LAN Attach - configured as PCLP Receiver	543 KB	486 KB	386 KB	391 KB	441 KB	477 KB	633 KB
Memory w/3270 Attach	522 KB	495 KB	486 KB	492 KB	541 KB	549 KB	633 KB
Extended Memory (XMS)	16 MB	none	16 MB (total)	16 MB (total)	16 MB (total)	16 MB (total)	16 MB(per app)
EMS 4.0 Memory***(2)***	16 MB	none	16 MB (total)	none	16 MB (total)	16 MB (total)	32 MB(per app)
Physical RAM for DOS	0-1 MB	0-640 KB	0-1 MB	0-1 MB	Any/Paged	Any/Paged	Any/Paged
Memory Overcommit***(3)***	None/Switch	None/Swap	None/Switch	None/Switch	4 × RAM	4 × RAM	Avail Disk
Swap File***(4)***	File System	File System	File System	File System	Physical (Preallocated) or File System	Physical (Preallocated) or File System	File System (Dynamic)
Number of DOS Apps	16	1	16	16	16	16	>32
Background Execution	No	No	No	No	Yes	Yes	Yes
Invocation	Shell/Cmd	Icon	Icon	Icon	Icon	Icon	Icon
Windowed***(5)***	No	No	No	No	Yes (Text, CGA)	Yes (Text, VGA)	Yes (Text, VGA)
Cut & Paste	No	No	Yes	Yes	Yes	Yes	Yes
Print Spooling***(6)***	Yes	Yes	No	No	No	Yes	Yes
Installable File System	No	Yes	No	No	No	No	Yes
Direct Hardware Access	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Timing Dependent Apps.***(7)***	Foreground	Foreground	Foreground	Foreground	Exclusive Option (Manual)	Exclusive Option (Manual)	Foreground/ Background (Automatic)
DPMI	No	No	No	No	Yes	Yes	Yes
VCPI and DOS Extenders	Yes	No	No	No	No	No	No
Continues after serious Application errors ***(8)***	Rarely	Rarely	Rarely	Rarely	Sometimes	Often	Yes

OS/2 2.0 compared with Windows 3.0/3.1

Table 10 (Page 1 of 3). OS/2 2.0 compared to Windows 3.0/3.1

	Windows 3.1	**OS/2 2.0**
Hardware		
Processor	286; 386SX+ ***(1)***	386SX+
Minimum hard drive	Approx 9MB	Approx 13MB
Hard drive for Full install	11MB - plus 50% of remaining partition for swap file (default) ***(2)***	27.6MB
Largest hard drive	1GB	64GB (HPFS)
Largest file size	1GB	2GB
SCSI exploitation	No	Yes
File System options	FAT only	Enhanced FAT or HPFS
Memory		
Physical Memory Limit	> 16 MB	> 16 MB
Virtual Memory Limit	4 x Physical	512 MB per process (or Disk space)
Memory Model	Segmented (64 KB)	Flat memory objects
Multi-tasking ***(3)***		
Multi-tasking - DOS Applications	Time Slicing	Pre-emptive Time Slicing
Multi-tasking - Windows & OS/2 Apps	Co-operative	Pre-emptive
Priority	Static (set by user)	Dynamic
Dispatchability	Process	Thread
System Services	Serial	Parallel / Overlapped
Reliability/Protection ***(4)***		
Protection between Applications	Limited	Protected
Kernel protection - DOS Applications	Limited	Protected
Kernel protection - Windows & OS/2 Apps	Limited	Protected
Remains in protected mode ***(5)***	No - access to real mode possible	Yes
Reliability/Availability/Service (RAS) Support ***(6)***	None in 3.0. In 3.1: Error Logging/Dump via Dr.Watson; No Trace APIs or Trace formatting	Standalone Dump, Error Logging, Trace Utilities & APIs
Compatibility		
Multiple Concurrent DOS Applications	Yes (enhanced mode only)	Yes
Windows 2.x Applications	No	Yes
Windows 3.0 Applications	Most ***(7)***	Most
Clipboard support	Windows and DOS only	Windows, DOS and OS/2
DDE support	Windows apps only	Windows and OS/2 apps
OLE support	Yes	Yes
16-bit OS/2 Applications	No	Yes
32-bit OS/2 Applications	No	Yes
Printing and Fonts		
Print spooling	Limited ***(8)***	Yes, for all applications
Adobe Type Manager standard	No	Yes
Network printing support	Some	Full ***(9)***
Background printing performance	Unpredictable	Predictable ***(10)***

Table 10 (Page 2 of 3). OS/2 2.0 compared to Windows 3.0/3.1

	Windows 3.1	**OS/2 2.0**
National Language Support		
Number of Language Versions	12	16
Data Interchange	ISO8859/CP819 (different from DOS)	CP850 (consistent throughout OS/2)
Host connectivity/Interchange	3rd party	Included in Extended Services for OS/2
Code Page	Single	Selectable
Other Factors		
Full 32-bit APIs	No	Yes
Concurrent High Speed Comms	Unreliable	Yes
Background Comms ***(11)***	Unreliable	Yes
OEM Hardware Support	Yes	Yes
Development Tools	Yes	Yes
Command Language	.BAT	.BAT, .CMD and REXX
Installation migration for existing apps	Limited	Yes
User Interface		
CUA compliance	Graphical Model (′89)	Workplace Model (′91)
Icons representing non-loaded files on desktop	No (3rd party)	Yes
Place icons anywhere on desktop	No (files in File Manager, programs in Program Manager, printers - no icons)	Yes
Group windows	Single-layer (can′t put group inside another group)	Multi-layer, hierarchical folders
Customise GUI look/feel	No	Yes (Workplace Shell, Windows 3.x, OS/2 1.x)
Context Menus	No	Yes
Drag/Drop across entire Shell environment	Windows 3.0 - No. Windows 3.1 - File Manager only	Yes
Object Management	No	Yes
Graphical Install	Yes	Yes
Intelligent fonts	Windows 3.0 - No (ATM separate purchase). Windows 3.1 - Yes (TrueType - 650 fonts)	Yes (Adobe Type Manager for OS/2 & Windows - 1200 fonts)
Long file names	No	Yes
Applets	Yes	Yes
Consistent GUI logon	No - requires Network vendor utility	Yes
Interactive Tutorial	Yes	Yes
Command Reference	No	Yes
Advanced Connectivity***(12)***		
Client and Server platform	No	Yes
Multiple Concurrent Protocols	Limited	Yes
SNA LU6.2	3rd party	Yes
APPN	3rd party	Yes
TCP-IP	3rd party	IBM TCP-IP for OS/2
Systems Management	3rd party	Various from IBM ***(13)***

Table 10 (Page 3 of 3). OS/2 2.0 compared to Windows 3.0/3.1

	Windows 3.1	**OS/2 2.0**
SQL Database Server	MS SQL Server (requires OS/2)	Yes
SQL Database Client	3rd party	Yes
NFS	3rd party	IBM TCP-IP for OS/2
X-Windows server	3rd party	IBM TCP-IP for OS/2

Notes:

1. Although Windows 3.1 will run on a 286, this does not enable enhanced mode support, which is required to be competitive with OS/2 function (multi-tasking DOS applications, demand paging, 32-bit support).
2. See “Disk space” on page 129
3. See “Multi-tasking of DOS applications” on page 35, “Multi-tasking” on page 25 and “Performance” on page 55
4. See “Reliability” on page 54, “Reliability and protection” on page 39 and “Protected mode operation” on page 14.
5. See “Reliability” on page 54, “Reliability and protection” on page 39
6. Windows 3.1 uses Dr.Watson, OS/2 2.0 uses SYSLOG and Standalone Dump
7. 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 all but a very small minority of Windows 3.0 applications, as well as all the Windows 2.x applications that Windows 3.1 will no longer support (no real mode support provided) - see “Real mode (NOT AVAILABLE IN WINDOWS 3.1)” on page 48 and “386 Enhanced mode” on page 58
8. Print spooling is not provided by Windows 3.1 for DOS applications, only for Windows applications. OS/2 2.0 provides print spooling for DOS, Windows and OS/2 applications
9. OS/2 2.0 has extensive user print management capabilities (40 APIs vs 12 APIs in Windows 3.1) for querying, holding, releasing and deleting jobs (including a graphical view of job and queue status).
10. OS/2 consistently outperforms Windows with background print operations, in multi-tasking environments
11. See notes on Background Execution in Appendix A on DOS applications.
12. OS/2 2.0′s “Yes” answers here are all using Extended Services for OS/2 except where stated. It is important to note that the Windows column refers to Windows specific programs (i.e. written to explicitly take advantage of Windows GUI, memory addressability, or time-slicing). Although there are many DOS connectivity options, and they may be usable under Windows, the integration of these complex subsystems and any co-residency of two or more options (eg TCP/IP and SNA) is completely the responsibility of the customer as a custom integration effort. Moreover, Windows on DOS has substantial architectural limitations which make multiple network connections difficult to integrate than under OS/2 (lack of memory, lack of protection, and weak multi-tasking support). OS/2′s base environment provides tools and system support designed to allow this type of multi-connectivity installation. Besides, all the extra software required for these functions under OS/2 comes from IBM, and one can therefore anticipate a greater degree of integration (see “OS/2 for client-server” on page 98).
13. See “Systems management” on page 103

Appendix B. Hardware requirements and performance

The main reference to OS/2 system requirements is the OS/2 2.0 Information and Planning Guide (G326-0160-00), and the equivalent guides for Extended Services for OS/2 and OS/2 LAN Server. Anyone planning installation of OS/2 across several workstations should obtain these guides. In this section we will discuss only the key issues - for the detail, you should consult the Information and Planning Guide. This section only discusses issues relating to the base system, not to the extensions, though many of the principles apply to both.

Minimum requirements

OS/2 2.0 is designed for personal computers with the following minimum requirements:

- Intel (or compatible) i386SX microprocessor or above
- 4MB of memory
- 60MB hard disk with 15-30MB of free disk space
- 2-button mouse or other pointing device

Please note that this *is* a minimum. Many will be satisfied with this configuration if simple tasks are undertaken, but for more powerful multi-tasking and larger applications, more memory may be required. There is a trade-off to be made between memory and performance, in particular: OS/2 can work around a limit in physical memory by swapping to disk, but since disk access is slower than memory access, this will tend to reduce overall performance, which will show itself more in switching between applications than in the performance of an individual session.

OS/2 2.0 will not run on machines equipped with an Intel 80286 processor. Therefore, computers such as the IBM PC AT, PS/2 Model 30-286, and Models 50, 50Z, and 60 cannot be used with OS/2 2.0. However, OS/2 2.0 does support non-386 based machines that have been upgraded with a 386 or 486 processor using the Aox Micromaster, Intel SnapIn, or Kingston SX/Now! card.

In addition, OS/2 2.0 is supported on a broad range of IBM-compatible systems, including models from Compaq, AST, Olivetti, Toshiba, Hewlett Packard, Dell, Gateway, Wang, DEC, NCR, Tandy, ACER, CompuAdd and many others.

The certification of compatibility is based on IBM′s tests of key functions of OS/2 2.0, based on selected model configurations provided by the manufacturers of these non-IBM hardware systems. Test results are available on CompuServe, IBM Forums (OEM and OS2ARENA), and other bulletin boards. If you need additional information, please consult your hardware supplier. If you are using a specific model, and wish to have it certified, please check with your IBM representative or Authorised Dealer. If it has not been tested by IBM, IBM welcomes your help in making contact with the manufacturer to have the machine tested. IBM wishes to make the range of OS/2 compatibility as broad as is possible.

In practice, OS/2 has been designed to operate correctly with a wide range of hardware, and the evidence from many of the users who have registered their copies already (see “Broad hardware support” on page 31) indicates that OS/2 2.0 is already running on a large number of different machines.

Processor

Because of its 32-bit addressing power, the OS/2 2.0 operating system requires a computer that has a system unit equipped with an Intel (or compatible) i386 (or higher) microprocessor. The i386SX microprocessor provides adequate performance for those who work in lower-demand application environments. In most environments that demand multiple concurrent processes, the i386DX will be adequate for satisfactory performance. The IBM 386SLC chip offers high performance from a 386SX design, and along with the new 486SLC chip, offers outstanding price-performance for an OS/2 system. Systems equipped with the 386SLC represent a viable entry point for a multi-tasking OS/2 client. 486SLC-based systems represent an excellent option for throughput even of heavy multi-tasking loads. With prices of 486SX systems being reduced, they can be a cost-effective option for client workstations. For computers that will be used as network servers, consider the i486 series. Also consider the 486 as a base processor for those who expect to switch frequently and rapidly among a large number of concurrent tasks.

Memory size

Memory and disk storage are closely related because of the ability of the operating system to manage the allocation of memory resources between real physical memory and hard disk space. In general, it is important, when calculating memory and disk requirements, to work on the concept of “working set,” in simple terms the sum of physical memory and swap file size. The balance of physical memory installed versus swap space is key to determining overall performance - the more working set is gained from swap space rather than physical memory, the lower overall system performance.

4MB is adequate for an entry-level workstation running few tasks. This enables users to run applications or other system utility programs concurrently, but it presents a constrained environment (limited memory) for some large applications. In limited memory configurations, performance of applications might be reduced, particularly when the operating system is loading an application or switching from one application to another or to the desktop. This is as a result of paging, as the code required is loaded to memory from disk, and the least recently used memory is paged to disk to make room.

Although more tasks can be accommodated by swapping, performance degradation would suggest more memory would be required for more intensive multi-tasking. However, an OS/2-only workstation (ie no DOS or Windows applications) with a LAN requester can perform acceptably in a 4-5MB configuration. This indicates that the entry level configurations are suitable for some client scenarios.

To estimate probable working set requirements, estimate 2.5MB for the base system without any applications, 0.6MB per concurrent DOS session, and 1MB per concurrent WIN-OS/2 session. (The latter two figures assume the default amounts of XMS, EMS and DPMI memory installed, which are 2048K, 2048K and 2MB respectively - the actual figure may therefore be more, or less, depending on the settings chosen.) Thus a system running DOS and Windows applications can run satisfactorily in 6MB. A connected workstation (eg LAN requester plus 3270 emulators) will require around 8MB (though note the low entry point above for OS/2-only clients with just a LAN requester).

The amount of memory installed can be an important factor in performance, particularly when swapping between applications. Since working sets of 1-2MB or more are being switched, the performance difference between a 4MB and a 6MB configuration can be substantial; a further increase to 8MB is appreciable, but not usually to the same extent. The system tends to handle overcommitment on memory well above the 8MB point. Large applications and heavy multi-tasking will benefit from 8MB or more.

Though adding more memory inevitably adds to the cost of the system, it is to be remembered that the cost of memory is decreasing. Although OS/2 2.0 will tend to require more memory than Windows 3.1

for equivalent software configurations, the cost of the memory will often be less than $100 at today′s memory prices. Many customers are finding that the extra costs for memory are compensated by the increased function and stability that OS/2 offers. Customers hoping to achieve the combination of a 32 bit, stable multitasking platform, on an affordable hardware configuration, may consider this trade-off between function and system requirements. Furthermore, independent analysts have predicted that Windows/NT will require an even higher configuration than OS/2 2.0. Gartner Group has told its customers that it believes “a mainstream platform for Windows/NT will be a 486DX with 12 to 16 megabytes of RAM (and up) on the workstation”[24].

It is to be remembered that in all the above estimates for OS/2 memory requirements, results will vary with the configuration of the machine and the applications run. Also, performance is a subjective matter - few people agree on what is acceptable relative to the cost of a configuration. Customers are recommended to carry out their own benchmarks to assess the most suitable configuration for their requirements.

Disk space

The requirements stated in this section are for the operating system, swap space, and print spool jobs. They do not consider space required for extra applications and data.

Although the maximum space required for OS/2 installation is 27.6MB plus swap space, the minimum installation can be as little as 12.7MB (no DOS or Windows support), 14.6MB (DOS, but no Windows support) and 16.9MB (DOS and Windows support). These are taken from the cumulative total in the selective install option; the total in each of these scenarios can be lower still by eliminating selected files after installation. Typical scenarios will require less than 20MB and sometimes only 15MB.

The decision of whether to install certain components can make a large difference to the space taken: WIN-OS/2 requires 3MB, Productivity applications 1.3MB, tools and utilities 3.6MB and games 0.8MB. The optional systems utilities (eg BACKUP, ATTRIB, TREE, RECOVER, SORT), which are often removed after installation on both OS/2 and DOS machines, take up 1.2MB.

The amount required for swap space will vary according to system load. The system allocates a minimal swap file on loading; this defaults to a figure based on the physical installed memory (the Information and Planning Guide has a table with the detailed figures), for example, a 6MB swap file for a system with 4MB of memory, a 4MB swap file for an 8MB machine. The figure can be modified by the user at installation time, or later by modifying the SWAPPATH parameter in CONFIG.SYS. The installation program also allows the user to set the location of the SWAPPATH to a different partition. This can be a good idea if the main partition is 30MB or less.

If you are choosing the type of machines to use as OS/2 machines, remember that for any environment that uses virtual memory (swapping to disk to create more memory), a fast disk is recommended. This is an important factor, especially for those who expect to switch frequently and rapidly among a large number of concurrent tasks.

Some figures in the industry have criticised the amount of space taken by OS/2 2.0, but there is always a cost associated with high function. In fact, if you consider what OS/2 offers:

- a full DOS environment
- a complete Windows environment
- a copy of the OS/2 kernel and system, with compatibility for both 16- and 32-bit applications
- a font manager for both OS/2 and Windows (ATM), plus associated fonts
- a series of utilities, including a file search tool, a charting package, and a personal organiser

[24] Gartner Group - Personal Computer Research Notes, P-230-853, July 31, 1992

- an object-oriented graphical shell

and then compare the likely disk space required to assemble all this function under either DOS or Windows (here are some suggested equivalents):

- DOS version 5.0
- Windows 3.0 or 3.1
- ATM for Windows
- (For example) XTree Gold, Micrografx Charisma, and Polaris Packrat
- Norton Desktop for Windows

then it is likely you have taken at least as much space (and probably more) than OS/2 2.0. (And that′s ignoring the extra monetary cost of acquiring these separate third party applications.)

Furthermore, the differences between Windows 3.1 and OS/2 2.0 in disk space are exaggerated. Although Windows will nearly always take less for the base installation, you can see from the above figures that OS/2 2.0 can come within 3 or 4MB of the Windows total. Moreover, the figures quoted by many people do not mention the fact that if you choose, during Windows installation, to have a permanent swap file (chosen by many Windows users to gain acceptable performance), the default is to take 50% of the free space on your partition. This can result in an ***overall*** disk usage (system plus swapper) of close to OS/2′s and sometimes more (dependent on the size of the partition). OS/2 allows detailed and granular control over what is installed, so it is at least easy to know what your likely requirements are.

Furthermore, many newer applications take large amounts of disk space anyway. A full installation of Word for Windows version 2.0 takes up to 15Mb of disk space. This puts the requirements for the operating system into some perspective.

For those concerned about disk space in general, data compression tools such as Stacker, which are available on DOS, are under development for OS/2. Stac Electronics have announced their intention to produce an OS/2 version of Stacker, which will be available before the end of 1992. Other vendors are producing OS/2 disk compression tools, which are expected to ship in 1992.

Performance considerations

IBM′s performance aims for OS/2 2.0 are as follows:

- Close to DOS for single tasking DOS applications
- Close to Windows 3.x for single tasking Windows applications
- Superior to Windows for multi-tasking
- Equivalent to or better than OS/2 1.3

It is very important to understand that achieving these aims is subject to the obvious limits on comparing single tasking performance:

- DOS applications which are processor-intensive, or require continual servicing of interrupts, are always going to perform better in DOS where they do not have to share the CPU with other processes. However, despite this, there is little performance loss for most DOS applications in a single tasking scenario in a full screen DOS session. In a windowed session, screen scrolling and update will be affected by runnning the application within a window, and the extra virtualisation that needs to occur. This will particularly affect applications running in graphics mode. If higher performance is required, switch to full screen (the Alt-Home toggle allows flexibility: use full screen most of the time for maximum performance, but switch to windowed when interaction with the rest of the system (eg clipboard) is needed. On the other hand, DOS applications that are disk and I/O-intensive (such as database programs) can perform as much as 200-300% faster, due to the superior caching in the OS/2 FAT system, or using HPFS (see “Enhanced FAT” on page 30). As most applications are a mix of disk and processor operations, comparisons will differ according to the application chosen.

- Although for Windows applications, single tasking scenarios may be between 5 and 20% slower, this is hardly the best basis on which to compare performance (since Windows is supposed to be a multi-tasking environment as well). The differences mainly relate to application load time (usually caused by the fact that the Windows application often loads into a separate VDM, and then loads its own instance of WIN-OS/2. (A real comparison would be to add the time to load Windows under DOS to the load time of the Windows application, and compare it to the total under OS/2 2.0). Running the application in a Full Screen WIN-OS/2 session rather than Seamless sometimes yields better performance.
- In fact, DOS applications often run faster under OS/2 2.0 than under Windows 3.1, particularly disk-intensive applications, due to the superior FAT implementation (see above).
- When considering multiple application scenarios, OS/2 shows its superiority. Even running two applications together can show a big difference between OS/2 2.0 and Windows 3.1. Try, for example, formatting a diskette from the DOS prompt while running another application under Windows 3.1, and compare the operation in OS/2 2.0. In Windows, if the format operation is in foreground, it can starve the background process of processor time, and if put to background, may scarcely progress at all. Under OS/2 2.0 the share of processor load is much more even. This is hardly a heavy processing load. Another illustrative scenario comes from the testing of National Software Testing Laboratories (NSTL), an independent testing and evaluation organisation: to load MS Word for Windows on a PS/2 Model 57 with nothing else running takes 7.2 seconds with Windows 3.1 and 9.3 seconds with OS/2 2.0. If you do the same load with an XCOPY in the background, Windows load time jumps to 41.1 seconds, compared with 15.3 seconds for OS/2. When more than one task is being done, OS/2′s performance advantage becomes evident. Because of OS/2′s superior multi-tasking, it can run background tasks, such as file copying, communications, or spreadsheet recalculation, with no visible impact on foreground work. With Windows, the cursor movement can lag behind the mouse movement, and displaying of characters can lag behind keyboarding to the point where system becomes almost unusable until the background job is done.

Benchmarks from some sources are misleading, and are easily skewed by the choice of scenario. The best guide is to run your own, to simulate your own usage.

Tuning hints

When considering overall performance, there are many variables, including how balanced a processing load you want to achieve, or whether you want to give one process as much CPU as possible. Therefore hints can only be generic, and must be considered in the light of specific application requirements. Nevertheless, here are a few basic ideas. More are contained in the chapter on “Optimising Performance” in the OS/2 2.0 Installation and Planning Guide.

- To conserve OS/2 system resources and reduce memory requirements:
  - close applications when they are not going to be used again.
  - close folders if they are not needed.
  - move commonly used functions out of folders and to the desktop, and close the folder that contained the object.
  - However, be careful not to overload the desktop - keep only as many objects as you really need
- If certain applications are always used, put them in the Startup folder. This will increase boot time but have everything ready to run

- Pre-initialising the SWAPPER.DAT file at boot time will help if a certain group of applications are always loaded. The setting can be made in the SWAPPATH parameter in CONFIG.SYS. You should be careful to observe swap growth beforehand, and not set the initial swapper too large, lest you have a detrimental effect.

- The cache sizes for FAT and HPFS can improve performance if increased. Experimentation is the best way to discover the right results for your configuration

- Set the system path statements (PATH and LIBPATH especially) based on usage patterns, so that applications are quicker to load.

- Turn off public clipboard and DDE if sharing between Windows and OS/2 applications is not required

- Load Windows programs into a single WIN-OS/2 session, except those that require greater protection. Ensure that the DPMI_MEMORY_LIMIT is set appropriately

- If XMS, EMS and DPMI memory are not required (as is the case for many DOS applications), set them to zero in DOS Settings

- DOS Settings parameters can have substantial effect on relative performance of individual applications, and on their effect on others when running in the background. See “DOS Settings” on page 41 and *Version 2.0 Volume 2: DOS and Windows Environment (GG24-3731-00)*. Remember that many applications require little attention when in background, as they are usually waiting for input, and can therefore safely be given little CPU.

Appendix C. Further reference materials

Here are some useful publications on OS/2 2.0 that are either already available or about to be published:

IBM ITSC OS/2 2.0 Technical Compendium (″Red Books″)

This is a detailed and comprehensive description of OS/2 2.0 in five volumes, published by the IBM International Technical Support Center in Boca Raton, Florida (the same location as the development laboratory for OS/2 2.0). It is essential reading for anyone needing to understand OS/2 2.0 in depth. This guide has borrowed much material from the Red Books. They can be ordered together, as *OS/2 Technical Compendium* (GBOF-2254), or separately, as follows:

Volume 1: Control Program (GG24-3730-00)
Volume 2: DOS and Windows Environment (GG24-3731-00)
Volume 3: Presentation Manager (GG24-3732-00).
Volume 4: Application Development (GG24-3774-00)
Volume 5: Print Subsystem (GG24-3775-00)

Another important Red Book is *OS/2 Remote Installation and Maintenance* (GG24-3780-00). This contains practical instructions on how to install OS/2 across a LAN, discussing the various approaches. It will be supplemented, by the end of 1992, by two other books:

Automated CID Installation of OS/2 V2.0 (GG24-3783)
Automated CID Install of ES 1.0, NTS, LS (GG24-3781)

Other red books may be of interest. Here is a selection of some of the ones covering OS/2 and its extensions:

GG24-3875-00	LAN SERVER 2.0 NEW FUNCTIONS AND FEATURES
GG24-3890-00	NETWARE FROM IBM: NETWORK PROTOCOLS & STANDARDS
GG24-3794-00	EXTENDED SERVICES FOR OS/2 DATABASE MGR NEW FEATURES
GG24-3781-00	AUTOMATED INSTALL FOR CID ENABLED EXT SERVICES...
GG24-3580-01	DEVELOPING A CUA WORKPLACE APPLICATION
ZZ81-0295-00	EVALUATION OF OS/2 APPLICATION DEVELOPMENT TOOLS
GG24-3641-01	PRACTICAL INTRO TO OBJECT-ORIENTED PROGRAMMING
GG24-3822-00	MIGRATING FROM A DOS/WINDOWS ENVIRONMENT TO OS/2
GG24-3749-00	MULTIMEDIA APPLICATION ENABLERS & PS/2 ULTIMEDIA
GG24-3653-01	IBM PERSONAL SYSTEM/2 MULTIMEDIA FUNDAMENTALS

Copies can be ordered through IBM via the publication numbers in parentheses.

The OS/2 Developer (previously Personal Systems Developer)

This is a quarterly publication from IBM′s US Developer Assistance Program (DAP). Subscriptions can be made direct with the publishers in the US. Contact your local IBM office for details. Its publication number is G362-0001.

The OS/2 Developer contains many articles on a wide variety of OS/2 topics, and there have been many articles on OS/2 2.0, and on the system extensions like Database Manager and LAN Server, as well as material on OS/2 applications and development tools.

Personal Systems Technical Solutions

This publication comes from the Personal Systems Competency Center in Dallas, USA. It covers both hardware and software, and contains many useful articles on OS/2 2.0. It is published quarterly. The publication number is different for each issue, but all are of the type G325-50xx-00, where xx is a two digit number. Issues 5012, 5014, 5015, 5016 and 5017 all contain useful articles on OS/2 2.0

OS/2 White Papers

These papers were first issued by IBM in April 1991, and a second set appeared in January 1992. They are an excellent guide to IBM′s OS/2 strategy and future product directions. They include:

- *OS/2 2.0 Considerations*: an excellent guide to OS/2 2.0′s features, market positioning, application migration, and future directions (reprinted in the *Personal Systems Developer* Summer 1991 issue). This guide makes use of some of the material in this paper.
- *OS/2 LAN Server*: a summary of current LAN offerings and future directions.
- *OS/2 System Performance Management*: an overview of current OS/2 and LAN Server parameters and tuning facilities as well as future directions.
- *OS/2 System Management*: a good overview of current IBM SAA SystemView facilities in OS/2 environments and future directions.
- *OS/2 Database Manager Highlights and Directions*: an overview of current database offerings and future directions.
- *OS/2 Communications Manager Highlights and Direction*: a detailed overview of Communications Manager function.
- *OS/2 Performance Considerations*: a guide to some of the issues involved in tuning performance for OS/2 2.0
- *OS/2 2.0 Windows environment*: an explanation of how the WIN-OS/2 environment is architected, and what benefits the environment has versus Windows 3.x
- *OS/2 LAN Server Positioning*: a guide to the relative strengths and weaknesses of LAN Server versus Microsoft LAN Manager 2.0 and Novell NetWare 3.11
- *OS/2 LAN Server Migration*: how to move both clients and servers to OS/2 LAN Server 2.0 from various previous IBM LAN products.
- *OS/2: The Bigger Picture*: an overview of IBM′s OS/2 strategy and future directions
- *Upgrading to OS/2 2.0*: a review of the various options in upgrading from DOS, Windows or OS/2 1.3 to OS/2 2.0, and what is required for each migration path
- *Getting Started with the OS/2 Workplace Shell*: a brief introduction to the capabilities of the Workplace Shell. This is worth a read before starting on the shell for the first time
- *Client-Server Computing*: an overview of the various IBM products that address the client server environment, including Extended Services for OS/2 and OS/2 LAN Server
- *OS/2 2.0: the Development Platform of Choice*: reasons why OS/2 2.0 is an excellent target for developers, including a review of the range of various development tools available

The White Papers can be obtained from your IBM representative. They are on the IBM MKTTOOLS disk as WPAPERS and WPAPERS2 packages. They can be copied and distributed to anyone who wants to understand about OS/2.

IBM OS/2 Information

IBM has published a variety of brochures and information about OS/2. Here is a selection, with the publication numbers. These can be obtained from your dealer or IBM representative. Please note the following list is specific to those issued in Europe. Local publication numbers may vary, and some brochures may not be available in all countries. Contact your local IBM representative for a list relating to your country.

Table 11. OS/2 product information

IBM OS/2 V2.0 Product Information	G 511 1545-01
Extended Services for OS/2 Spec sheet	G 511 1546
OS/2 Local Area Network Server 2.0 Spec sheet	G 511 1554
OS/2 2.0 Decision Maker's Brochure	G 511 1548
The OS/2 partnership	G 511 1738
OS/2 - The Customer's Choice	G 511 1739
The New OS/2: Wouldn't it be great if...	G U20 2020

IBM OS/2 Applications Solutions Directory

This is a guide, published in the US, listing most of the applications available for OS/2 today. There are thousands to choose from! The main guide is orderable through IBM as G362-0002-02, and there is a supplement of 32-bit applications (G362-0029) which are either in development or already shipping.

CUA Vision materials

This is a video and a companion booklet outlining IBM's vision for the way user interfaces will develop. The booklet includes a demonstration diskette with a graphical demonstration of the principles discussed. Both the video and the diskette offer a fascinating insight into the principles behind the OS/2 Workplace Shell, which embodies the first stages of the vision, and into where the Workplace Shell may be headed in future.

The video is orderable as GV26-1004-00 (VHS PAL) or GV26-1003-00 (VHS NTSC) and the booklet as G242-0215-00.

OS/2 in the Corporate Environment

This is a short book published by Intelligent Environments Ltd, one of the leading vendors of OS/2 developement tools. It is an independent view of where OS/2 fits in the corporate computing world, including specific examples of where customers are implementing client-server solutions using OS/2 today. Contact Intelligent Environments for details. They are based at:

Intelligent Environments Europe Ltd.,
Crystal house
PO Box 51
Sunbury-on-Thames
Middlesex
TW16 7UL
United Kingdom

Tel: +44 (0)932 772266
Fax: +44 (0)932 771499

Intelligent Environments Inc.
2 Highwood Drive
Tewkesbury
MA 01876
USA

Tel: +1 508 640 1080
Fax: +1 508 640 1090

OS/2 Notebook

This is a book from Microsoft Press, edited by Dick Conklin, who is the editor of the *Personal Systems Developer*, containing the best of articles from that publication. It includes a wide range of articles on OS/2 2.0 and its extensions like Extended Services and OS/2 LAN Server. Its ISBN number is 1-55615-316-3.

The Design of OS/2

This book is co-written by Mike Kogan, one of the OS/2 2.0 System Architects, and H.M. Deitel. It is published by Addison-Wesley (ISBN 0-201-54889-5). The following paragraph is from the publisher′s abstract:

> The primary goal of *The Design Of OS/2* is to provide insights into the design decisions and philosophy of the OS/2 operating system. It discusses the motivation, architecture, and realization of OS/2 in the personal computing marketplace. The design of the major components of OS/2 are described in terms of their API architecture, internal data structures, and algorithms. Each area focuses on bridging operating systems theory to the realization of the design and implementation of OS/2. Where it is significant, an objective comparison of the technical aspects of OS/2 and other operating environments is provided. A key thrust is to describe the evolution of personal computer operating systems from DOS through 16-bit OS/2 and 32-bit OS/2.

Other OS/2 Books

There are a variety of books published on OS/2, including many new titles for OS/2 2.0. Here are a few of the titles:

- *Using OS/2* by Barry Nance and Greg Chicares, from Que books (ISBN 0-88022-863-6)
- *Now that I have OS/2 2.0 on my Desk - What do I do next?* by Stephen Levenson and Eli Hertz, published by Van Nostrand Reinhold (ISBN 0-442-01227-6)
- *Inside OS/2 2.0*, by Mark Minasi, John W. Little, Marlene Semple and Bill Camarda, from New Riders Publishing (ISBN 1-56205-045-1)
- *OS/2 2.0 Quick Reference*, by Barry Nance, from Que books (ISBN 1-56529-068-2)
- *Stepping Up To OS/2 2.0*, by Robert Albrecht and Michael Plura, from Abacus (ISBN 1-55755-160-X)
- *OS/2 2.0 Complete*, by Peter Franken, from Abacus (ISBN 1-55755-157-X)
- *Integrating Applications with OS/2 2.0*, by William Zach, from Van Nostrand Reinhold (ISBN 0-44201-234-9)
- *Client-Server Programming with OS/2 2.0* by Bob Orfali and Dan Harkey, from Van Nostrand Reinhold (ISBN 0 442-01219-5)
- *Writing OS/2 Device Drivers in C*, by Steve Mastrianni, from Van Nostrand Reinhold (ISBN 0 442-01141-5)
- *OS/2 Presentation Manager GPI*, by Graham Winn, from Van Nostrand Reinhold (ISBN 0-442-00468-0)
- *Learning to Program OS/2 2.0 PM by Example*, by Stephen A. Knight, from Van Nostrand Reinhold (ISBN 0-442-01292-6)
- *OS/2 Application Development Tools* by Brian Proffit, from Premier Publishing (ISBN 1-881899-00-4)
- *Converting Applications to OS/2*, by David Moskowitz et al, from Brady Press (ISBN 0-13-171943-2)
- *OS/2 2.0: The Usable, Portable Guide*, by John Haber and Herbert R. Haber, from Usable Portable Publications Inc., (ISBN 0-945965-27-4)

At the time of writing, four other books were about to made available. ISBN numbers were not obtainable, but IBM order numbers are supplied here:

G362-0014 THE OS/2 2.0 USER¢S GUIDE FOR THE WORKPLACE SHELL by Maria Tyne
from Computer Information Associates,

G362-0010 THE COBOL PRESENTATION MANAGER PROGRAMMING GUIDE by David M.Dill
Published by Van Nostrand Reinhold

G362-0012 COMPREHENSIVE PERFORMANCE FOR THE OS/2 2.0 DATABASE MANAGER by
Bruce Tate, Tim Malkemus, and Terry Gray (IBM, Austin, TX).
Published by Van Nostrand Reinhold

G362-0013 C PROGRAMMING IN THE OS/2 2.0 ENVIRONMENT by V. Mitra Gopaul.
Published by Van Nostrand Reinhold

The book by Barry Nance sold its first print run in a few weeks, and moved into the top ten computer titles. Even Steve Mastrianni′s book, which addresses a more specific audience, is already into its first reprint. In October, it was announced that the Barnes and Noble computer book bestseller list contained three of the above OS/2 books, for 17 consecutive weeks. The books were the ones by Orfali and Harkey, Minasi et al., and Levenson and Hertz. Sales of books are often regarded as one of the signs of a flourishing platform (witness the sales in books about Lotus 1-2-3, WordPerfect, and Windows). This indicator also shows the amount of interest there is now in OS/2.

In fact, OS/2 books are not only appearing in English. An indicator of the success of OS/2 is that books in national languages are appearing, a testimony not only to the strength of OS/2 in the countries involved, but also of its widespread appeal (books in English would usually be considered to reach a wider market, especially among computer experts, but OS/2′s appeal to end users means a viable market at which to target books in the native language.) Here is a list of some OS/2 books written and published in Germany:

†OS/2 2.0, Grundlagen und Praxis†
von Hans Fremuth (mit Vorwort von H. Kahl) im tewi Verlag
(in deutscher Sprache)
ISBN 3-89362-212-8

†OS/2 2.0, Das Kompendium†
von Olaf Koch, Norbert Meder und Peter Scheuber (mit Beitrugen
von Whittle und Pignatelli) im Verlag Markt & Technik
(in deutscher Sprache)
ISBN 3-87791-302-2

†OS/2 - 2.0, Der Data Becker Fuhrer†
im Verlag DATA BECKER
(in deutscher Sprache)
ISBN 3-89011-637-X

†OS/2 Version 2 deutsch, Das Data Becker Handbuch†
von Peter Franken im Verlag DATA BECKER
(in deutscher Sprache)
ISBN 3-89011-505-5

†Von DOS und Windows nach OS/2 2.0, Tips & Tricks†
von Robert M. Albrecht und Michael Plura im Verlag DATA BECKER
(in deutscher Sprache)
ISBN 3-89011-543-8

†Schnell Anleitung OS/2 Version 2 deutsch†
im Verlag DATA BECKER
(in deutscher Sprache)
ISBN 3-89011-658-2

†QuickStart OS/2 2.0, Der Einsteig in 20 Schritten†
von Harald Babiel im SYBEX Verlag
(in deutscher Sprache)
ISBN 3-88745-859-1

OS/2 Monthly

This is an independent magazine dedicated to news, views and technical information about OS/2. It is published by:

JDS Publishing
PO Box 4351
Highland Park
NJ 08904
USA

Tel: +1-908-247-0952

Moving to the Workplace Shell video

This video (part number 41G5097) guides the new user of OS/2 2.0 through the Workplace Shell. It is meant to be a light and enjoyable introduction to the capabilities of the shell, to encourage you to explore for yourself.

OS/2 Frequently Asked Questions (FAQ)

This document, maintained by Timothy F. Sipples (sip1@ellis.uchicago.edu), is an excellent basic reference to many common questions and answers about OS/2. It is available in electronic form on most of the bulletin board systems mentioned in Appendix D, “OS/2 Bulletin Board Systems” on page 139. It is just one example of many of the excellent reference materials and tools made available by OS/2 enthusiasts around the world, on bulletin board systems.

OS/2 2.0 Information and Planning Guide

This publication (G326-0160-00) gives an overview of OS/2 2.0 function, as well as assistance in estimating memory and disk space requirements. It is recommended for anyone planning to install OS/2 2.0 on a number of machines. Guides are also available for Extended Services for OS/2 (G326-0161-00) and OS/2 LAN Server (G326-0162-00).

Appendix D. OS/2 Bulletin Board Systems

Bulletin board systems (BBS) are often a good way of getting information about OS/2, whether technical support, news or useful utilities. In many countries, IBM runs a BBS as a means of customer support; but there are also many other BBSs run by OS/2 enthusiasts. Here is a list of some of the BBSs:

IBM Bulletin Boards

Not all countries operate a BBS. Here is a list of some of the countries that do, with the connection details:

```
IBM Sweden

   Phone: 46+8-7932200 Dealers       46+8-7934222 Customers
   Lines: 10 Lines
   Speed: 1200-14400 Connection rate (V.32 and MNP)
          10 USR DS 14400 modems

********************************************************

IBM Austria

IBM Bulletin Board System Austria:

   Phone: 043-222-21145-6600
   Speed: 9600
   Lines: 2

********************************************************

IBM Germany

IBM Bulletin Board System Germany

   Fidonet : 2:241/7411.0@FIDONET
   Phone   : 049-711-785-7777

********************************************************

IBM Canada

Canadian Sofware Support Center, specializing in OS/2 and OS/2 support.

 Vancouver - 604-664-6464            Montreal  - 514-938-3122
 Toronto   - 416-946-4244 or 4255

********************************************************

IBM Switzerland

Name:  IBM-BBS HITLINE Communications
Phone: 0041 56 321 800 (16 lines/14000 Baud)
ISDN:  0041 56 320 589 (1 line/128kBits/s)

*********************************************************
```

```
IBM Denmark

  Name: IBM OS/2 BBS
  Phone: + 45 42 88 72 22
  Lines: 3 lines
  Speed: 9600 (V32 and MNP)
  Location: Lyngby, Denmark

********************************************************

IBM Belgium

International BBS name :    The IBM Belgium BBS
In country BBS name    :    End-User Node.
Location: IBM Belgium s.a.  J.F.Kennedylaan 2, B-1831 Diegem.
Modem number :              32-2-725.60.10
                            8 lines, up to 9600 bps, 8,n,1, V32  24/24
                            No MNP4 no MNP4.

********************************************************

IBM USA

IBM National Support Center BBS, located in Atlanta GA.

1-404-835-6600 =>  First available modem
1-404-835-6296 =>  First available Hayes Ultra
1-404-835-5300 =>  First available USR V.32bis with ASL
1-404-835-5578 =>  First available IBM 7855 model 10

********************************************************

IBM Finland

IBM OS/2 Bulletin Board System

      Phone: +358-0-480 422
      Speed: 9600, 8, n, 1

********************************************************

IBM Spain

Bulletin Board System IBM OS/2

 Phone: 34 1 397 55 80      34 1 397 58 73
        34 1 397 55 81      34 1 397 59 63
 Speed: 9600, 8, n, 1

********************************************************

IBM UK

Phone: +44-(0)256-336655 (>12000 - V.32 MNP-5)
Lines: 15

********************************************************
```

```
IBM Israel

Phone: 049-711-785-7777
Additional Info: 2:241/7411.0@FIDONET

********************************************************


IBM Netherlands

Phone: +31 (0)20-6974757
Additional Info: Lines 8 Courier Dual Standard 14K4

********************************************************


IBM Norway

Phone: 47-2-999450

********************************************************


ISM South Africa

Phone: 27.11.224.2000
Info - Lines: 47 55 81
       Speed: 9600,8,n,1 (V32 and MNP)

********************************************************
```

Other BBSs

The following information was provided by the US OS/2 Support Line:

```
O S / 2   B B S ¢ s   A C R O S S   T H E   W O R L D
-----------------------------------------------------

This list is a compilation of OS/2 BBS¢s across the world.  If you wish to
make an addition or correction to this list, please send the information to
the following (as netmail or logged onto the BBS itself):

BBS      : LiveNet, 1:170/110@fidonet, (918) 481-5715
Location : Tulsa, OK, USA
Sysop    : Dave Fisher

This list is distributed to many FidoNet nodes found in this OS/2 BBS listing
via the Fernwood distribution system.  All BBS¢s listed are in alphabetical
order by country, and then by BBS name.  Unless otherwise noted, all node
addresses are FidoNet.

A current list can always be file-requested from LiveNet as ¢OS2WORLD¢.

                                 Enjoy!
```

```
Last Update: May 14, 1992

Legend:  * : OS/2 is primary interest of board
         F : Board is a FidoNet node
         % : Entry is new or changed as of last list
        -A : HST, MNP modem
        -B : HST, MNP V.32 (and/or V.42) modem
        -C : HST, MNP V.32bis/V.42bis modem
        -D : MNP V.32/V.42bis modem
        -E : MNP V.32 modem

USA BBS¢s show states, International BBS¢s show three letter country codes.

-----------------------------------------------------------------------------

Graham Stair         3M Australia           +61-2-498-9184   Aus 9600-E * F
Ian Watson           OZ-Share OS/2 BBS      +61-7-398-3759   Aus 9600-E * F
Alan Salmon          PC User¢s Group        +61-6-259-1244   Aus 2400
Felix Tsang          Programmer¢s BBS       +61-2-875-1296   Aus 9600-E * F
Bill Bolton          Software Tools Mail Exc +61-2-449-2618  Aus 9600-E * F
                                            +61-2-449-9477   Aus 9600-E * F
John Della-Torre     The Poet¢s Dilemma     +61-2-804-6412   Aus 2400-E
Norbert Fuerst       The Styrian OS/2 Jumbo  +43-316-673237  Aus 9600-A * F
Danny Bruggeman      Hellfire               +32-2-7515203    Bel 9600-D * F
Bas Heijermans       Moving Sound OS/2 BBS  +32-3-3850748    Bel 9600-D * F
Benoit HUON          Os/2 MANiA BELGIUM     +32-2-3872021    Bel 9600-D * F
Tony Bearman         Bear Garden            (604) 533-1867   Can 9600-C * F
Chris Ange-Schultz   Home Front BBS         (514) 769-5174   Can 2400   * F
Peter Fitzsimmons    RT Labs                (416) 867-9663   Can 9600-B * F
Gerry Rozema         The Idle Task          (604) 273-5588   Can 14.4-B * F
Jerry Stevens        The Locutory           (613) 722-0489   Can 9600-D * F
Alec Herrmann        The Nibble¢s Roost     (604) 244-8009   Can 14.4-B * F
Kevin Lowey          University of Saskatche (306) 966-4857  Can 14.4-C * F
Jorgen Ollgaard      Josti-BBS              +45-47-380120    Den 9600-C * F
                                            +45-47-380524    Den 9600-C * F
Rene Carlsen         OS/2 Task & FrontDoor H +45-98451070    Den 9600-A * F
Emmanuel Sandorfi    Os/2 MANiA (Help Maximu +33-164-090460  Fra 9600-C * F
Romeo Bernreuther    CCWN-BOX               +49-7151-68434   Ger 14.4-B * F
Peter Plischka       IBM Mailbox            +49-201-210744   Ger 9600-C * F
                                            +49-201-295181   Ger 9600-B * F
Juergen Berger       JERRY¢S OS/2-BBS       +49-6134-26563   Ger 2400-A * F %
Oliver Lass          LRZ-System             +49-228-331214   Ger 2400   *
                                            +49-228-334372   Ger 9600-D *
Oliver Schwabedissen  MoonFlower            +49-6145-31602   Ger 9600-C * F
Richard Clement      OS/2 Express           +49-6183-74270   Ger 9600-D * F
Harald Kipp          OS/2 Point             +49-234-9279222  Ger 9600-B * F
Michael Breukel      PC Softbox OS/2        +49-6196-27799   Ger 2400   * F
Markus Noller        Second Source          +49-7191-56267   Ger 2400   * F
Kalle Braun          Terrania City          +49-228-317752   Ger 14.4-B   F
Thomas Tegel         The CAT                +49-7971-72446   Ger 14.4-D * F %
Karlheinz Kissel     The_File_Store         +49-6106-22266   Ger 9600-C * F
Chris Leuder         Zaphod BBS             +49-228-262894   Ger 14.4-B   F
                                            +49-228-229147   Ger 2400     F
Joop Mellaart        INFOBOARD              +31-4752-6200    Hol 2400   * F
Marcel Stikkelman    PC-Square              +31-79-424107    Hol 14.4-C * F
Pasquale Cantiello   FastForward BBS        +39-823-812099   Ita 14.4-C * F
Luigi Ravina         Italy Network          +39-11-8180069   Ita 9600-A * F
Roberto Sonzogni     Runnin¢ with The Devil  +39-363-302798  Ita 9600-C * F
```

Dave Jones	The TJD Support BBS	+31-1720-38558	Net	9600-A	*		
Terje Slydahl	PerlePorten	+47-83-33003	Nor	9600-C	*	F	
		+47-83-33003	Nor	2400	*	F	
Alex Wyss	Gepard¢s Oracle Zuerich	+41-1-3637037	Swi	14.4-C	*	F	
Michael Buenter	MICS OS/2 Paradise	+41-41-538607	Swi	9600-C	*	F	
Ernesto Hagmann	PC-Info	+41-61-9412204	Swi	9600-C	*	F	
Mike Gove	MonuSci CBCS	+44-454-633197	UK	9600-C	*	F	
Phil Tuck	The TJD Support BBS	+44-535-665345	UK	9600-C	*		
		+44-535-665345	UK	9600-A	*		
Patrick O¢Riva	AsmLang and OS/2	(408) 259-2223	CA	14.4-B	*	F	
Steve Lesner	Bullet BBS	(203) 329-2972	CT	9600-B	*	F	%
		(203) 322-4135	CT	9600-B	*	F	%
Jim Dailey	Cajon Zone OS/2	(619) 588-6634	CA	9600-D	*	F	
Bob Germer	Capital City BBS	(609) 386-1989	NJ	14.4-C	*	F	
Dennis Conley	Communitel OS/2 BBS	(702) 399-0486	NV	14.4-C	*	F	
Emmitt Dove	Fernwood	(203) 483-0348	CT	9600-C	*	F	
		(203) 481-7934	CT	14.4-B	*	F	
Bill Cook	GREATER CHICAGO Online!	(708) 895-4042	IL	9600	*	F	
Bogie Bugsalewicz	I CAN! BBS	(312) 736-7434	IL	9600-C		F	
		(312) 736-7388	IL	2400		F	
n/a	IBM National Support Ce	(404) 835-6600	GA	2400			
		(404) 835-5300	GA	9600-C			
Ed June	Information Overload	(404) 471-1549	GA	9600-A	*	F	
James Chance	Lee¢s Lounge	(410) 721-9452	MD	14.4-B	*	F	
Robert McA	Live-Wire	(214) 307-8119	TX	9600-B		F	
Dave Fisher	LiveNet	(918) 481-5715	OK	16.8-C	*	F	
Chuck Gilmore	Magnum BBS	(805) 582-9306	CA	9600-C	*		
Joe Salemi	Max¢s Doghouse	(703) 548-7849	VA	2400-A	*	F	
Paul Breedlove	Multi-Net	(503) 883-8197	OR	9600-C	*		
Ron Bemis	Nibbles & Bytes	(214) 231-3841	TX	9600-A		F	
Craig Swanson	OS/2 Connection	(619) 558-9475	CA	14.4-D	*	F	
Pete Norloff	OS/2 Shareware	(703) 385-4325	VA	9600-C	*	F	
		(703) 385-0931	VA	9600-C	*	F	
Brady Flowers	Oberon Software	(507) 388-1154	MN	14.4-C	*		
Unknown	Omega-Point BBS	(714) 963-8517	CA	2400			
Paul Beverly	PMSC OnLine Resource	(803) 735-6101	SC	2400	*	F	
Louis F. Ursini	Quantum Leap	(215) 967-9018	PA	2400			
Ken Rucker	RucK¢s Place/2	(817) 485-8042	TX	14.4-C	*	F	
Randy Edwards	Socialism OnLine!	(719) 392-7781	CO	9600-B	*	F	
Ed Barboni	System-2 RBBS	(215) 631-0685	PA	9600-D		F	
		(215) 584-1413	PA	9600-D		F	
Mark Lehrer	The Akron Anomoly	(216) 688-6383	OH	9600-C		F	
Bill Schnell	The Asylum BBS	(918) 832-1462	OK	9600-B	*	F	
Felix Tang	The Excelsior	(203) 466-1826	CT	14.4-C	*	F	
		(203) 466-1892	CT	2400	*	F	
Bob Hatton	The Monster BBS	(908) 382-5671	NJ	9600-A	*		
Woody Sturges	The OS/2 Woodmeister	(314) 446-0016	MO	14.4-C	*	F	
Troy Kraser	The Other World	(904) 893-2404	FL	9600-D		F	
Mark Wheeler	The SandDollar	(407) 784-4507	FL	9600-A	*	F	
Art Fellner	The Soldier¢s Bored	(713) 437-2859	TX	9600-C	*	F	
Bill Andrus	The Systems Exchange	(703) 323-7654	VA	9600-A	*	F	
Unknown	WSI BBS	(901) 386-4712	TN	2400			

Printed in U.S.A.

```
+++ DOCNUM tag required with COVER tag. (Page 144  File: OS2TECH SCRIPT)
′.EDF#CV′ WAS IMBEDDED AT LINE 210 OF ′.EDF#BCV7′
′.EDF#BCV7′ WAS IMBEDDED AT LINE 460 OF ′.EDF#END′
′.EDF#END′ WAS IMBEDDED AT LINE 189 OF ′EDFPRF30′
STARTING PASS 2 OF 2.
+++ DOCNUM tag required with COVER tag. (Page 144  File: OS2TECH SCRIPT)
′.EDF#CV′ WAS IMBEDDED AT LINE 210 OF ′.EDF#BCV7′
′.EDF#BCV7′ WAS IMBEDDED AT LINE 460 OF ′.EDF#END′
′.EDF#END′ WAS IMBEDDED AT LINE 189 OF ′EDFPRF30′
```