IBM

Operating System/2™
Standard Edition
Version 1.2

Using Advanced Features

Programming Family

First Edition (September 1989)

About This Book

Using Advanced Features introduces you to many OS/2® functions and includes sections on configuring your system and maximizing system performance to get the most out of using IBM Operating System/2™ Version 1.2 (referred to as the OS/2 program in the remainder of this book). Topics also describe the concepts behind OS/2, its functions, and how they interrelate. Practical tips and techniques are included in these topics to help you use the features of OS/2 more easily and productively.

Who Should Read This Book

This book is primarily written for experienced operating system users and is intended for use with the online *OS/2 Command Reference*. It is helpful if you are familiar with Presentation Manager™ and the information outlined in *Getting Started*.

How This Book Is Organized

Information in this book is organized in the following manner:

Chapter 1 familiarizes you with the basic features of the OS/2 program, including its multitasking capabilities.

Chapter 2 provides information to help customize your system to meet your individual needs.

Chapter 3 contains material to assist in maximizing system performance through process and memory management.

Chapter 4 explains OS/2 command symbols and gives instructions for redirecting input and output on your system.

Chapter 5 furnishes information to aid in diagnosing system problems.

Chapter 6 compares the differences between IBM DOS and DOS mode of OS/2.

Chapter 7 supplies data on creating and running batch files.

Chapter 8 provides information on editing and creating files with the System Editor.

Appendix A contains key assignments for the System Editor.

Appendix B contains explanations to messages that may be displayed in DOS mode.

Appendix C includes an alphabetic listing of OS/2 commands.

Contents

Chapter 1. About This Operating System

Operating System/2™ (OS/2®) is a program that controls the workings of your computing system. It provides easy access to programs and information by allowing you to state your computer requirements in terms that are easily understood. This lets you complete your work quickly without having to view the behind-the-scenes programming code that makes your system function.

You can compare features of OS/2 to desktop materials you would normally use in an office environment.

Figure 1. Operating System/2 - the Desktop Manager

For example, multiple windows on your screen are similar to pieces of paper on a desktop. Like papers, you can stack windows, blocking other windows from view. You can also cascade (overlap) windows so that you can partially see the windows underneath.

Operating System/2 is a trademark of the International Business Machines Corporation.

OS/2 is a registered trademark of the International Business Machines Corporation.

Because OS/2 allows you to view several windows at one time, you don't need to remember or write down information available from different sources. You can install desktop programs that make a calendar, notepad, or telephone book just a touch away. Also, if you normally manage and maintain documents in a file cabinet, you will feel comfortable using the File Manager. Let the File Manager help you organize and control information. You can think of directory icons (⌷) as folders inside a file cabinet, and file icons (⌷) as papers inside the folders.

The functions of the OS/2 program that you are probably most familiar with are file-related activities such as creating, copying, or erasing files. In addition to these tasks, OS/2 provides other services. These services include allocating resources, scheduling, protecting programs, and providing a suitable computing environment in which to run your programs.

This book includes OS/2 topics such as multitasking capabilities, memory management, and DOS compatibility. Also introduced are several distinct features of the OS/2 program. One of these is the System Editor, a program which provides text editing functions for creating, editing, and managing files. Also explained are beneficial ways to get the most from the OS/2 program by, for example, customizing your system to maximize performance.

Operating Modes

The OS/2 program provides the operating environments of OS/2 mode and DOS mode. Although each mode affects the processing of programs and commands differently, most commands work in both modes. There are, however, some commands that provide functions specific to only one mode. Refer to the "Table of OS/2 Commands" on page 167 for a list of OS/2 commands.

OS/2 mode

Commonly called *protect mode*, this multiprogramming environment allows you to run several OS/2 programs at the same time. This gives each program the potential of taking advantage of available physical memory up to 16 megabytes (MB). OS/2 mode allows you to achieve a high level of interaction with an OS/2 program. It permits you to run some programs in interactive windows on your screen. At the same time, you can run other programs in the background, away from view. Your operating system also protects each program so that another program running at the same time cannot change it.

DOS mode

When using the OS/2 program, you may choose to use DOS mode, an environment similar to IBM Disk Operating System Version 4.0. Unlike most programs in OS/2 mode, a program running in DOS mode does not run in a window. Instead, it takes over the entire screen and runs only in the foreground session. You can, however, run a program in DOS mode at the same time with several active OS/2 programs.

Commonly called *real mode*, this mode allows you to continue using DOS programs with your new operating system. You can start DOS tasks or begin a DOS program almost as though you were using DOS 4.0. However, not all programs originally written for a personal computer work properly in DOS mode (for example, timing- and network-dependent programs). Refer to "DOS Mode Compatibility" on page 97 for program limitations and answers to compatibility questions.

Working with Input/Output (I/O)

Work begins when you enter *input*, or information, into your computer system for processing. You do this by performing operations such as typing on your keyboard, clicking a mouse button, or gathering data from a disk. Work ends with *output*, the results of an operation. Examples of output operations are displaying results on your screen, storing information on a disk, or printing information on a printer.

Figure 2. Input/Output (I/O) Devices

Depending on your type of system, the central processing unit (CPU) processes I/O operations differently. A program in a single-tasking system, such as IBM DOS, can perform only one I/O operation at a time. The program working on an I/O request waits for the CPU to return the result from an operation before the CPU can process other information. In a multi-tasking system such as OS/2, however, many programs can request I/O operations at the same time. OS/2 queues these multiple requests and processes them each in turn.

Multitasking with OS/2

The multitasking features of OS/2 allow you to process many tasks (basic units of work) at the same time. OS/2 divides processor time between multiple tasks. This gives the impression that your programs, jobs, or files are running at the same time. By sharing the system's resources among tasks, multitasking increases productivity by reducing the time required to switch between programs. It allows processing to take place in the background while you are working on another task in the foreground.

The simplest example of multitasking is working with sessions, running several tasks at the same time. For example, while you are doing word processing in the foreground, recalculation of a spreadsheet could be processing in the background. Moreover, OS/2 protects the information contained in each running program from being changed by another program running at the same time.
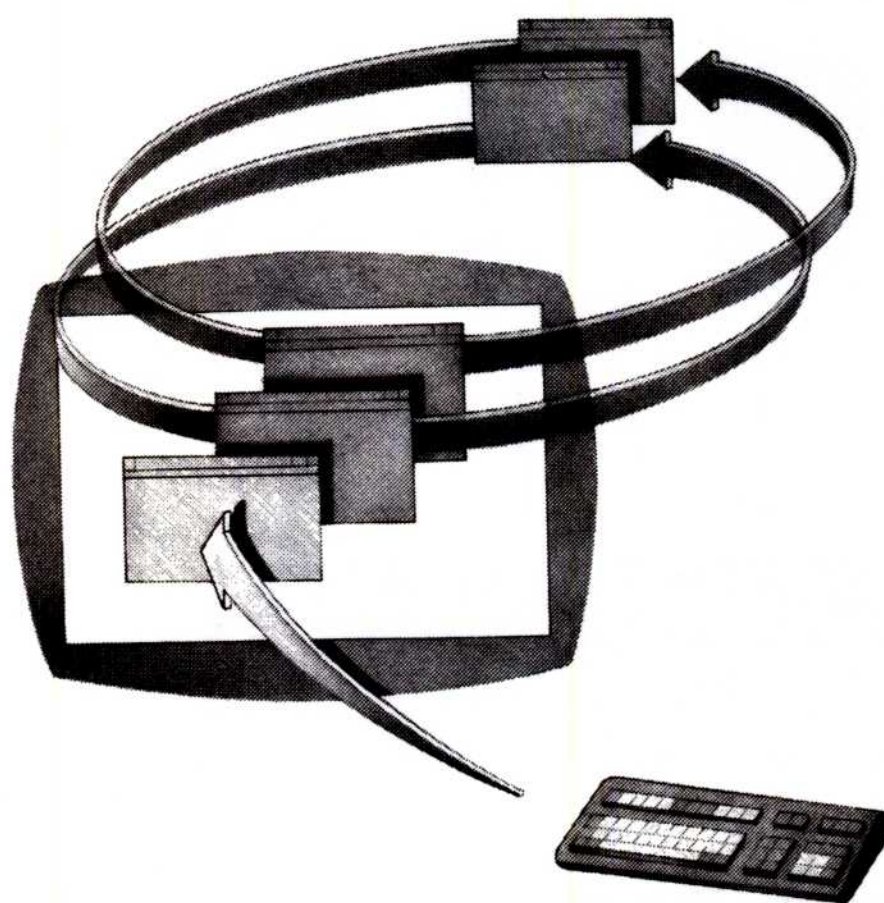
Figure 3. Multitasking. Allows tasks to process in background windows while you are working on another task in the foreground.

Although many programs appear to be processing at the same time when you view windows on your screen, the CPU carries out only one task at a time. The processor sometimes waits for input or output from a slower device (such as the keyboard or disk). For this reason, many program developers today design programs to create or control one or more tasks. For more complex processing requirements, developers can design a program so that its functions are divided among a collection of cooperating processes and threads, as shown:
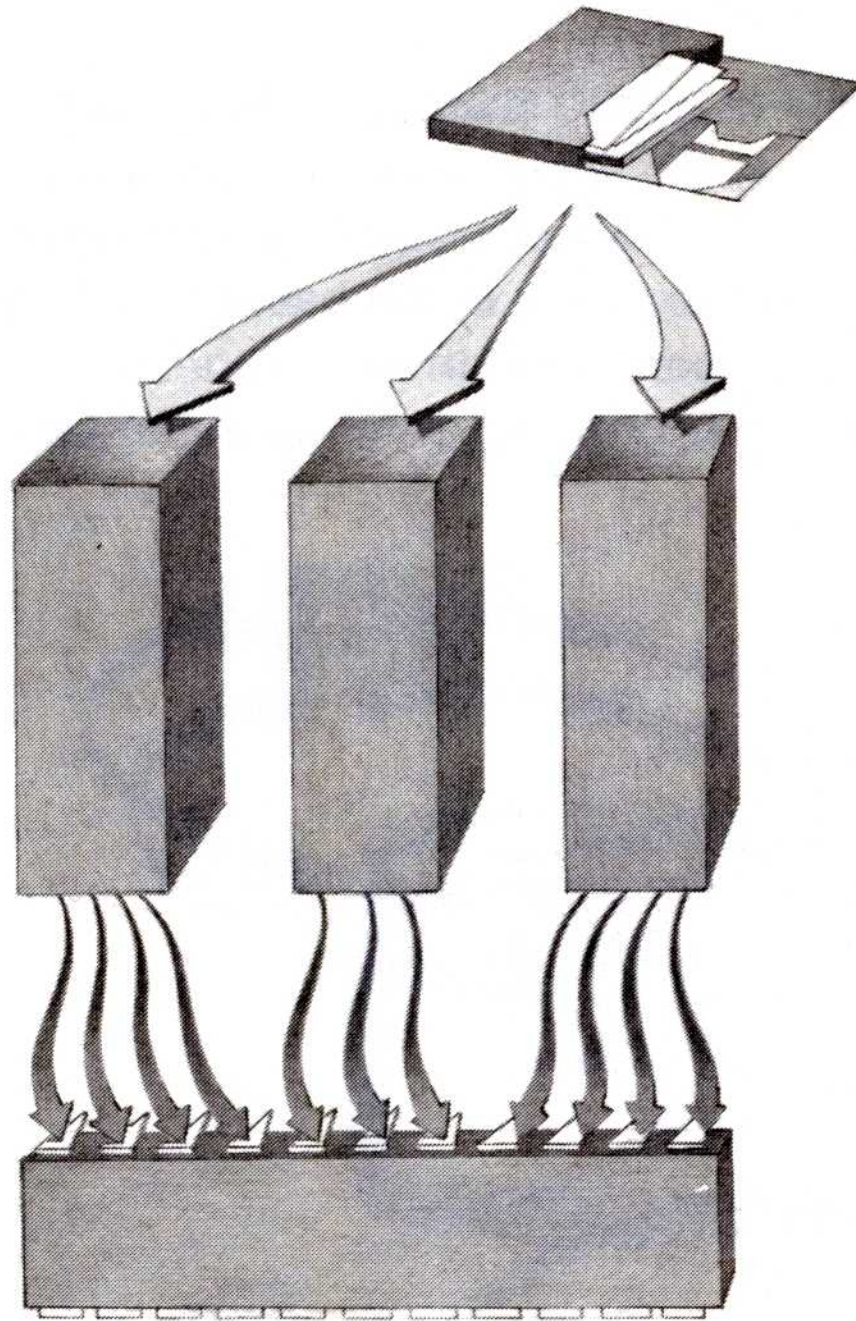
Figure 4. Processes and Threads. A program can be designed as separate processes, each consisting of many threads waiting to be processed by the CPU.

Note: For information on processes and threads, refer to "Processes and Threads" on page 10.

For example, suppose a programmer was developing a copy
program. He might decide to read data from one drive with one
thread, while at the same time, write to the other drive with another
thread. This maximizes system performance because programs use
the processor's time more productively.

Windows and Sessions

Each task or program you start runs in its own window, giving the
appearance that each task is running in a separate, protected per-
sonal computer. Windows, called *sessions*, are protected environ-
ments that allow many programs to coexist within the system, and not
interfere with one another. Because each OS/2 session runs inde-
pendently, changes you make in one session do not affect other ses-
sions. For example, setting a new path for an OS/2 session affects
only that OS/2 session; it does not change paths to other sessions.

The following screen represents two cascading windows:
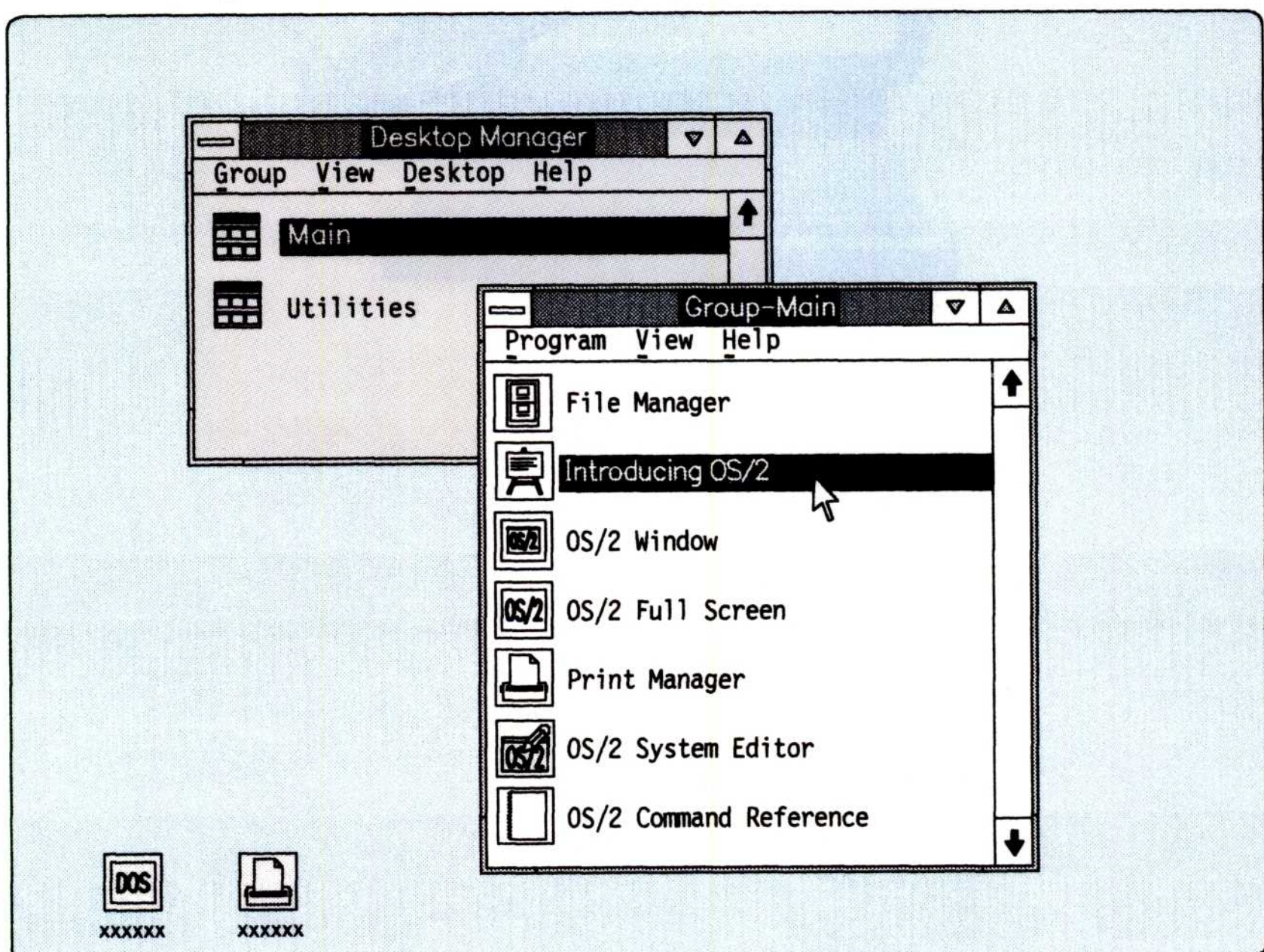
A session can be one of two types:

Foreground Session
You can work directly with this interactive session. It receives input
from your keyboard and mouse, or displays information on your
screen. In addition, the CPU gives the foreground session a higher
priority than a background session. This allows a program in the
foreground session to run faster, improving performance for this
session.
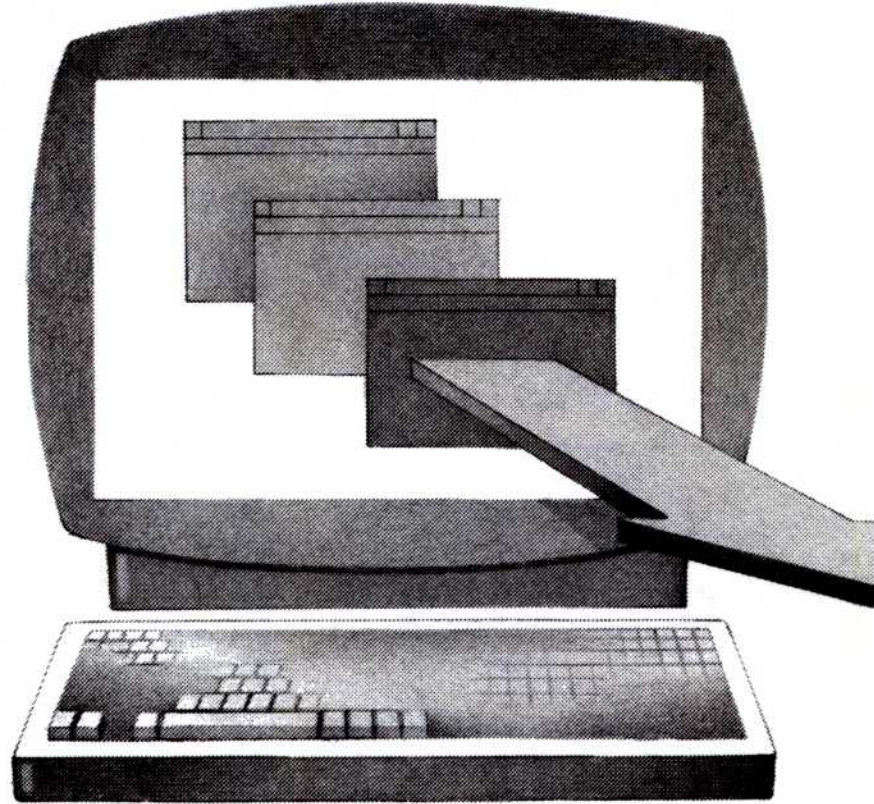
Background Sessions

You cannot work directly with this session. Instead, this session continues to run in the background, away from your view. Within windows, background sessions share the display screen with the foreground session. You cannot, however, interact with a background session (such as typing on the keyboard) until you bring the job to the foreground session.
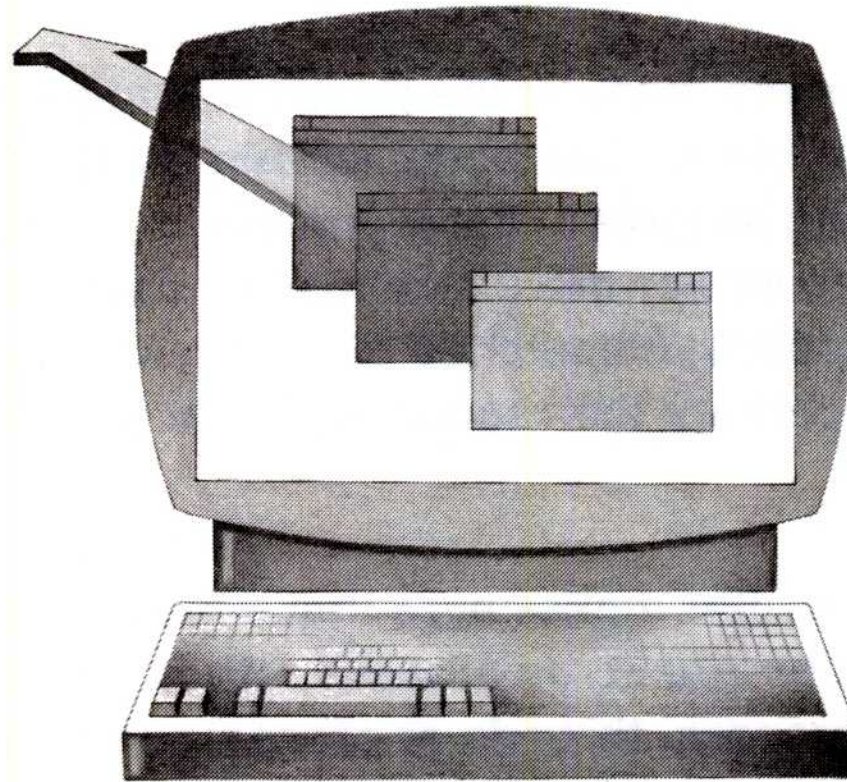
Programs can run unattended in background sessions, receiving processor time as it becomes available. However, if a program in a background session is awaiting further instruction, it stops running and stays suspended until you either end the session or bring it to the foreground.

When running a program in DOS mode, OS/2 suspends the program when you switch it to the background. This is why DOS mode does not support timing-dependent programs such as communications and real-time programs. These programs have defined time constraints in which the CPU completes processing.

Processes and Threads

Many tasks, such as those involving programs or files, can occupy various windows of your screen at the same time. Besides viewing and running these tasks at the same time, programs can also simultaneously reside in memory, share memory, and alternate the use of the CPU among them. Called *multiprogramming*, a scheduler manages this process and improves the overall efficiency of the system by coordinating priority-based jobs going to the CPU.

One *process* manages many of the programs that you run. This process is responsible for the mode, screen contents, and resources used by the running program. Each process, or instance of program operation, contains one or more *threads*, the most elementary units in processing. Threads, dispatched on a priority basis, use processor running cycles and share the resources of the process that created them.

A time-slicing scheduler makes sure that threads of equal priority receive an equal opportunity to process. Because threads use the resources of the process, threads or processes running at the same time must communicate with one another. Communication is necessary to coordinate both their access to shared resources and their order of running.

Although one process manages most programs, program developers can design a more complex program as separate processes, or multiple threads, within a single process. In this way, a complex program can send more than one thread to use processor cycles, thereby speeding up its performance. Then, when many programs are running and occupying memory at the same time, the CPU can switch between these processes. This makes your system more productive.
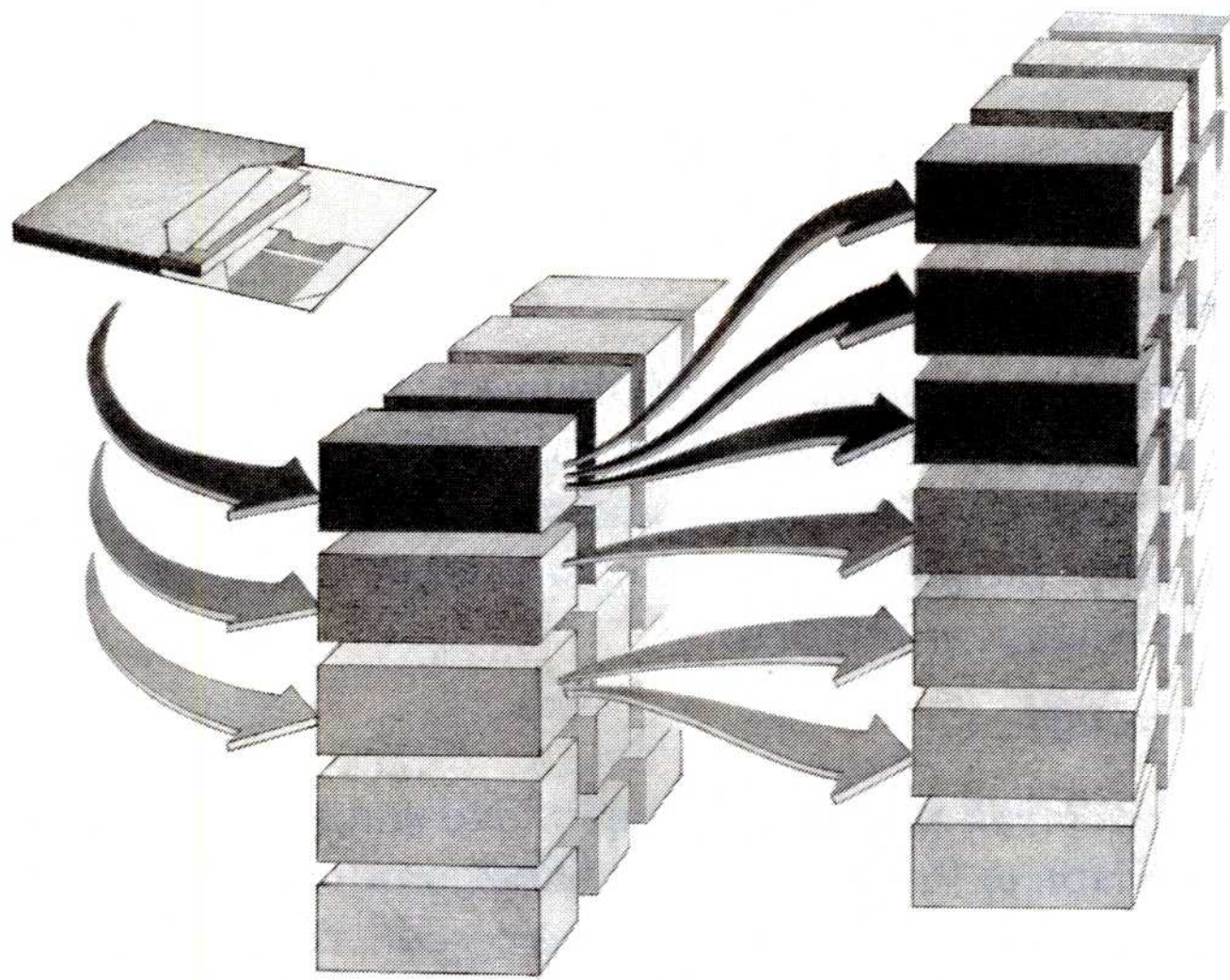
Figure 5. Processes and Threads.   A program can be designed as separate processes or multiple threads within a single process.

Note: For further information on processes and threads, refer to "Process Management" on page 64.

Running OS/2 Programs

The three types of OS/2 programs that you can run in sessions are as follows:

Presentation Manager™

Presentation Manager programs use full windowing capabilities. These programs create one or more windows that make the function of your program visible and easy to access. You can monitor what other programs are doing and easily switch between programs whenever required.

Text-windowed

Text-windowed programs have limited windowing capabilities in comparison to Presentation Manager programs. Instead of creating one or more windows on the screen, OS/2 supplies this type of program with a standard window that you can move and size.

An example is the OS/2 windowed command prompt. If you choose to run an OS/2 command prompt in a text window, other program windows remain visible on the screen. This environment is well suited to text-based programs that can operate in the background, such as compilers or editors. You can then interact with another program in the foreground session.

Full-screen

Full-screen programs fill the entire screen. Although other programs may be currently running, the full-screen program does not share the display screen. You can access other running programs by repeatedly pressing the Alt and Esc keys. You can also switch to the Task List window to select a program by pressing the Ctrl and Esc keys. It is possible for you to run several full-screen programs at the same time you might be running other programs in windows.

Presentation Manager is a trademark of the International Business Machines Corporation.

You can start and run several OS/2 programs at the same time by selecting:

- a program title listed in the Group-Main or Group-Utilities window,

- a program file from the File Manager window, or

- an OS/2 full-screen or window command prompt from the Group-Main window, as shown:
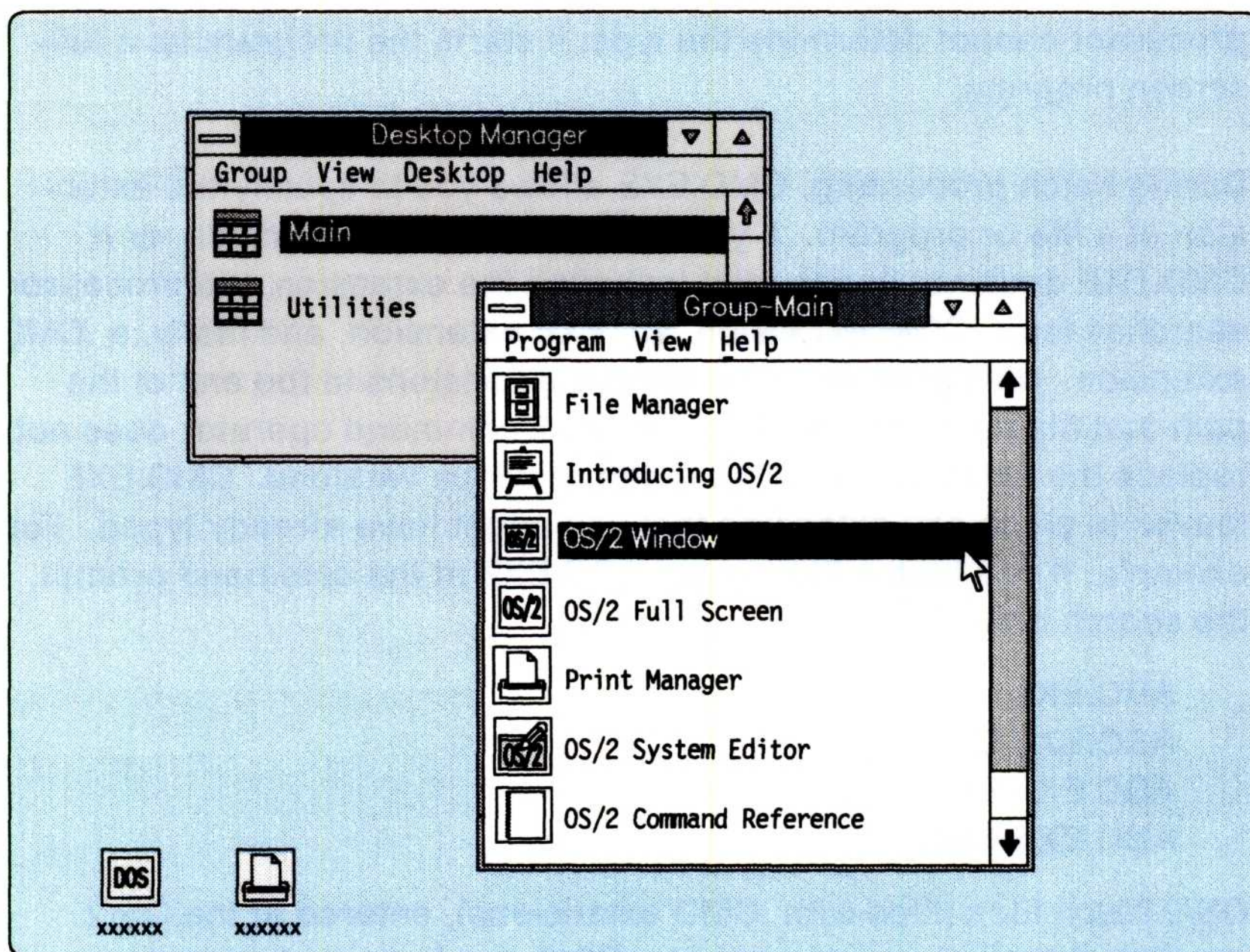
Note: If you want to return to the Group-Main window, type EXIT at the command prompt and press the Enter key.

An OS/2 program continues to run even if it is not visible. You can start as many as 16 Presentation Manager programs, 16 text-windowed programs, and 12 full-screen programs at the same time. However, the actual number of sessions that you can start at the same time depends on the amount of memory available in your system.

OS/2 Command Processing

The OS/2 mode command processor, CMD.EXE, is a program that interprets and runs commands. It reads the type of program you enter into the system and tries to determine what action to take. The information that came with your program should describe its program type. If it does not and you choose the option of letting the program decide, the OS/2 command processor attempts to run the program in the session for which the program was designed. If the program information does not contain the program type, and the command processor cannot determine the type, it starts the program as a full-screen program.

During batch processing, CMD.EXE allows you to specify the extension of a file or program. If you do not specify an extension, or if CMD.EXE does not find the file including the extension, the processor searches for a .COM extension, an .EXE extension, and lastly, a .CMD extension. CMD.EXE adds the default extensions to the end of the path and file name you typed. The OS/2 command operator does not replace the extension as it did in the previous versions. CMD.EXE leaves in place any extension that you might have already typed. For example, if you type a file named ABC.EXE at the command prompt, the search order is:

    ABC.EXE
    ABC.EXE.COM
    ABC.EXE.EXE
    ABC.EXE.CMD

OS/2 batch files (files with .CMD extensions), entered at the OS/2 command prompt or processed by OS/2, run in text windows by default. However, if you use the following START command parameters:

    /WIN (text-windowed)
    /FS (full-screen)
    /PM (Presentation Manager)

you can force a program to run in a certain type of session. Make sure that your program is capable of running in the session you choose. You can also use the START command in a special batch file named STARTUP.CMD to begin programs in additional OS/2 sessions. This file begins a process automatically when you start OS/2. Refer to page 111 for further information on STARTUP.CMD.

Running DOS Programs

OS/2 provides DOS mode so that you can run a single DOS program in the foreground session while running OS/2 programs in the background. OS/2, however, limits the DOS programs you want processed to 640KB. They can run only within the first megabyte of system storage.

The following are ways to enter DOS mode from Presentation Manager:

- Press the Alt and Esc keys enough times, depending on how many programs are running.

- From the Task List window, double-click on **DOS**, or select it and press the Enter key.

- Double-click on the DOS mode icon, as shown:

Note: Once your program completes, press the Ctrl and Esc keys to display the Task List window.

You can start only one DOS prompt to enter a command or begin a DOS program. This is because only one DOS command processor can process a command or program at a time. Also, because a DOS program requires the entire screen in the foreground session, it cannot take advantage of OS/2 windowing. Therefore, with only one DOS program running, the CPU only processes the program when it displays on your screen.

Suppose for example that you are working with a DOS program and return to the Presentation Manager. The DOS program stops running and only begins running again when you return to the DOS session. (If the DOS program is keeping track of the time of day by counting clock ticks, the program may have an incorrect time when it resumes.)

Many existing programs written for IBM DOS can run in DOS mode. Programs that may *not* run in DOS mode include:

- **Timing-dependent programs**
  such as communications and real-time programs.

- **Hardware-specific routines**
  such as device drivers and plotter programs.

- **Network-dependent programs.**

When a program is running under IBM DOS, it is the only task that the system can process at that time. The only other tasks that share the processor are background jobs, such as print spoolers or network message retrievers.

In OS/2, several programs can process in background OS/2 sessions while the DOS session is operating in the foreground. This means the behavior of a DOS program may change somewhat when there are many OS/2 programs running in the background. To ensure greater compatibility while running programs in the DOS session, make sure that no OS/2 programs are running in the background.

Notes:

1. Refer to "DOS Mode Compatibility" on page 97 for further information on running DOS programs.

2. To display the amount of fixed disk space available, use the CHKDSK command while in DOS mode. Refer to the *OS/2 Command Reference* for more information on CHKDSK.

Chapter 2. Customized Installation

There are many choices you can make to customize this multitasking, virtual memory operating system. This chapter describes ways to set up your system by making choices based on your computer type and the tasks you want to perform.

Configuring OS/2

During installation, OS/2 adds a CONFIG.SYS file in the root directory of your system. This single CONFIG.SYS file, which contains command statements that set up your system, configures the system for both DOS and OS/2 modes. Each time you start your system, OS/2 searches the root directory of the drive where you started OS/2 for a file named CONFIG.SYS. When OS/2 finds the file, OS/2 reads the file and interprets the statements.

Changing CONFIG.SYS

The statements OS/2 places in the CONFIG.SYS file provide default values to maximize performance for your particular system. Your operating system uses these values when you do not specify a value, but you may change them at any time. One reason to change the statements in the CONFIG.SYS file would be if the information that came with your programs recommends special values.

If you decide to change your CONFIG.SYS file after you install OS/2, you can either reuse the installation program and add, change, or delete command statements, or use an editor. If a CONFIG.SYS file already existed on your system before you installed OS/2, the installation program renames the old CONFIG.SYS file to CONFIG.BAK. This is so that the existing CONFIG.SYS file remains intact under the backup (.BAK) name. You can then, if you choose, combine the command statements from the two files. Also, in case the CONFIG.SYS file becomes damaged, system installation automatically creates a backup copy of CONFIG.SYS. (Refer to page 95 for the recovery procedure.)

In most cases, you probably would not want to repeat the entire installation process to make a few changes in the CONFIG.SYS file. However, you may consider rerunning the installation program if you have to change interrelated or device-dependent statements in the CONFIG.SYS file.

For example, suppose you did not start code page switching during system installation and you now want to use this feature. By reinstalling OS/2, you can let the installation program set up the inter-related DEVINFO, COUNTRY, and CODEPAGE statements in CONFIG.SYS for you. It uses the information provided by your responses to construct the appropriate statements for your CONFIG.SYS file. If you do decide to change your CONFIG.SYS file by reinstalling the system, refer to "Installing OS/2" in *Getting Started*.
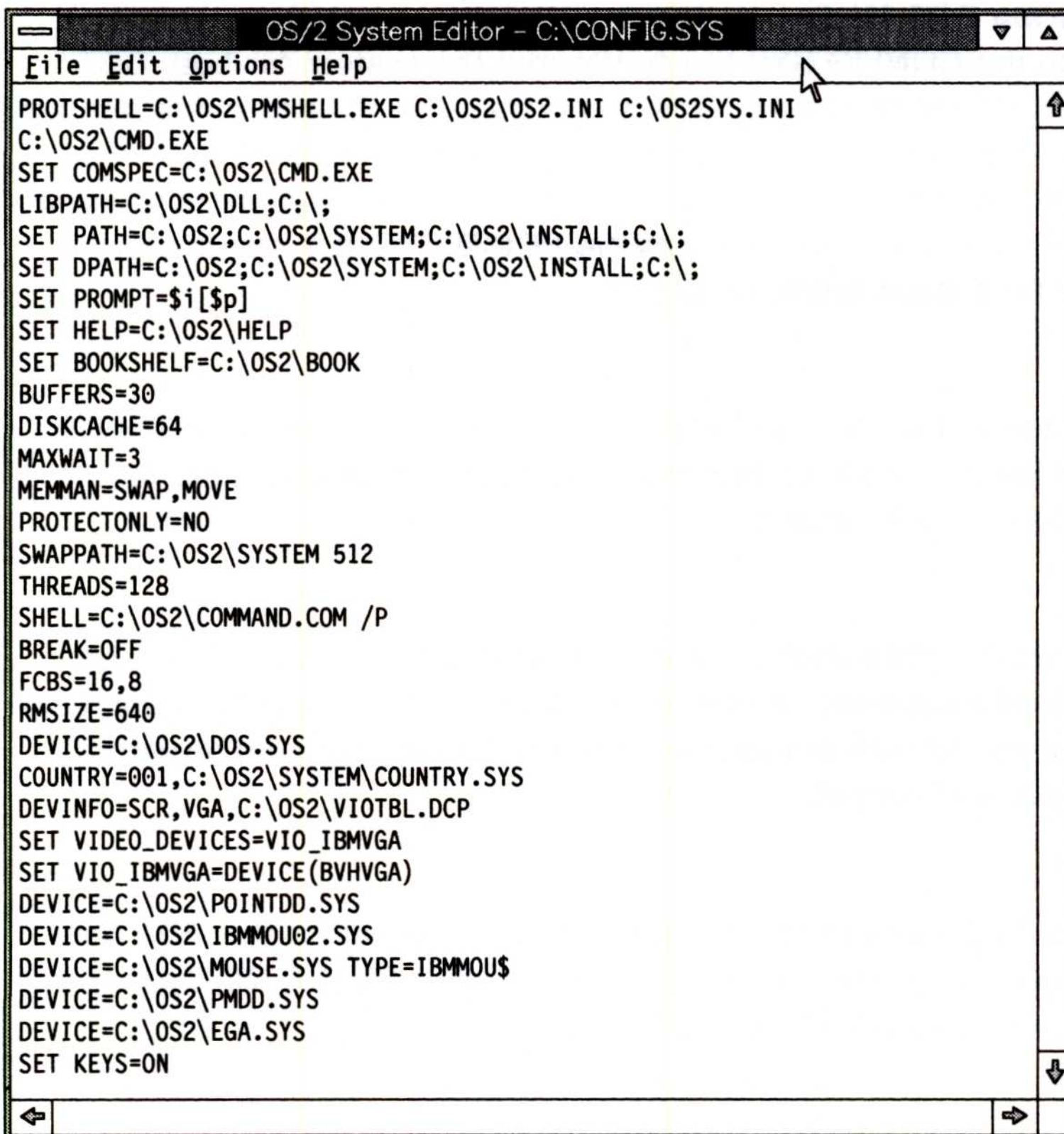
On the following page is an example of what you might find in your CONFIG.SYS file if you install OS/2 selecting the defaults on an IBM PS/2® Model 60 with an attached IBM Personal System/2® mouse. Keep in mind that OS/2 reads CONFIG.SYS only during system startup. Any changes to the file do not take effect until you restart the system.

Note: You can specify the following statements multiple times in your CONFIG.SYS file. Keep in mind that since some device drivers, such as MOUSE.SYS, POINTDD.SYS, and COM0x.SYS, have interdependencies, OS/2 processes these statements in the order you specify.

> DEVICE
> DEVINFO
> IOPL (*list only*)
> REM
> RUN
> SET
> TRACE

If you specify any statements more than once, OS/2 uses the last correct statement.

PS/2 and Personal System/2 are registered trademarks of the International Business Machines Corporation.

```
┌─────────────────────────────────────────────────────────────────────┐
│ ═══        OS/2 System Editor – C:\CONFIG.SYS              ▽  △        │
│  File  Edit  Options  Help                        ⬉                   │
├─────────────────────────────────────────────────────────────────┬───┤
│ PROTSHELL=C:\OS2\PMSHELL.EXE C:\OS2\OS2.INI C:\OS2SYS.INI         │ ⇧ │
│ C:\OS2\CMD.EXE                                                    │   │
│ SET COMSPEC=C:\OS2\CMD.EXE                                        │   │
│ LIBPATH=C:\OS2\DLL;C:\;                                           │   │
│ SET PATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;                 │   │
│ SET DPATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;                │   │
│ SET PROMPT=$i[$p]                                                 │   │
│ SET HELP=C:\OS2\HELP                                              │   │
│ SET BOOKSHELF=C:\OS2\BOOK                                         │   │
│ BUFFERS=30                                                        │   │
│ DISKCACHE=64                                                      │   │
│ MAXWAIT=3                                                         │   │
│ MEMMAN=SWAP,MOVE                                                  │   │
│ PROTECTONLY=NO                                                    │   │
│ SWAPPATH=C:\OS2\SYSTEM 512                                        │   │
│ THREADS=128                                                       │   │
│ SHELL=C:\OS2\COMMAND.COM /P                                       │   │
│ BREAK=OFF                                                         │   │
│ FCBS=16,8                                                         │   │
│ RMSIZE=640                                                        │   │
│ DEVICE=C:\OS2\DOS.SYS                                             │   │
│ COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS                             │   │
│ DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP                                 │   │
│ SET VIDEO_DEVICES=VIO_IBMVGA                                      │   │
│ SET VIO_IBMVGA=DEVICE(BVHVGA)                                     │   │
│ DEVICE=C:\OS2\POINTDD.SYS                                         │   │
│ DEVICE=C:\OS2\IBMMOU02.SYS                                        │   │
│ DEVICE=C:\OS2\MOUSE.SYS TYPE=IBMMOU$                              │   │
│ DEVICE=C:\OS2\PMDD.SYS                                            │   │
│ DEVICE=C:\OS2\EGA.SYS                                             │   │
│ SET KEYS=ON                                                       │ ⇩ │
├───┬───────────────────────────────────────────────────────────┬──────┤
│ ⇦ │                                                           │  ⇨   │
└───┴───────────────────────────────────────────────────────────┴──────┘
```

Note: Refer to the *OS/2 Command Reference* for the syntax of the command, associated parameters, and examples.

Statements Required to Restart Your System

If you change certain statements in your CONFIG.SYS file after installation, your system might not be able to restart. This is because OS/2 requires certain statements in CONFIG.SYS that inform the system of the location of files that do not exist in the root directory of the startup drive.

The following statements are required in the CONFIG.SYS file to restart your system. Keep in mind that parameters may vary based on your system setup.

```
SET PATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
LIBPATH=C:\OS2\DLL;C:\;
COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
DEVICE=C:\OS2\PMDD.SYS
```

Configuring DOS Mode

Although the CONFIG.SYS file configures the system for both DOS and OS/2 modes, some statements pertain only to DOS mode and are ignored in OS/2 mode. The following commands apply only to DOS mode and then only if the PROTECTONLY=NO statement is present in the CONFIG.SYS file. Refer to the PROTECTONLY command on page 72 for a description of its use.

BREAK

Allows you to instruct DOS to check if you have pressed the Ctrl and Break keys together when a program requests OS/2 to perform any functions.

FCBS

Determines file control block management information for the DOS environment. A file control block (FCB) is a record that contains all the information about a file (for example, its structure, length, and name).

RMSIZE

Specifies the highest storage address allowed for the DOS operating environment. If RMSIZE is less than 640KB, OS/2 programs can use the remaining memory up to 640KB.

SHELL

Specifies the name of the top-level command processor that OS/2 loads in DOS mode. (Usually DOS mode requires the default file named COMMAND.COM.)

Note: OS/2 ignores the DOS commands FILES and LASTDRIVE.

Changing AUTOEXEC.BAT

The first time you select the DOS command prompt from the Presentation Manager, the DOS command processor, COMMAND.COM, automatically processes a batch file named AUTOEXEC.BAT in the root directory of the startup drive. Similar to the CONFIG.SYS file, OS/2 creates this file for you during system installation. It is useful if you want to run commands in DOS mode, set a path, or process another program or batch file. After you have installed OS/2 on your fixed disk, the following is an example of what you might find in the AUTOEXEC.BAT file:

```
OS/2 System Editor - C:\AUTOEXEC.BAT
File  Edit  Options  Help
PATH C:\OS2;C:\OS2\SYSTEM
APPEND C:\OS2;C:\OS2\SYSTEM
SET COMSPEC=C:\OS2\COMMAND.COM
CALL HELP ON
```
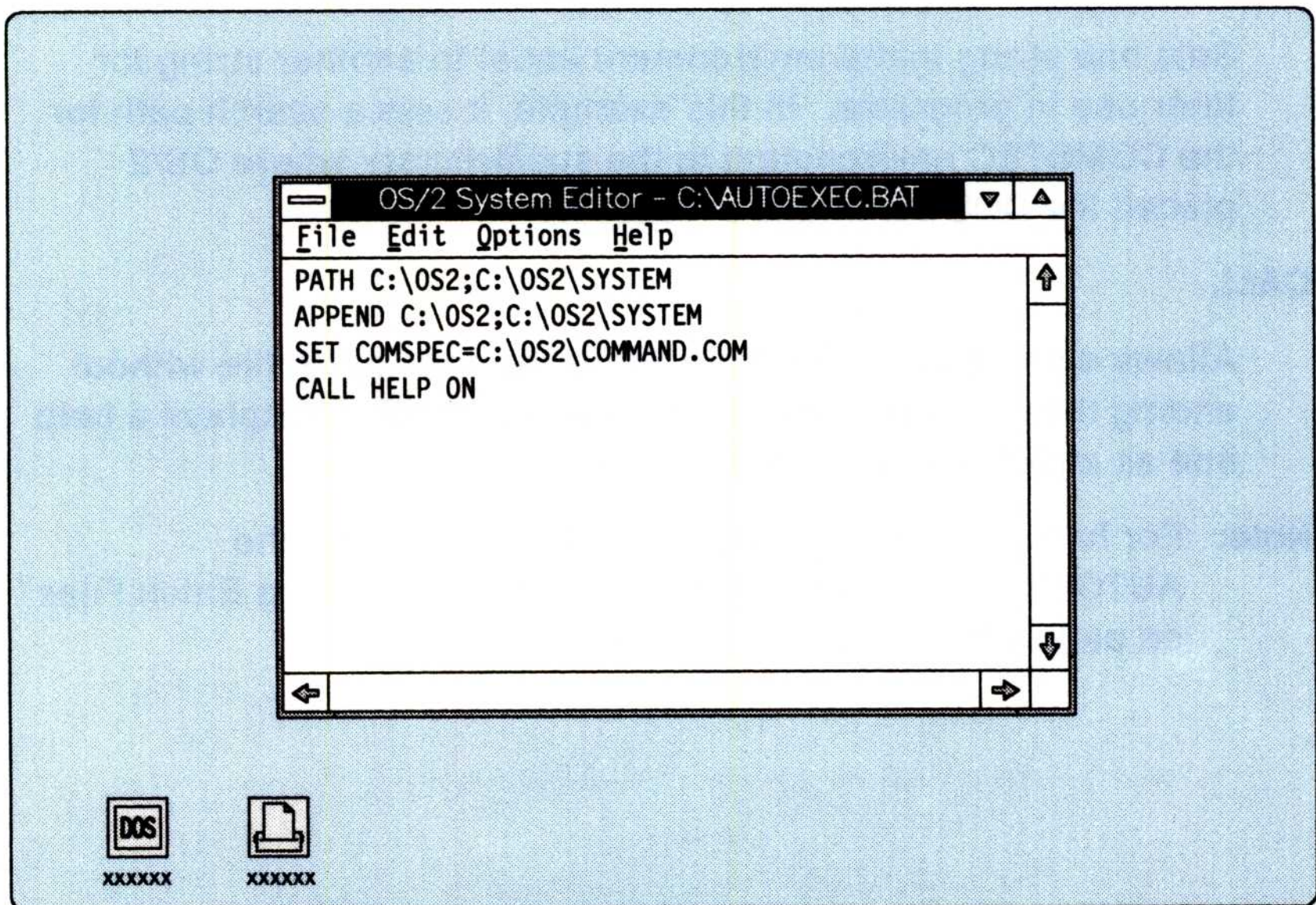
Definitions of the command statements are as follows:

PATH

> Searches the specified directories for program files that OS/2 cannot find by searching the current directory.

APPEND

> Establishes a path for your system to search for nonprogram (data) files outside the current directory.

SET

> Sets one string in the environment equal to another string for later use in programs. In this example, it sets a search path for the COMSPEC environment to the subdirectory where OS/2 placed the DOS command processor.

CALL

> Allows a file to be called from within another batch file without ending the first batch file. In this example, HELP displays a help line as a part of the command prompt.

Note: For further information on batch files, including the AUTOEXEC.BAT file, refer to "Creating and Using Batch Files" on page 107.

Installing Device Drivers

In a multitasking environment, programs share resources including physical devices. A device driver is a file containing the code needed to manage information being sent to or being received from a specific hardware device (for example, from the keyboard to your display). It defines the interface between the operating system or its programs and the physical device.

You can replace device drivers or add other devices by using DEVICE statements in the CONFIG.SYS file. The DEVICE statement installs device drivers by specifying in your CONFIG.SYS file the path and complete file name of the device driver.

When you start your system, OS/2 device drivers can process requests from either DOS or OS/2 programs. You also start DOS device drivers when you start your system, but they can process requests only from DOS programs. Also, keep in mind that OS/2 processes DEVICE statements in the order in which they appear in the CONFIG.SYS file.

OS/2 and its device drivers manage the physical devices for programs running in both the OS/2 and DOS environments. In OS/2 mode, there are also special drivers called *presentation drivers* that service requests from programs running in Presentation Manager sessions. These drivers add additional subsystem support for printers, plotters, and video screens.

OS/2 loads the standard default presentation drivers that support the keyboard, display, printer, diskette drive, fixed disk, and clock devices. You must not list these device drivers in the CONFIG.SYS file with DEVICE statements.

Note: For information on adding a printer or plotter to your system, refer to "Adding a Printer Driver" on page 37.

Your OS/2 diskettes include the following device drivers. For further information on these device drivers, refer to the DEVICE command in the *OS/2 Command Reference*.

Device Driver	Purpose
ANSI.SYS	Allows extended screen and keyboard support in DOS mode.
COM0x.SYS	Allows OS/2 application programs or system programs such as SPOOL to use serial devices.
EGA.SYS	Allows DOS programs that require Enhanced Graphics Adapter support to be run.
EXTDSKDD.SYS	Allows access to an external diskette drive by referencing a logical drive letter.
MOUSE.SYS	Implements support for pointing devices.
PMDD.SYS	Provides pointer draw device driver support for the Presentation Manager.
POINTDD.SYS	Provides mouse pointer draw support.
VDISK.SYS	Installs a simulated disk called a virtual disk.

To use these device drivers, include a DEVICE statement in the CONFIG.SYS file. Once you have added it to the CONFIG.SYS file, you must restart the system before the device driver operates.

Note: After you have installed OS/2, OS/2 provides an automated way to install new device drivers (except presentation drivers) without using the installation program. Refer to the DDINSTAL command in the *OS/2 Command Reference* for further information.

Updating Support for Your Display Adapter

If you change your display and adapter after you have installed OS/2, you must also change the display adapter support on your system. You can either reinstall OS/2 or update Presentation Manager display adapter support. To change display adapter support, follow these steps:

1. Select the correct combination of .DLL files that support your display adapter:

IBMCGA.DLL **BVHCGA.DLL**	IBM Color Graphics Adapter
IBMCGA.DLL **BVHEGA.DLL**	IBM Enhanced Graphics Adapter with 64KB of graphics memory
IBMEGA.DLL **BVHEGA.DLL**	IBM Enhanced Graphics Adapter with greater than 64KB of graphics memory
IBMVGA.DLL **BVHVGA.DLL**	IBM Personal System/2 Video Graphics Array
IBMVGA.DLL **BVHVGA.DLL**	IBM Personal Computer AT® or an IBM PC XT™ Model 286 with an IBM Personal System/2 Display Adapter
IBMVGA.DLL **BVHVGA.DLL** **BVH8514A.DLL**	IBM Personal System/2 8514/A
IBMBGA.DLL **BVHVGA.DLL** **BVH8514A.DLL**	IBM Personal System/2 8514/A with memory expansion
IBMVGA.DLL **BVHMPA.DLL**	IBM Personal Computer AT or an IBM PC XT Model 286 with a PS/2 Display Adapter and a PS/2 Monochrome Display
BVHMPA.DLL	IBM Personal Computer AT with a Monochrome Adapter

---

Personal Computer AT and PC XT are registered trademarks of the International Business Machines Corporation.

2. Turn on or restart your system.

3. Edit the CONFIG.SYS file using an editor such as the System Editor.

4. Type and save the appropriate statements in your CONFIG.SYS file using the correct files in step #1 and these parameters:

```
SET VIDEO_DEVICES=VIO_IBMxxx, VIO_IBMxxx
SET VIO_IBMxxx=DEVICE(BVHxxx,BVHxxx)
```

Note: Depending on your particular adapter type, the second `,VIO_xxx` and `BVHxxx` files may be optional. This is because for every `VIO_IBMxxx` file attached to the `SET VIDEO_DEVICES=` statement, there must a corresponding `SET VIO_IBMxxx=` statement.

For example, if you have a Color Graphics Adapter, type:

```
SET VIDEO_DEVICES=VIO_IBMCGA
SET VIO_IBMCGA=DEVICE(BVHCGA)
```

If your system has only one display and it is attached to the 8514/A adapter, type:

```
SET VIDEO_DEVICES=VIO_IBM8514A
SET VIO_IBM8514A=DEVICE(BVHVGA,BVH8514A)
```

If you are installing an 8514/A adapter and have two displays attached to your system, type:

```
SET VIDEO_DEVICES=VIO_IBMVGA, VIO_IBM8514A
SET VIO_IBMVGA=DEVICE(BVHVGA)
SET VIO_IBM8514A=DEVICE(BVH8514A)
```

5. If you are adding a high resolution display, insert OS/2 diskette #5.

6. Enter the following, substituting the correct file in step 1 to rename the support file:

```
UNPACK A:IBMxxx.DL@ C:\OS2\DLL
UNPACK A:BVHxxx.DL@ C:\OS2\DLL
```

Note: If you are installing a monochrome adapter on an IBM Personal Computer AT, there is no `IBMxxx.DLL` file to unpack.

For example, enter the following for an 8514/A Adapter with memory expansion:

```
UNPACK A:IBMBGA.DL@ C:\OS2\DLL
UNPACK A:BVHVGA.DL@ C:\OS2\DLL
UNPACK A:BVH8514A.DL@ C:\OS2\DLL
```

7. Remove OS/2 diskette #5 from drive A.

8. Insert the *Installation* diskette in drive A; then, press and hold the Ctrl, Alt, and Del keys together to restart the system.

9. Press the Escape (Esc) key when OS/2 displays the IBM logo.

10. Enter the following to change to drive C:

```
C:
```

11. Enter the following to change to the DLL subdirectory:

```
CD \OS2\DLL
```

12. To copy the support file, enter the following substituting the correct file in step 1:

```
COPY \IBMxxx.DLL DISPLAY.DLL
```

Note: If you are installing a monochrome adapter on an IBM Personal Computer AT, there is no IBM*xxx*.DLL file to copy to DISPLAY.DLL because you cannot run Presentation Manager on a monochrome display.

For example, enter the following for an 8514/A Adapter with memory expansion:

```
COPY IBMBGA.DLL DISPLAY.DLL
```

13. Turn the system off, and add the physical adapter you want to install.

14. Turn the system on to update the adapter support on your system.

Adding a Mouse after Installation

To add a mouse after installation without having to re-install the OS/2 program, follow these steps:

Note: All statements for serial mouse device drivers must precede any COM0x.SYS statements in the CONFIG.SYS file.

1. To provide mouse pointer draw support for the mouse types listed below, edit the CONFIG.SYS file and enter:

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

2. Enter the DEVICE statement listed below in your CONFIG.SYS file, using only the parameters that are appropriate for your particular system and mouse.
   (The `SERIAL=COMx` and `MODEL=xxx` parameters are optional.)

   `DEVICE=C:\OS2\filename SERIAL=COMx MODEL=xxx`

   - Substitute these parameters if you have an IBM Personal Computer AT, IBM PC XT Model 286, or IBM PS/2 Model 30-286:

filename	Specifies the complete name of the file containing the mouse device driver. Use:
IBMMOU01.SYS	IBM Personal System/2 Mouse
MSBUS01.SYS	Microsoft[1] Bus Mouse
MSINP01.SYS	Microsoft In-Port Mouse Microsoft Mouse with Bus Interface
MSSER01.SYS	Microsoft Serial Mouse Microsoft Mouse with Serial Interface
MSPS201.SYS	Microsoft Mouse with PS/2 Interface
PCMOU01.SYS	PC Mouse Systems[2]
VISION01.SYS	Visi-On[3]

---

[1] Trademark of the Microsoft Corporation.

[2] Trademark of the Metagraphics/Mouse Systems.

[3] Trademark of VisiCorp.

SERIAL = COMx Specifies the serial communications port
connected to the mouse device driver.
Use:

1   First COM port (default)
2   Second COM port

Note: This parameter is not valid for the IBM Personal
System/2 Mouse, Microsoft Mouse with Bus Interface,
or Microsoft In-Port Mouse.

MODEL = xxx Specifies the model number for the mouse.
Use:

099 or 199   Microsoft Serial Mouse, Microsoft Bus
Mouse, or Microsoft Mouse with Serial
Interface

Note: This parameter is only applicable to the above mice.

• Substitute these parameters if you have an IBM Personal
System/2 Model 50, 55, 60, 70, or 80:

filename   Specifies the complete name of the file con-
taining the mouse device driver. Use:

IBMMOU02.SYS   IBM Personal System/2 Mouse

MSSER02.SYS   Microsoft Serial Mouse
Microsoft Mouse with Serial Interface

PCMOU02.SYS   PC Mouse Systems

VISION02.SYS   Visi-On

MSPS202.SYS   Microsoft Mouse with PS/2 Interface

SERIAL = COM*x*   Specifies the serial communications port connected to the mouse device driver. Use:

1 through 8   Communication ports
                           (COM1 - default)

Note: This parameter is not valid for the IBM Personal System/2 Mouse (IBMMOU02.SYS).

MODEL = *xxx*   Specifies the model number for the mouse. Use:

099 or 199   Microsoft Serial Mouse
                           Microsoft Mouse with Serial Interface

Note: This parameter is only applicable to the above mice.

3. Enter a second DEVICE statement in your CONFIG.SYS file, using only the parameters appropriate for your system and mouse.

DEVICE=C:\OS2\MOUSE.SYS MODE=*m* QSIZE=*q* TYPE=*name*

Note: The MODE=*m* and QSIZE=*q* parameters are optional.

MODE = *m*   Specifies the mode that the mouse device driver supports. Use:

P   OS/2 mode only

R   DOS mode only

B   Both (default)

QSIZE = *q*   Specifies the length of the queue for events to be used for all OS/2 mode tasks. Use:

1 - 100   Event records (default = 10)

TYPE=*name*	Specifies the name of the device driver handling the hardware. Use:

IBMMOU$	IBM Personal System/2 Mouse
MSSER$	Microsoft Serial Mouse Microsoft Mouse with Serial Interface
MSINP$	Microsoft In-Port Mouse Microsoft Mouse with Bus Interface
MSPS2$	Microsoft Mouse with PS/2 Interface
PCMOU$	PC Mouse Systems
VISION$	Visi-On

4. Restart the system by pressing the Ctrl, Alt, and Del keys together.

If, for example, you want to install an IBM Personal System/2 mouse on a PS/2 Model 50, 55, 60, 70 or 80, enter the following in CONFIG.SYS:

```
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\IBMMOU02.SYS
DEVICE=C:\OS2\MOUSE.SYS TYPE=IBMMOU$
```

Instead, if you want to install a Microsoft mouse with a serial interface on your IBM Personal System/2 using communications (COM) port 2 in OS/2 mode, enter:

```
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MSSER02.SYS SERIAL=COM2 MODEL=199
DEVICE=C:\OS2\MOUSE.SYS TYPE=MSSER$ MODE=P QSIZE=15
DEVICE=C:\OS2\COM02.SYS
```

When you start your system, OS/2 loads the mouse device driver using COM2 port, leaving the COM1 and COM3 ports available for the COM02.SYS device driver. As the COM02.SYS device driver installs, a message tells you that OS/2 could not locate the device adapter; the device adapter is not available because the mouse claimed the port. You must enter the COM0x.SYS statement after the mouse statements so that the mouse can gain access to the COM port.

Installing the OS/2 Command Reference
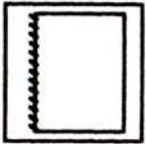
If you would like to install the *OS/2 Command Reference* without having to re-install your entire system, follow these steps:

1. Start your system.

2. Insert OS/2 diskette #3 into drive A.

3. From the Group-Main window, double-click on **OS/2 Window**, or select it and press the Enter key.

4. At the command prompt, enter:

   ```
   MKDIR  C:\OS2\BOOK
   ```

   This creates a BOOK subdirectory.

5. At the command prompt, enter:

   ```
   UNPACK  A:CMD*.*  C:\OS2\BOOK
   ```

   This copies files beginning with CMD from the diskette to the BOOK subdirectory.

6. At the command prompt, enter:

   ```
   UNPACK  A:VIEW*.*  C:\OS2
   ```

   This copies the *OS/2 Command Reference* program from the diskette to the OS2 subdirectory.

7. Remove the diskette from drive A.

8. Using an editor program such as the System Editor, type and save this statement in the CONFIG.SYS file:

   ```
   SET BOOKSHELF=C:\OS2\BOOK
   ```

9. Restart your system by pressing the Ctrl, Alt, and Del keys together.

From the Group-Main window, double-click on **OS/2 Command Reference**, or select it and press the Enter key.

Note: If you want to add the *OS/2 Command Reference* to the Group-Utilities window, refer to "Adding Program Titles to a Program Group Window" in *Getting Started*.

Spooling and Printing

You can send information directly to a device or you can send it to the *spooler*. This manages your system's printing process by placing data as print jobs in temporary files on disk. Since disk I/O is faster than printer I/O, OS/2 can then select which job to run next when scheduling jobs without being affected by the type and volume of printer I/O. This gives you flexibility in organizing and optimizing the use of system printing resources.

In addition, spooling allows the printing devices to be shared among several processes in the system without interference. OS/2 avoids intermixing data among programs running at the same time by intercepting and separating information going to the printing device from different processes. The data is printed or plotted when it is complete and the required device is available.

The spooler is installed and available by default. You can, however, start the system without the spooler (see page 53). You can also remove the spooler during a session, and then restart the system. When you restart the system, existing printer queues remain intact, though print jobs in the process of being created may be lost. In addition, the spooler responds dynamically to any changes you make. For example, if you add a printer to a running system, the spooler automatically supports the printer without having to be restarted.

The instructions on the following pages help you use the Control Panel and the Print Manager windows to configure the printing resources of your system. Also, refer to "Printing Files" in *Getting Started* for details on how to print a file through direct manipulation to the Print Manager.

The following figure is an example of the spooling process:
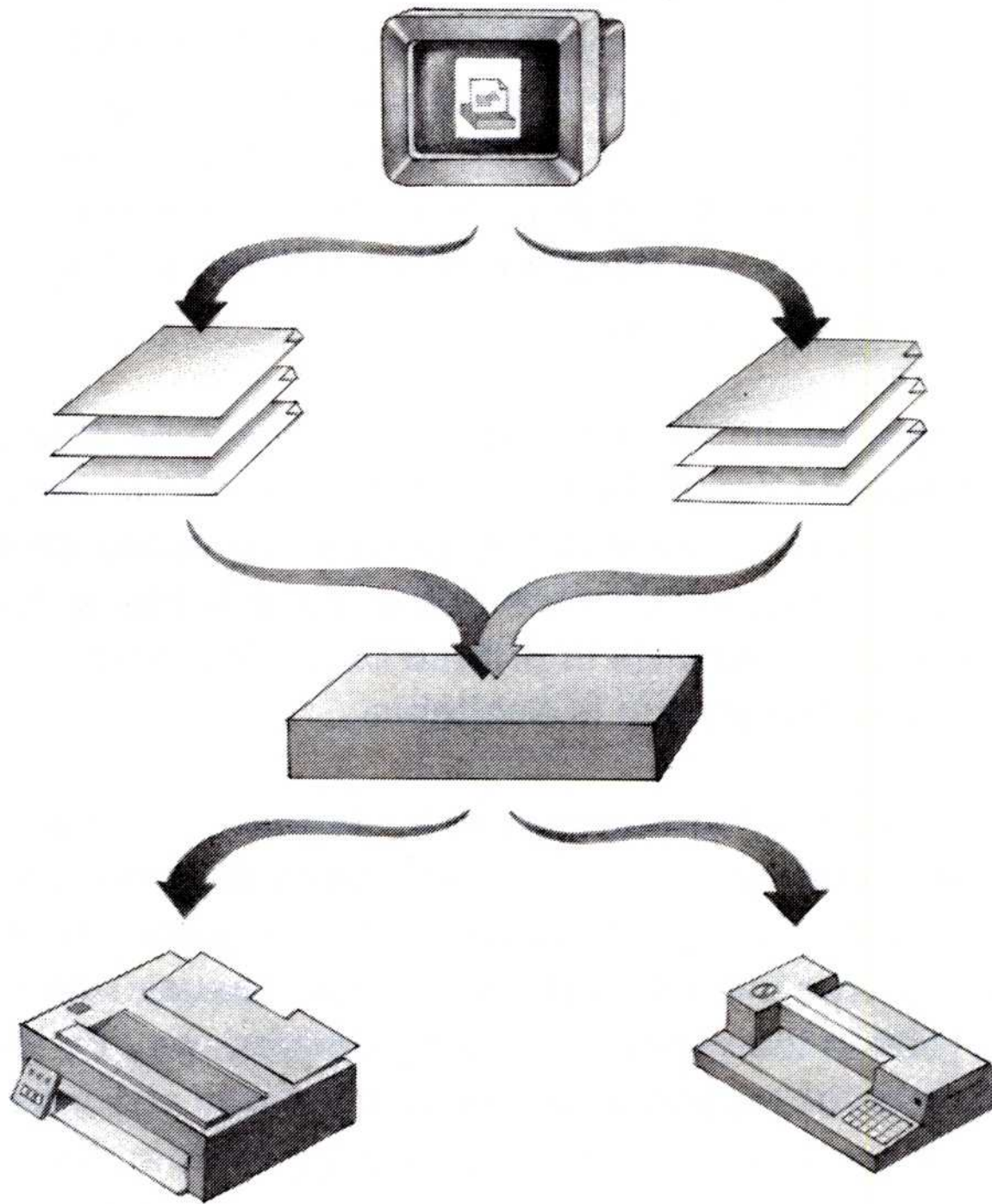
Figure 6. Spooling

When you submit a job for printing, the job is distributed and stored in a print queue with other submitted print jobs. The print job stays in the queue until the spooler, which continuously examines each available print queue, decides which print job to send to the next available printer or plotter.

Next, each print queue uses an installed queue driver (like PMPRINT) to do hardware-independent processing common to all jobs on the queue. The queue uses a queue driver to translate the jobs received into a format which it can pass to the device driver of the physical device.

Although several queues can use the same queue driver, each queue is set up to use only one queue driver unless a program requests a different queue driver for a particular job. Each queue can use one printer or it can be set up to use several printers at the same time so that it is making better use of available resources.

Procedures for Setting Up a Printer or Plotter

Follow the instructions outlined in the table below to set up either a printer or plotter on your system.

To attach a new printer or plotter, you first need to set up printer support on your system before creating or setting up queues by following the steps in the table below. For example, you will need to:

- Add a printer driver (parallel or serial, depending on the device).
- Add a printer name; then, connect it to a port and a printer driver.
- Add a queue name.
- Change your printer driver settings.

Once you have completed these tasks, you have the choice of taking optional actions such as changing your default printer or queue. If you are installing a serial device, you might also want to set it up for base printing (refer to page 50).

Procedure	Page
Adding a printer driver	37
Adding a printer name	44
Connecting the printer name you added with a port	45
Connecting printer drivers to your printer	46
Adding a queue name	47
Changing your default printer and queue	48
Changing printer driver settings	49
Setting up a serial device for base printing	50

Displaying the Control Panel

The Control Panel is a program which allows you to select choices to set up system preferences that act across the system. The Control Panel contains system settings, such as display and file options, in a file called OS2.INI, located in the \OS2\DLL directory of your fixed disk.

Follow these steps to display the Control Panel:

1. Press the Ctrl and Esc keys to view the Task List window.

2. Double-click on **Desktop Manager**, or select it and press the Enter key.

3. Double-click on **Utility Programs**, or select it and press the Enter key.

4. Double-click on **Control Panel**, or select it and press the Enter key.

   OS/2 displays the Control Panel window.
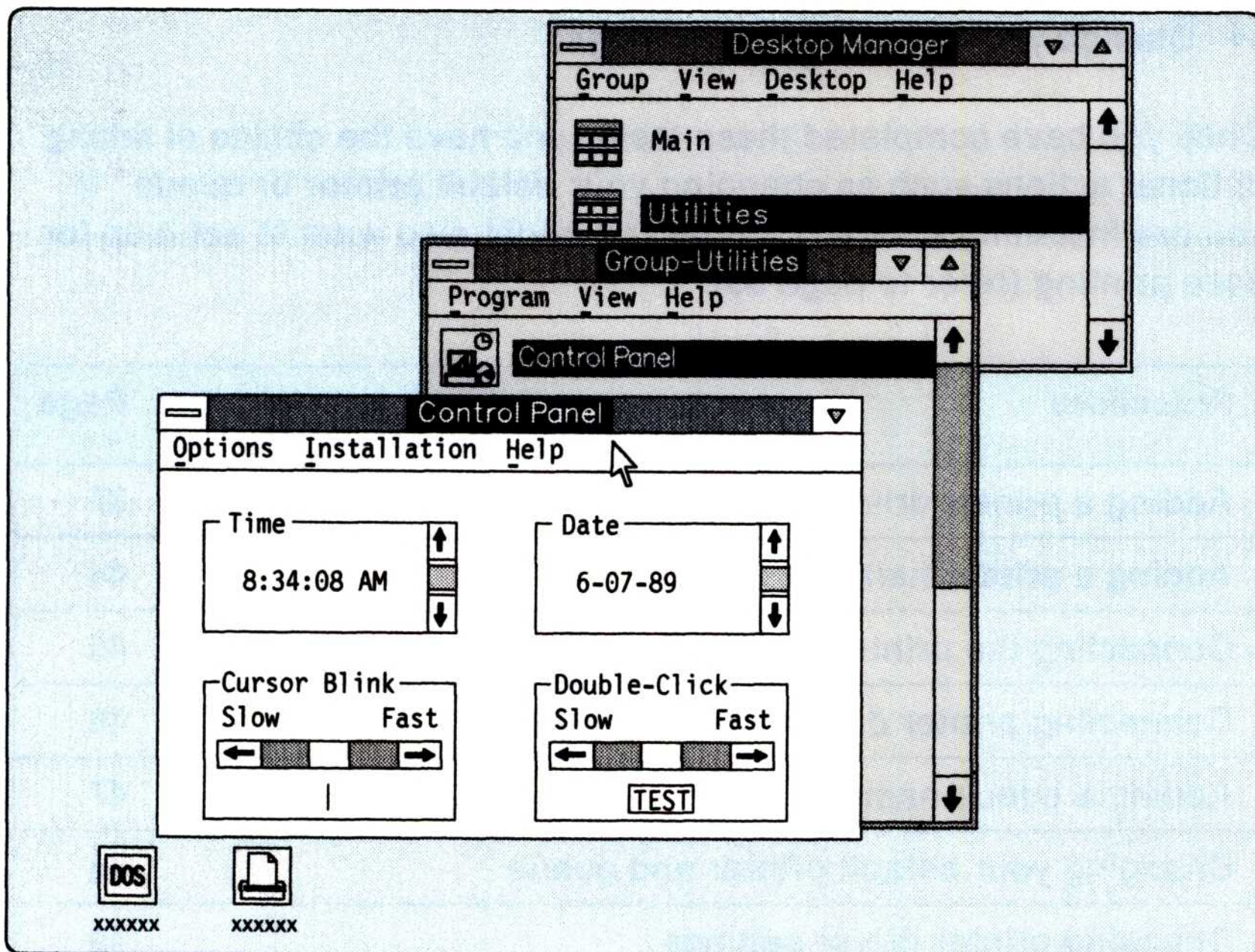
When you are ready to close the Control Panel window, select **Exit** from the Options pull-down. You may also select **Close** from the System Menu located in the pull-down in the upper left-hand corner of the window.

Note: You can install a maximum of nine device drivers using the Control Panel. This does not include device drivers you have installed by adding statements to the CONFIG.SYS file.

Adding a Printer Driver

Printer drivers are device drivers that provide hardware-specific information about a particular printer or plotter. The system needs this information to use the device.

Some types or models of printers are supported by more than one device driver, which means that they can emulate different types of printers. Which printer driver is used depends on the selection made by the print queue or program that is printing the job.

If you want to use a printer or plotter with the OS/2 program, you need to install on your system a printer driver to support the physical device. During installation, the 4201 printer driver is installed for you. Located in the C:\OS2 subdirectory, this printer driver supports most graphic and text printers. If, however, you want to add a printer that has special control characters, you might need to install a printer driver from the *Presentation Manager Device Drivers* diskette.

Before you add a printer to your system, you need to determine its type. The instructions that came with your printer should describe whether it is serial or parallel. Once you know your printer type, follow the instructions for your particular printer or plotter. Keep in mind that you can set up several printer drivers which support the same printer.

Also, before beginning, make sure that you have attached your printer cable to the correct port on the back of your computer. Connect a parallel printer to any parallel port; connect a serial printer to a COM (communications) port.

Parallel Printer Driver

To add a parallel printer driver, follow these steps:

1. From the Control Panel, click on **Installation**, or select it and press the Enter key.

2. Click on **Add printer driver**, or select it and press the Enter key.

3. Insert the *Presentation Manager Device Drivers* diskette containing the printer driver into drive A.

   If your printer driver programs are in a directory on your fixed disk instead of on a diskette, type the correct drive and directory name in the entry field. Do not type the file name.

4. To display the names of the printer drivers, click on **Add**, or press the Enter key.

   Wait until the drivers appear in the Add Printer Driver window.

5. Click on or select the printer driver name for the printer you are installing.

   You may select more than one printer driver in a directory to install at once.

6. Decide whether you want to copy the printer driver to another directory or the C:\OS2\DLL directory.

7. Click on **Add**, or press the Enter key.

Serial Printer or Plotter Driver

Follow these steps to add a serial printer driver. (If you selected the option of "Serial Device Support" during system installation, begin this procedure with step 3. You can bypass steps 1 and 2 because the OS/2 program automatically sets up the DEVICE statements in the CONFIG.SYS file for you.)

1. Edit the CONFIG.SYS file (located in the root directory) using an editor such as the System Editor.

   Note: A backup copy of the CONFIG.SYS file is located in the C:\OS2\INSTALL directory. Refer to page 95 for the recovery procedure if you run into any difficulty.

2. Choosing from the following statements, type the appropriate DEVICE statement in your CONFIG.SYS file:

   * If you have a Personal Computer AT, Personal Computer XT Model 286, or PS/2 Model 30-286, type:

     `DEVICE=C:\OS2\COM01.SYS`

     This device driver supports ports COM1 and COM2.

   * If you have an IBM Personal System/2 Model 50, 55, 60, 70, or 80, type:

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

     This device driver supports ports COM1, COM2, and COM3.

   Note: If you already have a DEVICE statement that supports a printer driver at serial ports COM1 through COM3, the port is unavailable to the COM0x.SYS device driver.

3. From the Control Panel, click on **Installation**, or select it and press the Enter key.

4. Click on **Add printer driver**, or select it and press the Enter key.

5. Insert the *Presentation Manager Device Drivers* diskette containing the printer driver into drive A.

Personal Computer XT is a trademark of the International Business Machines Corporation.

If your printer driver programs are in a directory on your fixed disk instead of on a diskette, type the correct drive and directory name in the entry field. Do not type the file name.

Note: If you need to add support for more than one plotter, enter the location where you previously installed PLOTTERS.DRV. The default path when you first installed OS/2 is C:\OS2\DLL.

6. To display the names of the printer drivers, click on **Add**, or press the Enter key.

   Wait until the printer drivers appear in the Add Printer window.

7. Click on or select the printer driver name for the printer you are installing.

   You may select more than one printer driver in a directory to install at once.

8. Decide whether you want to copy the printer driver to another directory or the C:\OS2\DLL directory.

9. Click on **Add**, or press the Enter key.

10. From the Control Panel, click on **Options**, or select it and press the Enter key.

11. To add a port, click on **Communications port**, or select it and press the Enter key.

    OS/2 displays the Communications Port window and lists the default settings for the COM1 port. Use these settings for communication between programs and serial printers.

12. Click on the settings that your serial printer requires, or use the settings that have already been selected by the system.

    Refer to the publication that came with your serial printer to determine how to set the communication settings and switches on your printer.

13. Click on **Set**, or press the Enter key.

Note: Continue the procedure outlined on page 35 for further instructions on completing the installation of your serial printer.

Deleting a Printer Driver

To delete a printer driver, follow these steps:

1. From the Control Panel, click on **Installation,** or select it and press the Enter key.

2. Click on **Delete printer driver,** or select it and press the Enter key.

3. Click on the printer driver that you want to delete, or select it.

4. Click on **Delete,** or select it and press the Enter key.

    OS/2 displays a panel so that you can confirm that you want to delete the selected printer driver from your system.

5. Click on **Yes,** or select it and press the Enter key.

Note: When deleting a printer driver for a plotter, OS/2 displays a panel so that you can delete the associated PLOTTERS.DRV file. Select NO if you intend to use a different plotter entry.

Displaying the Print Manager

The Print Manager, installed in the Group-Main window, incorporates the printer and queue setup so that you can manage and modify your entire printing environment. Use the Print Manager to check the status of print jobs, or use the functions in its window to set a priority, or to reprint, or cancel print jobs. Print Manager also provides you with the ability to hold and release jobs and queues.

To display the Print Manager window on your screen, do one of the following:

- Double-click on ▣ , or

- Press the Ctrl and Esc keys to display the Task List window. Double-click on **Print Manager**, or select it and press the Enter key.

  OS/2 displays the Print Manager window.
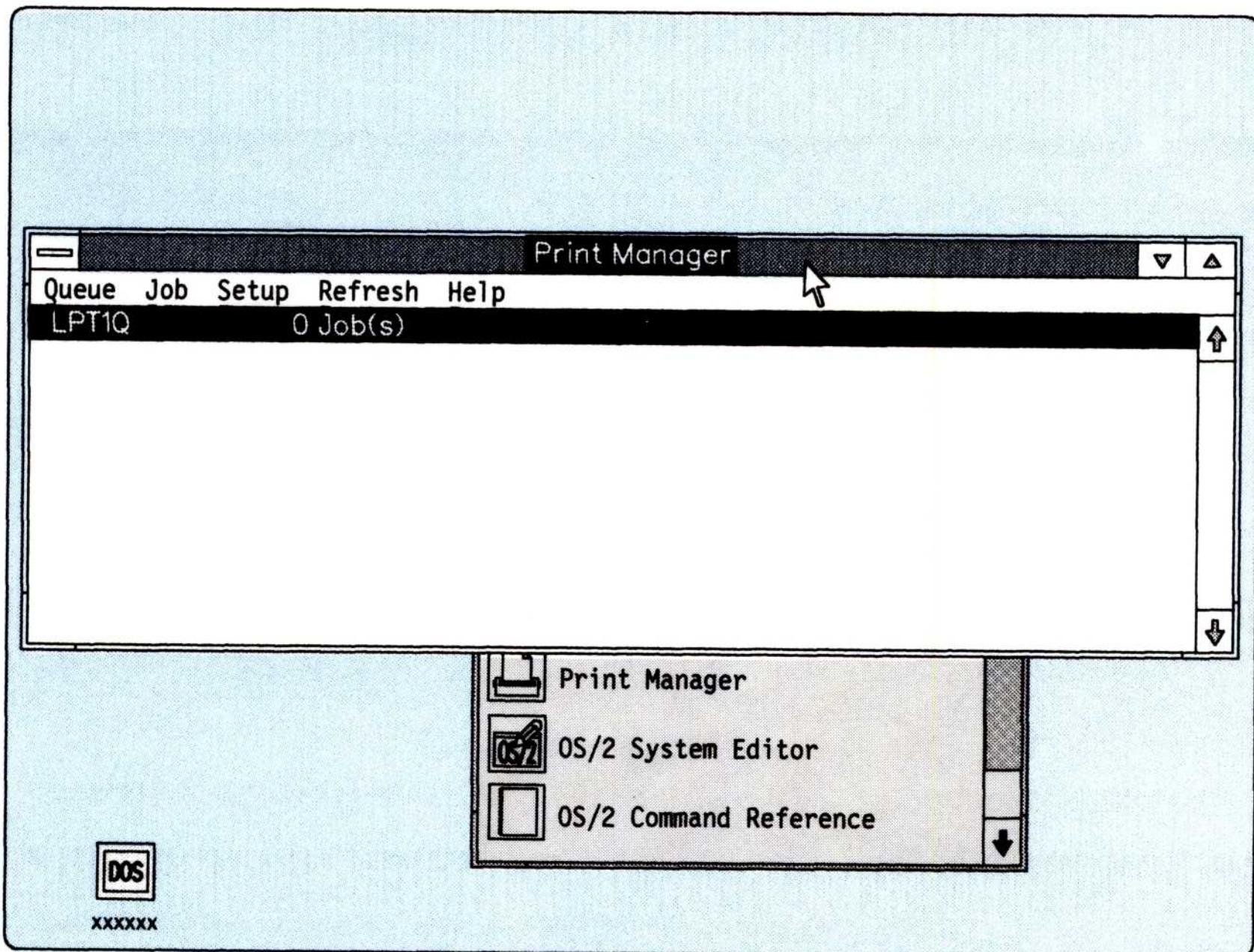
Notes:

1. When you have finished using the Print Manager window, select **Minimize** from the System Menu located in the pull-down in the upper left-hand corner of the window. As an alternative method, you can select the minimize icon in the upper right-hand corner of the window.

2. If you are running OS/2 without print spooling, start the Print Manager by selecting **Print Manager** from the Group-Main window. With spooling prevented, the Print Manager can still perform printer/queue tasks.

Adding, Changing, or Deleting Printer Names

Each printer is identified to the spooler by a name defined by you.
Follow these steps to add, change, or delete printer names:

1. From the Print Manager, click on **Setup**, or select it and press the Enter key.

2. To display the list of printers set up on your system, click on **Printers**, or select it and press the Enter key.

3. Choose from the following steps:

   To add a printer:

   a. Click on **Add**, or select it and press the Enter key.

   b. From the Add Printer window, type the name and description of the printer you want to add.

      Skip the other fields. They will be addressed later.

   c. Click on **Add**, or press the Enter key.

   d. At the Printers window, click on **Ok** or press the Enter key.

   To change a printer:

   a. Click on or select the name of the printer you want to change.

   b. Click on **Change**, or select it and press the Enter key.

   c. From the Change Printer window, make the necessary changes.

   d. Click on **Change**, or press the Enter key.

   e. At the Printers window, click on **Ok** or press the Enter key.

   To delete a printer:

   a. Click on or select the name of the printer you want to delete.

   b. Click on **Delete**, or select it and press the Enter key.

   c. Click on **Yes** to confirm your choice, or select it and press the Enter key.

Connecting Printer Names with Ports

A port is an outlet on computers where you connect printers and plotters. To connect a printer driver to a port, follow these steps:

1. From the Print Manager, click on **Setup**, or select it and press the Enter key.

2. Click on **Printers**, or select it and press the Enter key.

3. From the list of printer names, click on or select the name of a printer that you want to connect with a port.

4. Click on **Change**, or select it and press the Enter key.

5. From the Change Printer window in the Device field, click on the drop-down list box next to the device field for a list of available ports, or select a port using the arrow keys.

   Note: If no ports are available, click on or select **None**. While changing the connections between printers and ports, you can connect unattached printers to the None port. Unlike the other ports, you can connect more than one printer to the None port.

6. Click on **Change**, or press the Enter key.

7. At the Printers window, click on **Ok** or press the Enter key.

Connecting Printer Drivers to a Printer

To connect printer drivers to a printer, follow these steps:

1. From the Print Manager, click on **Setup**, or select it and press the Enter key.

2. Click on **Printers**, or select it and press the Enter key.

3. From the list of printer names in the Printers window, click on or select the name of the printer you want.

4. Click on **Change**, or select it and press the Enter key.

5. From the Change Printers window, click on the names of the printer drivers you want to connect with the selected printer.

   If you select more than one printer driver to be connected with a printer, the last one selected becomes the default driver. If you only want to select one printer driver, you will need to deselect the other one.

6. Click on **Change**, or press the Enter key.

7. At the Printers window, click on **Ok** or press the Enter key.

Adding, Changing, or Deleting Queue Names

You may set up multiple queues (ordered lists of jobs) on your system. This means that you can sort your print jobs according to some general criteria (for example, large batch jobs or small documents). Each job waits on the queue until it is printed on one of the printers that the queue uses.

A queue can use one printer or it can be set up to use several printers at the same time so that it is making better use of available resources. Also, any of these printers can be shared with multiple queues at the same time.

To add, change, or delete a queue, follow these steps:

Note: Set up the printers available on your system before creating or setting up queues.

1. From the Print Manager, click on **Setup**, or select it and press the Enter key.

2. To display a list of queues available, click on **Queues**, or select it and press the Enter key.

   The LPT1Q is already set up.

3. Choose from the following steps:

   To add a queue:

   a. Click on **Add**, or select it and press the Enter key.

   b. From the Add Queue window, type the name and description of the queue you want to add.

   c. Click on or select a queue driver, printers to be used by the queue, and the printer driver for this queue. If you only want to select one queue driver, you will need to deselect the other one.

   d. Click on **Add**, or press the Enter key.

   e. At the Queues window, click on **Ok** or press the Enter key.

To change a queue:

a. Click on or select the name of the queue you want to change.

b. Click on **Change**, or select it and press the Enter key.

c. From the Change Queue window, make the necessary changes.

d. Click on **Change**, or press the Enter key.

e. At the Queues window, click on **Ok** or press the Enter key.

To delete a queue:

a. Click on or select the name of the queue you want to delete.

b. Click on **Delete**, or select it and press the Enter key.

c. Click on **Yes** to confirm your choice, or select it and press the Enter key.

Changing Default Printers and Queues

Frequently, a local printer is attached to port LPT1 or, device name LPT1 is redirected to a network printer server. A queue is initially set up for use with this printer with the name LPT1Q.

Note: DOS and many OS/2 non-Presentation Manager programs print directly to the logical name PRN or the port name LPT1.

If you would like to change your default printer and queue, follow these steps:

1. From the Print Manager, click on **Setup,** or select it and press the Enter key.

2. Click on **Application defaults,** or select it and press the Enter key.

3. Click on or select the name of the queue that you want as your default queue.

4. Click on or select the name of the printer you want as your default printer.

5. Click on **Set**, or press the Enter key.

Changing Printer Driver Settings

To change printer driver settings, follow these steps:

Note: If you are changing plotter settings, refer to plotter consider-
ations in the README.DAT file located on the *Presentation
Manager Device Drivers* diskette.

1. From the Print Manager, click on **Setup**, or select it and press the
   Enter key.

2. Click on **Printers**, or select it and press the Enter key.

3. From the list of printer names in the Printers window, click on or
   select the name of a printer that is connected to the printer driver
   you want.

4. Click on **Change**, or select it and press the Enter key.

   This displays the Change Printer window which shows the setup
   of the selected printer.

5. Click on or select the name of the printer driver you want to
   change.

6. Click on **Printer Properties**, or select it and press the Enter key.

   The system displays a window that contains the settings you can
   change in the printer driver. The settings that are displayed
   depend on the printer driver you select. This is where you would
   add, change, or delete printer forms. It is also the only method to
   change the default spool file type.

7. After making your selections, click on **Enter**, or press the Enter
   key.

Setting Up Serial Devices for Base Printing

Base printing is printing without using Presentation Manager functions; for example, using the SPOOL command on the command line, or entering keyboard functions like PrtSc or Ctrl + PrtSc. Base printing is also printing using file system API calls such as DosOpen, DosWrite, or DosClose.

If you are adding a serial device and plan to use base printing, it is necessary for you to set up a *dummy* (fictitious) printer. A dummy parallel printer is required since base printing only supports output going to parallel devices (for example, LPT1, LPT2, LPT3, and PRN).

To add a dummy redirect printer, follow the instructions on pages 44, 45, and 46 for adding printer names, ports, and drivers. The parameters you need are:

Name	Redirect
Port	A parallel port which is not yet used (such as LPT3)
Printer Driver	None (deselect all drivers)

You then can use the SPOOL command at the command prompt to redirect the parallel port to the serial port. For example, to redirect input from a parallel port (LPT3) to the printer attached to a serial port (COM1), enter:

```
SPOOL /D:LPT3 /O:COM1
```

At this point, if you want to print to the serial device attached to COM1, enter the following at a command prompt:

```
COPY filename LPT3:
```

Substitute the name of your file in place of *filename*.

Notes:

1. To avoid having to enter the SPOOL command each time you start your system, you can include it in the STARTUP.CMD file, a batch file that begins processes automatically when you start OS/2.

2. Refer to the *OS/2 Command Reference* for further information on the SPOOL command.

Adding a Queue Driver

The OS/2 program provides the PMPRINT queue driver to link the queue to the correct printer. If you would like to add a different queue driver, follow these steps:

Note: After you add a queue driver, you must associate it with a queue by selecting the **Queues** choice from the Setup pull-down of the Print Manager.

1. From the Control Panel, click on **Installation**, or select it and press the Enter key.

2. To display the Add Queue Driver window, click on **Add queue driver**, or select it and press the Enter key.

3. Insert the diskette containing the queue driver into drive A.

   If your queue driver is in a directory on your fixed disk instead of on a diskette, type the correct path and directory name in the entry field. Do not type the file name.

4. To display a list of queue driver names, click on **Add**, or press the Enter key,

5. Click on or select the queue driver name that you want.

6. Click on **Add**, or press the Enter key.

Deleting a Queue Driver

To delete a queue driver, follow these steps:

Note: Before deleting a queue driver, remove its associations with queues by selecting the **Queues** choice from the Setup pull-down of the Print Manager.

1. From the Control Panel, click on **Installation**, or select it and press the Enter key.

2. Click on **Delete queue driver**, or select it and press the Enter key.

3. Click on or select the queue driver that you want to delete.

4. Click on **Delete**, or select it and press the Enter key.

   A panel is displayed so that you can confirm that you want to delete the selected queue driver from your system.

5. Click on **Yes**, or select it and press the Enter key.

Changing Queue Driver Settings

To change queue settings, follow these steps:

1. From the Print Manager, click on **Setup,** or select it and press the Enter key.

2. Click on **Queues**, or select it and press the Enter key.

   This displays the Queues window which lists queue names.

3. Click on or select the name of the queue that uses the queue driver that you want to change.

4. Click on **Change**, or select it and press the Enter key.

   This displays the Change Queue window, which shows the setup of the selected queue name.

5. Click on or select the name of the queue driver that you want to change.

6. Click on **Change**, or select it and press the Enter key.

   The system displays a window that contains the settings you can change in the queue driver. The settings that are displayed depend on the queue driver you select.

7. After making your selections, click on **Select**, or press the Enter key.

Running OS/2 without Print Spooling

The spooler, available by default, helps avoid errors when printing from multiple programs. If you, however, do not want to share a printing device with other programs in your system, you may want to turn the spooler off.

To prevent spooler function, follow these steps:

1. From the Print Manager, click on **Setup**, or select it and press the Enter key.

2. Click on **Spooler**, or select it and press the Enter key.

3. From the Spooler window, click on the button to turn the spooler off, or select it and press the Spacebar.

4. Click on **Set**, or press the Enter key.

5. For the change to take effect, press and hold the Ctrl, Alt, and Del keys together to restart the system.

Note: Once you have disabled spooling, you can close the Print Manager or use it for non-printing tasks such as changing queue settings.

Spooling to a Different Directory

You can spool to a different directory. You may want to do this if you partitioned your disk when you installed the system and now have insufficient room in your current partition to store spooled files. You can have files spooled to a directory in another partition on your disk by following these steps:

Note: Spooling to a directory on drive A (or any diskette) is not supported.

1. In the Spooler window, click on the **Spooler path** entry field, or select it.

2. Specify a different path than the one listed.

3. Click on **Set**, or press the Enter key.

Note: The change in directory takes place immediately, and the print jobs residing in previous directories are still active in the print queues.

Changing Printer Response Time

If your printer stops working properly, or if OS/2 does not send the file you selected to the printer correctly, the printer driver may display a message. Follow these steps to specify the number of seconds that you want to have before receiving a message if the printer driver detects such problems.

Note: Existing IBM-supplied device drivers do not use timing fields. Disregard this field unless you are working with a device driver that allows you to change printer response time.

1. From the Print Manager, click on **Setup**, or select it and press the Enter key.

2. Click on **Printers**, or select it and press the Enter key.

3. Click on or select any printer whose time you want to change.

4. Click on **Change**, or select it and press the Enter key.

5. From the Change Printers window, alter the number (in seconds) of one of the following printer timeout fields:

   - **Device not selected**, which specifies the length of time before a printer driver stops trying to send data to a printer that is off or not connected.

   - **Transmission retry**, which specifies the length of time before a printer driver detects that data did not arrive at the printer and should be sent again.

6. Click on **Ok**, or press the Enter key.

Changing Country Information

Computer systems store data as numeric values. When you need to display or print information, the system translates the numeric values into letters, numbers, symbols, and characters that you can recognize. Your system uses a table called a *code page* to achieve this.

If you divide a code page even further, it contains the definition of one or more *character sets*. For example, a character set in the U.S. code page (437) may include:

```
abcdefghijklmnopqrstuvwxyz
0123456789
~!@#$%^&*()_+,./;'[]?
```

Many countries require accented characters, which are not contained in the 437 code page, to support their languages. Certain countries, therefore, created their code pages containing character sets that allow the use of these accented characters. A character set in the Portuguese code page (860), for example, may include:

```
abcdefghijklmnopqrstuvwxyzÀÈÌÒÙ
0123456789
~!@#$%^&*()_+,./;'[]?
```

These new code pages caused difficulties because information created in a certain code page must be viewed in that code page; otherwise, some characters will not look the same as when they were entered. Menus, for example, may turn into strings of accented characters, and help text may become unreadable because alphabetic letters may display as graphics.

To solve this dilemma, IBM developed a code page that includes all the required accented characters to support many of the European languages. With this multilingual code page (850), data can be interchanged without problem between personal computers in various countries and other IBM systems. For example, suppose you create a file using characters from code page 850 and send it to someone in another country. When that file is received and viewed or printed, it is identical to your copy. This means that identical operations performed on the data in two files, such as searching and updating, can produce identical results.

However, because some existing files still use the old character sets, code page switching is necessary to provide a dual code page environment where files can be viewed in the old character set as well as in the new character set. Once the code pages have been defined on your system, you can display and switch between the prepared code pages.
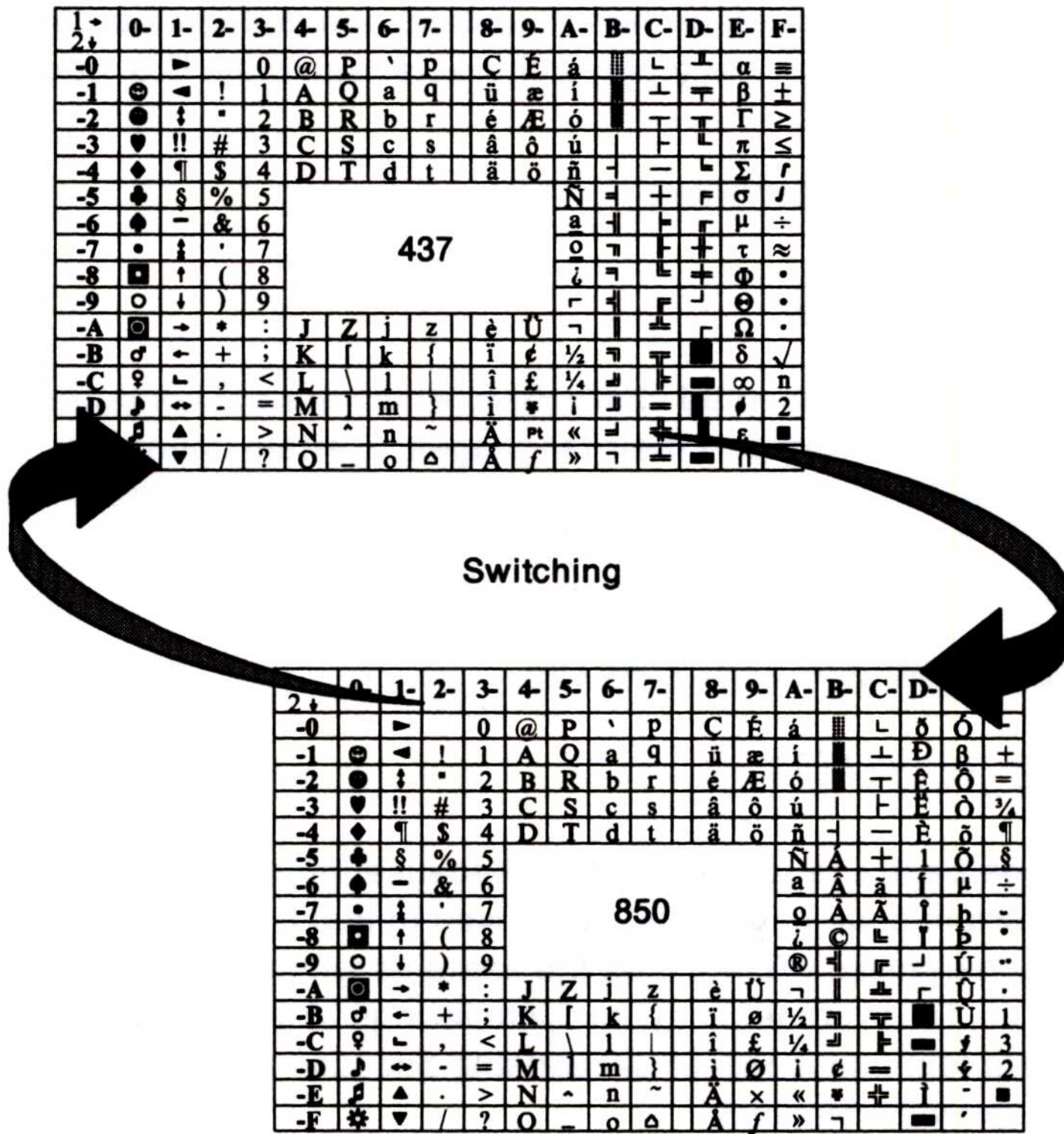
Figure 7. Code Page Switching

If your country does not require code page switching, you do not need to use it unless you want to display or print country-specific characters located on specific code pages — for example, if you come across files that have been created in a code page other than that for your country, or if your character set is not contained in the hardware code page 437. For those countries that required code page switching, the code page statements are set up in your CONFIG.SYS file during installation.

If your system has been set up for code page switching, you can switch back and forth between:

- Your **national language code page**, which contains the characters used by the language of your country and can include characters not found in any other code page. (National language code pages are standard only for personal computers; thus their use is limited.)

- The **multilingual code page**, which has characters in common with each of the national language code pages and includes all characters for the languages of most European countries.

Warning: Some PC applications written for one code page may have usability problems when using the multilingual code page. If this happens, end the program and switch out of the multilingual code page to the other code page before restarting the application.

OS/2 error messages, in most cases, use characters common to both your national language code page and the multilingual code page. When you switch code pages, OS/2 error messages remain the same.

Preparing for Code Page Switching

You must have the necessary devices and interrelated CONFIG.SYS statements to successfully switch between two code pages at run time.

Supported Devices

Code page switching is supported on the following devices:

Display Terminals

IBM Enhanced Color Display
IBM Personal System/2 family

Display Adapters

IBM Enhanced Graphics Adapter
IBM Personal System/2 Video Graphics Array
IBM Personal System/2 Display Adapter
IBM Personal System/2 8514/A

Printers

IBM 4201 Proprinter™ family except for Model 001
IBM 4202 Proprinter XL
IBM 5202 Quietwriter® III.

Note: To use code page switching on an IBM 4207 Proprinter X24 or an IBM 4208 Proprinter XL24, your printer requires a multilingual character set to be downloaded to the printer from a National Language Support (NLS) diskette. These diskettes are available in countries outside the United States.

Proprinter is a trademark of the International Business Machines Corporation.

Quietwriter is a registered trademark of the International Business Machines Corporation.

Statements for Code Page Switching

Certain statements must be added to your CONFIG.SYS file to start code page switching. These statements identify the code pages to be used and also prepare devices such as your keyboard, display terminal, and printer.

Note: Refer to "Configuring OS/2" on page 17 for further information on CONFIG.SYS.

If the printer is correctly set up for code page switching, printing jobs started in DOS mode or in the current OS/2 session — after a successful CHCP command is issued — will print in the new code page.

Incorrect, partial, or mismatched setup of statements for code page selections, country code, keyboard layout, display, or printer may cause ineffective switching between code pages.

Note: If you plan to switch code pages, it is recommended that both files and subdirectories be named using only the characters A — Z and 0 — 9 (no accented characters). This prevents file access problems when switching between code pages and their associated character capitalization rules. If you use code page 850 only, however, you will not run into any difficulties.

To properly set up code page support, the following statements must be correctly included in your CONFIG.SYS file or functional irregularities may result. Refer to the *OS/2 Command Reference* for examples on how to add the COUNTRY, CODEPAGE, and DEVINFO statements to your CONFIG.SYS file.

CODEPAGE

Identifies the two code pages to be used for code page switching. These code pages must be those supported for the country specified by the country code in the COUNTRY statement.

Note: You can change the active code page at any time using the CHCP command.

COUNTRY

Specifies the country code and the complete name of the file that contains a set of country information for each code page supported for that country.

DEVINFO

Prepares a device for code page switching. Separate DEVINFO statements are required for each device to be used for code page switching.

Keyboards: The keyboard statement specifies your keyboard layout ID (keyboard country and subcountry codes) and a file named KEYBOARD.DCP that contains a keyboard layout table for translating keystrokes into the characters of each code page supported by OS/2. Refer to the KEYB command in the *OS/2 Command Reference* for a table containing the keyboard country and subcountry codes.

Displays: The display statement specifies your display name and a file named VIOTBL.DCP that contains a video font table for displaying characters in each of the code pages supported by OS/2.

Printers: The printer statement specifies your printer name and a file with a .DCP extension that contains a printer font table for each code page supported by OS/2.

Example

For example, if you want to change from a U.S. keyboard to the French enhanced keyboard (120), follow these steps:

1. Change the COUNTRY statement in the CONFIG.SYS file to:

   COUNTRY=033

2. Add the following DEVINFO statement to the CONFIG.SYS file:

   DEVINFO=KBD,FR120,C:\OS2\KEYBOARD.DCP

3. Press the Ctrl, Alt, and Del keys together to restart your system.

Controlling Code Pages

Use the following commands to manipulate the code pages set up on your system, once you have installed code page switching.

Note: Refer to the *OS/2 Command Reference* for the syntax of the command, associated parameters, and examples.

CHCP

Displays or changes the current code page for the command processor CMD.EXE or COMMAND.COM. OS/2 displays an error message if you select a code page that has not been prepared for the system.

KEYB

Selects a keyboard layout to replace the current keyboard layout.

GRAFTABL

Allows additional characters from a language code page to be displayed when using display adapters in graphics mode.

Chapter 3. System Performance

Each person has different computer requirements, so various types of programs are available in the marketplace. Depending on which programs you choose to run, each program uses varying amounts of resources from the system — memory and storage, for example. Careful planning is required to select the correct system configuration for OS/2 to satisfy both short-term and long-term requirements.

This chapter describes OS/2 functions directly related to improving system performance. Background information concerning OS/2 memory management and file usage is included to help you understand the reasons for the required memory and fixed disk space. In addition, the following section summarizes CONFIG.SYS statements that help maximize your system's resources. This allows you to get the most work done in the least amount of time.

Remember that system installation already set up default values in your CONFIG.SYS file. Depending on the values recommended with the information that came with your programs, you might want to change these defaults to suit your particular system configuration.

Note: Refer to the *OS/2 Command Reference* for the syntax of the command, associated parameters, and examples.

Process Management

Improving performance begins with a scheduler, which distributes information to the central processing unit (CPU). The CPU, which is responsible for processing jobs and performing system activities, has functions similar to those of an office secretary. The CPU receives information that needs to be processed the same way a secretary receives messages or letters that need to be answered.

Suppose for a minute that a secretary leaves to run an errand; the rest of the work waits until the secretary returns. Similarly, in older operating systems, the CPU must wait while an input or output (I/O) operation is running. The CPU does not continue processing until the I/O operation is complete.

Because the CPU is waiting while I/O operations take place, the concept of *multiprogramming* was devised. This concept allows the operating system and other programs to use the CPU when it is waiting for an I/O operation to complete.

In the following figure, suppose the CPU is processing job 1. The CPU may eventually stop processing to wait for an I/O operation to finish. With multiprogramming, however, the processor does not have time to wait. Instead, it switches to job 2 (next in line) and con- tinues processing. When it is job 2's turn to wait, the CPU moves on to another job and eventually continues processing job 1. This proc- essing sequence, where the scheduler allocates a specific amount of time to each program, is called *round-robin scheduling*.
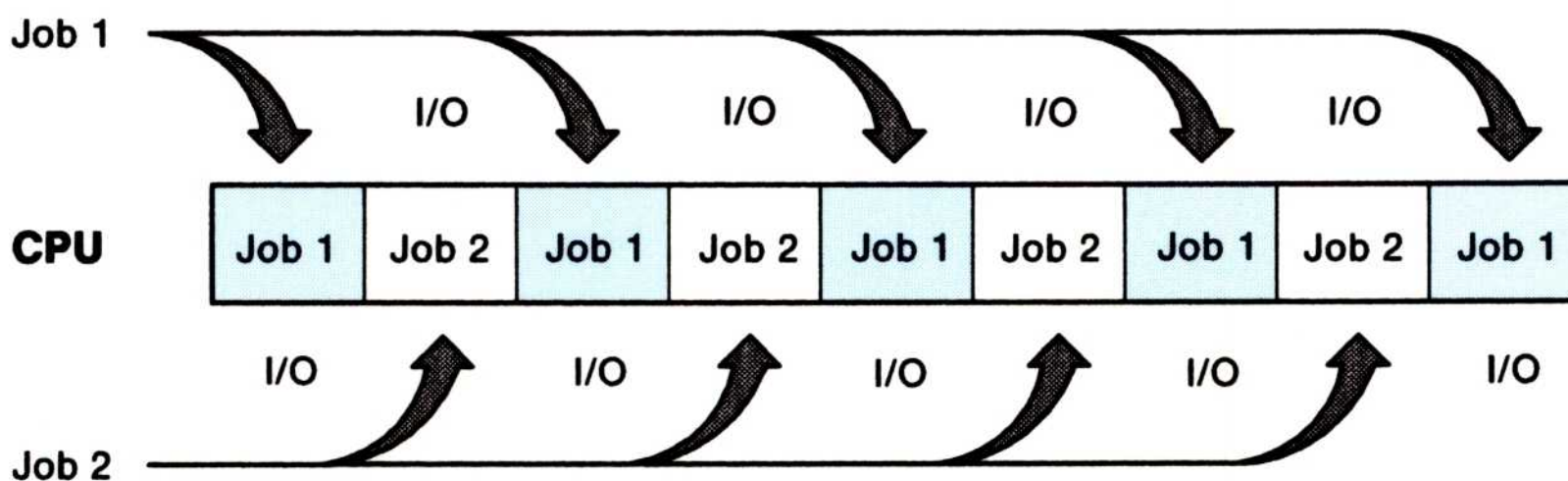
Figure 8. Multiprogramming

Time sharing assists multiprogramming by concentrating on how fast the CPU completes jobs instead of how effectively the OS/2 program uses processor time. With multiprogramming, there are many jobs wanting access to the processor, competing for its time. With time sharing, each job receives a series of small, equal shares of time.

These *time slices* are designated intervals of time allocated for processing a thread. The continuous allocation of time slices to competing jobs ensures that all jobs get an opportunity to use the CPU. In this way, no single job can monopolize the CPU when it is not performing I/O operations.

The OS/2 program uses a scheduler to handle the distribution of time slices. This gives control to a thread according to its priority class and makes sure that threads of equal priority are given equal chances to process. When a time slice elapses, or when the active thread starts an operation for which it must wait (such as I/O), control is passed to another thread according to priorities of waiting threads. At any time, if an active task waits for an event, such as I/O, it loses its time slice until that event occurs. The CPU then reactivates the task after the program completes its I/O operation.

For example, suppose job A receives a time slice from the scheduler and runs for a few milliseconds. When job A's time slice is over, the CPU picks up another program to process, even though job A has not completed its processing. The CPU eventually gets back to job A to process another time slice, and so on. This appears as though the CPU is processing several of your programs at the same time. In actuality, however, the processor can only work on a single task at a time.

The following are CONFIG.SYS statements for process management. To control the OS/2 operating environment and multitasking capabilities, you may regulate the following CONFIG.SYS statements.

Specifying the Number of Threads (THREADS)

The THREADS statement determines the maximum number of independent actions, known as threads, that can be active in the system at any time. The default value includes an allowance for system threads (depending on your system configuration) to be used for OS/2 itself and an allowance for programs. You must account here for any system extensions, additional applications, or complex operating environments that require threads.

Allocating CPU Time (TIMESLICE)

The TIMESLICE statement specifies in milliseconds the amount of CPU time allocated to each thread. If you only specify one value with TIMESLICE, OS/2 sets both the minimum and maximum TIMESLICE values to that value. The *minimum* value is the first number in the statement. The OS/2 program uses this value for special situations and it should generally be allowed to default. The *maximum* value is the optional second number in the statement. This number is the amount of time used for round-robin scheduling among threads of equal priority.

Setting Thread Priority (PRIORITY)

The PRIORITY statement selects priority calculation in scheduling regular class threads. OS/2 has a concept of priority classes and priority levels. OS/2 dynamically varies the priorities of regular class threads. This dynamic change in priority is calculated from a base level, which may be changed by a system call. The priority of a process can be either absolute or dynamic. The dynamic priority is the default and means that the system continually recalculates the priority of a process.

Setting Maximum Wait Time (MAXWAIT)

The MAXWAIT statement specifies the maximum time in seconds (from 1 to 255) that a thread must wait to get the processor resources before having its priority boosted. In other words, it is the number of seconds that pass before the priority of a process gets increased because of lack of processor time. When the scheduler denies a regular class thread the processor for a defined number of seconds, it receives a temporary boost in priority for a minimum time slice.

Memory Management

Memory management dynamically allocates and releases memory as needed and collects fragmented memory as appropriate.

Virtual Memory and Segment Swapping

By using *virtual memory*, operating systems give computers the illusion of being larger than they are in reality. It allows more program code and data to be concurrently active than the amount of memory physically installed on your computer.

This is accomplished by using a memory management technique called *swapping*, where the OS/2 program attempts to keep in physical memory only those portions of a program that are currently being used. Inactive portions of a program are placed in a *swap file* where they remain until they are needed in physical memory. This method allows OS/2 to maintain a greater number of programs in an active state than would be possible if a non-virtual memory system were being used.

Random access memory (RAM), commonly called physical memory, temporarily stores data while the system is still processing the information. By swapping RAM in OS/2 mode, it is possible to store and start programs that use up to 16MB of address space.

Segment swapping becomes necessary when you start a combination of programs and data which together require more memory than the amount physically installed on your system. For example, a program could process a large spreadsheet and have all of the contents resident in memory at once. The user would not have to split the spreadsheet into smaller segments. Since the entire spreadsheet is resident, recalculation can be faster.

By allowing *memory overcommitment*, you can execute more programs than will fit into available memory at the same time. Your virtual operating system divides the information from your files and programs into sections and stores the sections on the fixed disk. OS/2 can then load active segments that are needed instead of having to load an entire program into memory at once. As programs require more sections, the OS/2 program swaps segments to the fixed disk to free physical memory space for additional programs. In this way, more program code and data may be concurrently active than the available physical memory can contain.

For swapping to occur, the system requires enough free disk storage to contain the maximum number of segments that OS/2 will swap to disk during an OS/2 session. The size of the swap file at system installation is 640KB. If the system determines that programs require additional space, it is dynamically allocated as needed.

The OS/2 program might swap memory segments that are not currently active to your fixed disk to make one large area of unused contiguous memory. An algorithm decides which segments to swap to disk, swapping the least recently used segments first.

OS/2 does not swap segments that programs use frequently, because it would probably have to swap the segments back to physical memory, slowing down performance. OS/2 might also discard memory segments out of RAM instead of swapping them out. This is as long as you did not change the segment since you first loaded it from disk. When the discarded segment is later referenced, a fresh copy is read from the disk.

The following pages show CONFIG.SYS statements for memory and storage management provided by OS/2. Refer to the *OS/2 Command Reference* for further information on command parameters.

Determining Disk Buffers (BUFFERS)

Buffering, which deals with the timeliness of I/O devices, is one memory management method. The disk buffer is a 512-byte portion of storage that the OS/2 program uses to hold I/O information temporarily. It assists in helping the I/O device and the CPU to work at the same time.

By using the disk buffer, OS/2 can read and write blocks of information. Once read into the buffer, the CPU is ready to process the information. While the information is in the buffer or being processed by the CPU, the input device can begin reading new pieces of information so the processor does not have to wait unnecessarily to process information and program instructions. The CPU can complete its operation, because the next block of information to process is already in the buffer area.

Some other situations might also occur. For example, suppose the CPU completes its work and there is no information in the buffer to process. The CPU must wait until the next block of information is read into memory by the input device. Similarly, if the input device reads information into the buffer before the CPU has time to process the records, the buffer might reach its capacity and have to wait until the CPU accepts more information to process.

In your CONFIG.SYS file, there is a BUFFERS statement that determines the number of disk buffers that the OS/2 program will keep in memory. Depending on how many programs you work with at a given time, you may want to experiment with this number to maximize performance on your system.

If you run many programs under OS/2, you can increase the speed of your system by increasing the value specified for BUFFERS (for example, BUFFERS=70). However, remember that when you increase the number of disk buffers, you decrease your available memory by 512 bytes for each buffer specified. Additional buffers may cause some programs to run more slowly because there is less memory available for the program to keep information. This can result in more frequent memory swapping, which will slow down performance (see page 67).

Allocating Storage Blocks (DISKCACHE)

Another method of improving system performance is *disk caching*, which allows a portion of the computer's system storage to be used as an additional fixed disk buffer. Disk caching assists the buffer area by holding the most frequently used information, thus enabling the CPU to continue processing with fewer interruptions.

For example, suppose that the input device is operating slower than the CPU. Without disk caching, the processor slowly empties the buffers of information and eventually stops processing. The CPU is delayed until the input device supplies additional information. In this situation, the performance of the input device, which should rely on the speed of the CPU, is seemingly independent of CPU processing.

The disk cache speeds up computer operations that read from the fixed disk by keeping frequently accessed information in a cache buffer. When a program requests fixed disk information that is already in the cache buffer, the disk cache sends the information directly to the program. This method of accessing information is much faster than if the information has to be read from the disk each time.

In OS/2 mode, you can allocate the number of blocks of storage for control information and use by specifying the DISKCACHE statement. Although specifying this statement increases the speed of your system, when you increase the size of your disk cache you also decrease the size of available storage. For this reason, you may want to experiment with the number of blocks you specify to get maximum performance.

Setting Swapping and Segment Motion (MEMMAN)

The MEMMAN statement allows or prevents swapping and segment motion in your system. The first parameter, SWAP or NOSWAP, specifies whether you want the system to swap memory segments to disk. If you want to run several large programs at the same time, or if you have one program that is too large for available memory, set this parameter to SWAP.

The second parameter, MOVE or NOMOVE, specifies whether memory data segments in your system are allowed to move. SWAP and MOVE are the system defaults. However, where timing is critical (for example, when process control tasks are being performed), it is a good idea to prevent all swapping and segment motion in your system. By specifying the NOMOVE parameter, you can prevent storage compaction; use NOSWAP to prevent swapping. When you turn off swapping, the OS/2 program requires more physical memory to restart your system; 5MB of memory is required.

Specifying the Swap File (SWAPPATH)

With advanced memory management techniques, the OS/2 program attempts to keep only those portions of a program that are actually active in physical memory. Inactive portions of a program are placed in a special swap file where they remain until they are needed in physical memory.

The SWAPPATH statement specifies the size and location of the swap area on disk. Using a swap file allows OS/2 to maintain a greater number of programs in an active state than would be possible to fit into physical memory if a non-virtual memory system was being used.

If you use the MEMMAN command to permit swapping, the segments being swapped from running programs go to a temporary file named SWAPPER.DAT. This file grows in size as you start or stop programs and load, modify, or save data files. This file can sometimes be hundreds of thousands of bytes in size, an indication that your system may need additional memory to improve performance. The swap file needs to be at least 512KB in size, so you must have at least 512KB of available storage on the swapping drive. The size of the swap file increases as OS/2 overcommits storage.

The default SWAPPATH statement is

SWAPPATH=C:\OS2\SYSTEM 512

but you could change it to reside in another directory or in a logical drive (SWAPPATH = D) on a partitioned fixed disk if you are concerned about the swap file using all available disk space. Having the swap file reside in a separate partition might also improve swapping performance.

The value of 512 at the end of the SWAPPATH statement is a parameter called *minfree*, which regulates the amount of free space that the swap path leaves for other program use. To ensure there will always be some free space on the fixed disk, you can use *minfree*. This will limit the growth of the swap file and make sure that you will always have at least the minimum *minfree* free space on the fixed disk.

Selecting the Operating Mode (PROTECTONLY)
The OS/2 program requires this statement in the CONFIG.SYS file. The PROTECTONLY = YES statement allows memory under 640KB, which is normally used for DOS programs, to be available for OS/2 programs. When PROTECTONLY = YES, you cannot run programs in DOS (real) mode. If you later decide that you want to run DOS programs in the lower 640KB of memory, specify PROTECTONLY = NO. This allows you to use both DOS and OS/2 mode programs.

Setting DOS Mode Size (RMSIZE)
If you decided to specify PROTECTONLY = NO, you can further reduce the size of the DOS environment by specifying RMSIZE. This allows you to make the size of the DOS environment smaller than the maximum amount available if all the remaining memory below 640KB were used for real mode.

This statement specifies the highest storage address allowed for a program in DOS mode. It is important to note that if you specify a number that exceeds the amount of memory allowable for your system configuration, the system sends you an error message. This message states that the value was not acceptable and ignores the statement. If you do not specify a RMSIZE statement in your CONFIG.SYS file, the default depends on the total memory installed. If this total is 640KB or less, then the default size is the total memory minus the minimum required for protect mode operations. If this total

is 1024KB or greater, the default size is the amount of memory installed below 1024KB, either 512KB or 640KB.

This is the largest usable size for DOS mode that your system can operate. If you enter a size larger than your system is capable of operating, the system prints out an error message during startup and automatically calculates the largest default value for your system.

High Performance File System Support

The high performance file system (HPFS) is an installable file system (IFS) designed to provide better performance than the existing file allocation table (FAT) based file system. HPFS is designed to provide extremely fast access to very large disk volumes. OS/2 also supports the coexistence of multiple, active file systems on a single personal computer, with the capability of multiple and different storage devices.

If you select HPFS as your standard file system, the OS/2 installation procedure inserts an IFS statement as the first entry in the CONFIG.SYS file.

System installation automatically sets up the HPFS cache for you if you format the primary partition with HPFS (using the OS2.INI file). If you format the primary partition for FAT instead, and then later want to format another partition with HPFS, you need to add an IFS statement to the CONFIG.SYS file together with:

```
RUN=C:\OS2\CACHE.EXE /LAZY:ON
```

Keep in mind that these statements are device-dependent and must be included before any DEVICE statements.

Note: Refer to the *OS/2 Command Reference* for further information on IFS, CACHE, and RUN statements.

Important:
If you set the /LAZY parameter to ON, always use the Shutdown option in the Desktop pull-down of the Desktop Manager before turning off your system. Failure to do so will cause loss of data if the HPFS cache buffers have not been flushed out to the disk. For safety, and if performance is not a concern, use the /LAZY:OFF option, which causes a write through the cache to the disk.

If you install HPFS on your system, your system files are replaced. If you want to restart DOS from diskette, the data files are no longer accessible.

Both the FAT-based file system and installable file systems support standard naming conventions and the existing logical file and directory structure. Features of HPFS include:

- File names up to 254 characters in length

- Fast access to very large disk volumes

- Strategic allocation of directory structures

- Extended attribute support

- Caching of directories, data, and file system structures

- Processing of metacharacters that are generally used for displaying and printing graphics.

- Large file support.

HPFS Caching

The system installation program sets up caching for the primary partition through the OS2.INI file. The HPFS file system manages a cache of memory divided into blocks of 2KB. Data that is read from and written to the disk is transferred through this cache so that it can be used in satisfying future requests. This cache is separate from the BUFFERS and DISKCACHE commands discussed earlier in this chapter.

When a user requests data that is not present in the cache, HPFS selects the least recently used (LRU) block, writes the data within that block to disk if necessary, and then fills the block with the requested data. This significantly increases the chances that often-used data will be found in memory, thus saving the expense of a disk read operation.

In addition, the file system takes advantage of its knowledge of cache block contents. Data, not expected to be reused soon by the file system, is placed in cache blocks marked for immediate reuse. In most cases, when a write request is received, it is not necessary to write the data immediately to disk. HPFS takes such data and copies it to the block cache without actually performing the disk-write operation. Once the data is in the cache, it is written to disk as a background activity.

Also, because these *lazy-writes* are performed during disk idle time, incoming requests need not wait for large disk writes to complete. This optimizes overall system throughput and response time for all requesting processes.

HPFS also contains additional levels of caching that speed up access to directories. HPFS uses this cache to quickly get to the directory on which the request is being made.

Note: Although the file system owns the cache memory, it is the cache driver which manages lazy-writing of cache blocks. If the cache driver is not loaded, no lazy-writing is performed, and all write requests are written to the disk driver. Refer to the *OS/2 Command Reference* for further information on the CACHE driver.

Extended Attribute Support

In this version of OS/2, a file system can support additional information about files and directories. This information is called extended attributes. Through extended attributes, a program can attach information to a file system object (files or directories) describing the object to another program, to the operating system, or to the file system driver (FSD) managing that object.

For example, the name of the file's originator can be stored on a file object. Other uses include categorizing file objects such as icons or bit maps, and describing formats of data in the file object such as a data record.

A file object may have a list of extended attributes associated with it. These are not part of a file object's data and are managed by the file system that manages objects. An extended attribute must have a name and a value; the name is restricted to the same character set as a file name.

The maximum size of an extended attribute is 64KB and the value is arbitrary data. However, a standard set of conventions for extended attribute use is provided so that data is more easily communicated between programs. You can also search for files based on their attributes by using the Search option of the File pull-down window of the File Manager.

Note: There is a possibility of losing extended attributes if the file is rewritten or copied by a previous version of OS/2, DOS, or an application written before this version of OS/2.

Chapter 4. OS/2 Command Symbols

OS/2 provides symbols which act as command operators to help you effectively communicate between processes. This chapter describes the functions of these command symbols and demonstrates how they might be advantageous to you.

Symbols Interpreted as Command Operators

The following symbols are interpreted by OS/2 command processors as command operators:

Symbol	Mode	Function
>	OS/2 DOS	Redirects output.
>>	OS/2 DOS	Appends redirected output to existing data.
<	OS/2 DOS	Redirects input.
¦	OS/2 DOS	Pipes output.
&&	OS/2	Allows a command to run only if the command that precedes it is successful.
¦¦	OS/2	Allows a command to run only if the command that precedes it fails.
&	OS/2	Separates commands.
( )	OS/2	Groups commands.
^	OS/2	Allows input of command symbols as text.

Using Redirection Symbols >, > >, <

Suppose you want to print a list of all the files in a directory named ACCOUNTS. You know that if you enter the DIR command while in the ACCOUNTS directory, the names of the files in that directory display on your screen. How can you redirect this listing to your printer or to another file?

OS/2 provides redirection symbols to select devices other than the keyboard and screen. Refer to the following examples where DIR is used as the input source. Keep in mind that the results of these examples come from the current directory on the current drive.

Warning: When using this method of providing input to a program, be sure that *all* the program's input is in the file. If the program attempts to obtain more input after it reaches end-of-file, OS/2 is unable to supply the input and stops processing.

If you want to:	Enter:
Redirect the output of DIR to your printer	DIR >PRN
Redirect the output of DIR to a file named LISTING, creating a file named LISTING if one does not exist, or opening and writing over the file if it exists	DIR >LISTING
Redirect the output of DIR to a file named LISTING, creating a file named LISTING if one does not exist, or adding the results of DIR to the end of the file if it exists	DIR >>LISTING
Have an application named PROGRAM receive input from a file named DATA.IN rather than from the keyboard	PROGRAM <DATA.IN

Redirection Sequences Using Numbers (0-9)

In OS/2 mode, before a command is issued, its input and output can be redirected to files or other devices using special sequences interpreted by the OS/2 command processor, CMD.EXE. The numbers in these sequences are used as internal file IDs (handles) by CMD.EXE and by OS/2 programs. When used in redirection sequences with a command, input or output from or to an internal file ID is received or obtained from the newly substituted device or file.

Any internal file ID with a single digit number can be redirected. Redirection takes place even if the internal file ID is currently being used by another program. This includes the following three standard internal file IDs (shown below with their numbers) and handles 3 through 9, which are not normally used by CMD.EXE or programs. At the completion of the command containing the redirection sequence, the original handle condition is restored.

Number	Meaning
0	Standard input
1	Standard output
2	Standard error

Note: A program's input and output can be redirected only if the program is reading from standard input and writing to standard output, including standard error. If the program or command is reading directly from the keyboard and writing directly to the screen, then its I/O cannot be redirected.

The following redirection sequences may appear anywhere in a simple command or within the body of one of the special commands IF, FOR, or DETACH. Redirection is performed in the order it appears on the command line; therefore, the order in which you enter the commands is very important. Substitution of both environment and batch file variables occurs before using *name* (a valid file name or device) or *n* (numbers 0 through 9) in performing the actual redirection.

Sequence	Interpretation by CMD.EXE
<_name_	Use file _name_ as standard input (internal file ID 0). If the file doesn't exist, an error occurs.
>_name_	Use file or device _name_ as standard output (internal file ID 1). If the file doesn't exist, it is created; otherwise, it is truncated to zero length.
>>_name_	Use file _name_ as standard output (internal file ID 1). If the files exists, output is appended to the end of the file; otherwise, the file is created.
<&_n_	Standard input (internal file ID 0) is duplicated from the internal file ID _n_.
>&_n_	Standard output (internal file ID 1) is duplicated from the internal file ID _n_.
	Note: If any of the above sequences is preceded by _n_, then the internal file ID redirected is the one specified by the first _n_ instead of the default 0 or 1, as shown in the next two examples.
2>&1	Standard error (internal file ID 2) becomes a duplicate of standard output (internal file ID 1). Any output written by a process to internal file ID 2 has the same effect as if it had been sent to internal file ID 1.
4>_name_	Internal file ID 4 becomes the file specified by _name_. Any output written by a process to internal file ID 4 is sent instead to this file.

Note that blanks may appear only between the redirection symbol and its related file name, or between the & character and its related number.

Also, a leading, or first, file-ID number must exist in a redirection statement so that OS/2 interprets the number as the start of a new redirection statement instead of part of the text in a file name. To be associated with redirection, the number must be the first character on a line, or it must be immediately preceded by either a blank or one of the following delimiters:

& ¦ : , ( ) = "

Otherwise, the number is interpreted by CMD.EXE as part of the previous redirection sequence. The redirection symbol following the number is then interpreted as the beginning of the redirection sequence. Redirection commands cannot be enclosed in quotes.

For the following examples, assume that the programs use standard input, standard output, and standard error.

If you want to:	Enter:
Have DISKCOPY's standard output directed to a file named OLOG and any error messages sent to a file named ELOG	`DISKCOPY A: B: >OLOG 2>ELOG`
Display the word HELLO on your display and append standard error output to a file named OUTFILE (If OUTFILE does not exist, it is created.)	`ECHO "HELLO"2>>OUTFILE`

Direct all output and error messages from the DIR command to a file named FILELOG	`(DIR *.* >FILELOG)2>&1`
	
	or
	
	`DIR *.* >FILELOG 2>&1`

Send the output from the COBOL compiler to the OUT file and send the error messages to the ERROR file (Not specifying a number by the > symbol is equal to having the number 1 by it.)	`COBOL PROG.COB >OUT;2>ERROR`

Process the APP program noninteractively, send the output to a file named DATALOG, and have any output written to internal file ID 2 (standard error) redirected to the current target of internal file ID 1 (here, DATALOG)	`DETACH >DATALOG APP 2>&1`
	
	or
	
	`DETACH APP >DATALOG 2>&1`

Prevent All Output by a Program

It is possible to redirect any internal file ID from 0 through 9 for input or output. Files opened as input internal file IDs are opened read-only, whereas files opened as output internal file IDs are either write-only (for normal redirection) or read-write (for appending). No attempt is made by CMD.EXE to detect standard output internal file IDs redirected as input or vice versa. Thus, by redirecting both standard output and standard error as input internal file IDs, it is possible to end with no output at all. For example:

command parameters 1<INFILE 2<INFILE

All output can also be prevented by redirecting standard output and standard error to NUL. For example, to run a program named PROGRAM.EXE without collecting its output and error messages, enter:

PROGRAM.EXE 1>NUL 2>NUL

or

PROGRAM.EXE 1>NUL 2>&1

Echoing of Redirection Statements

Because the two default internal file IDs can be redirected (internal file ID 0 for standard input and internal file ID 1 for standard output), it is necessary to visually determine which internal file IDs are being manipulated. For this reason, all statements containing redirection sequences are *echoed* (displayed on the screen) by CMD.EXE in a format where the internal file ID number precedes the redirection symbol. For example:

Running:	Echoes:
DIR >DIRFILE	DIR 1>DIRFILE
MORE <DIRFILE	MORE 0<DIRFILE

Filtering Information — FIND, MORE, and SORT

You can also redirect the input of a file by *filtering* information. A filter reads information from the standard input device, changes the information in some way, and writes the result to the standard output device.

There are three OS/2 commands that act as filters:

FIND   Searches files for occurrences of specified lines of text, locates the lines, and sends the specified lines to the output device.

MORE   Receives input and then displays it on an output device, one screen at a time (useful for long files). After each screen, OS/2 pauses with the message

--More--

until you press the Enter key to continue.

SORT   Receives input from an input device, sorts the information (by letter or number), and writes the changed information to an output device.

These filter commands only work with ASCII (readable) text files. (ASCII is the standard code used for information interchange among data processing systems, data communication systems, and associated equipment.)

Note: Refer to the *OS/2 Command Reference* for the syntax diagrams, associated parameters, and examples of the FIND, MORE, and SORT commands.

Piping Output to Another Command ¦

OS/2 has a feature that allows the output of one program, as it would normally display on the screen, to be used as the input to another program without re-entering the information on the keyboard. *Piping*, as this is called, is symbolized by a vertical bar (¦). Pipes are a fixed length with the maximum that any pipe can hold being 64KB. Here are some examples using the piping symbol with OS/2 filters and redirection symbols:

If you want to:	**Enter:**
Display the names of all the files in the current directory of drive A that contain the word IBM	DIR A: ¦ FIND "IBM"
Display a sorted directory listing on your standard output device	DIR ¦ SORT
Place the sorted directory in a file named ABCLIST	DIR ¦ SORT > ABCLIST
Display the contents of a file named ABCLIST one screen at a time, using the piping symbol (¦)	TYPE ABCLIST ¦ MORE
Pipe the output of DIR to SORT, sort the listing starting with the 25th column, and send the output to your screen	DIR ¦ SORT /+24
Cause the output listing from DIR to be used by SORT as its input, redirect the output to your screen, and pause after each screen with the message	DIR ¦ SORT ¦ MORE

--More--

Note: Depending on the code page or keyboard you are using for your particular country, the piping symbol (ASCII character 124) may be shown as a solid vertical bar (|) or a split vertical bar (¦). The appearance of the symbol on your screen may not match the symbol engraved on the key.

Processing Commands Conditionally &&, ¦¦

There are special symbols that allow you to make the processing of commands dependent on the success or failure of other commands.

The && symbol allows a command to run only if the one that precedes it is successful. That is, if the command on the left side of the AND operator (&&) is successful, the command on the right side is performed. For example:

If you enter:

DIR MEMO && TYPE MEMO

OS/2 displays:

the contents of a file named MEMO on the screen (TYPE MEMO) only if it exists in your current directory.

The ¦¦ symbol allows a command to run only if the command that precedes it fails. That is, if the command on the left side of the OR operator (¦¦) is not successful, the command on the right side is performed. For example:

If you enter:

TYPE MEMO ¦¦ TYPE \ABC\MEMO

OS/2 displays:

the contents of the MEMO file in the current directory. If it does not exist there, OS/2 looks for a file named MEMO in the ABC subdirectory to process TYPE \ABC\MEMO.

Separating and Grouping Commands &, ()

The & command separator symbol processes commands on both sides of the symbol, from left to right. It produces the output from both commands after the second command is run. For example:

If you enter:

```
DIR C:*.BAT & DIR D:*.CMD
```

OS/2 displays:

the directory of files with an extension of .BAT in drive C followed by the directory of files with an extension of .CMD in drive D.

The ( ) grouping symbol gathers internal commands, thereby ensuring that the commands are run in the proper order. For example:

If you enter:

```
DIR A && TYPE A ¦ SORT > AZ
```
or
```
DIR A && (TYPE A ¦ SORT > AZ)
```

OS/2 sorts:

the file and puts the results in a file named AZ, only if a file named A exists.

```
(DIR A && TYPE A) ¦ SORT > AZ
```

the directory entry for A and the contents of the A file.

If you use parentheses and you forget to enter the right parenthesis, OS/2 displays the message:

More?

You have two choices: either finish the command with the missing closing parenthesis and press the Enter key, or press the Ctrl and Break keys together to cancel the command.

Allowing Symbols to Be Input as Text   ^

The escape symbol ^ allows input of command symbols as text.
When used outside quoted strings during command line input, the ^
symbol causes the next character following it to be treated as normal
text. The following is an example of its effect when you use it with the
ECHO batch command.

Note: A *string* is a sequence of elements of the same nature, such as
characters in text.

If you enter: **ECHO displays:**
ECHO HELLO ^>FILE   HELLO >FILE

If you want the escape symbol (^) to display, enter it 3 times (^^^) as
follows:

If you enter: **ECHO displays:**
ECHO HELLO ^^^>FILE   HELLO ^>FILE

The first escape symbol tells the second escape symbol to print and
the third escape symbol tells the redirect output symbol to print.
Keep in mind that only in the above sample would you enter the ^
three times, normally you would only need to enter it twice.

Chapter 5. Problem Determination

OS/2 has several utilities to ensure better reliability, availability, and serviceability (RAS). IBM provides these utilities to help in gathering information to isolate and correct system problems. This chapter outlines the utilities available for your personal use and also discusses those intended for use with help from your IBM service coordinator.

Utilities Available to You

There are certain utilities available to you that do not require assistance from the service coordinator.

HELP - Providing System Help

HELP is one such utility, providing a help screen that tells you how to:

- Switch to the next session or to the Task List
- Exit the current OS/2 session
- Find information in the *OS/2 Command Reference*
- Get additional help on error and warning messages.

In addition, HELP displays a help line as a part of the command prompt. This reminds you which keys to press to return to the Presentation Manager. It also requests additional information related to warning and error messages.

AUTOFAIL - Displaying Error Information

This CONFIG.SYS statement allows the OS/2 program to display information about error conditions such as hard errors. A hard error is an error condition that requires you to reconfigure the system or remove the source of the error before the system can resume reliable operation.

The system default is AUTOFAIL=NO, which causes a pop-up window to appear that informs you of an error condition. The YES parameter causes the appropriate error code to appear rather than a pop-up window.

PSTAT - Displaying Process Status Information

PSTAT displays process status information such as current processes and threads, system semaphores, dynamic linked libraries, and shared memory. PSTAT helps you determine which threads are running in the system, along with their current status and current priorities.

This command also aids you in determining why a given thread is blocked (waiting on a system event), or why the thread's performance is slow (low priority compared to other threads.) Moreover, it displays the process ID which has been assigned from each process. The process ID can then be used as input to the TRACE utility for tracing on a per-process basis.

Diagnostic Tools for the Service Coordinator

OS/2 provides additional aids for gathering information to help isolate problems, allowing for more productive serviceability for system problems. These tools are intended for use only with help from your IBM service coordinator.

Your service coordinator is the first level of customer assistance whose role includes investigating hardware and software problems. This person, who could be either an IBM authorized dealer or the person assigned to be the coordinator for your particular organization, will contact IBM, if necessary. Please do not provide IBM with a dump or any other materials unless directed to do so by the IBM service coordinator or an IBM service representative.

Error Logging Facility

To aid in system problem diagnosis, OS/2 offers a facility for logging certain system errors. Providing information of this kind allows IBM to determine and fix problems in a timely manner. Error Logging also provides the capability to generate a formatted error log report.

LOG - Log System Events
This statement provides information used for problem determination by logging system events and placing a record of each event in the System Log file. By adding this statement in the CONFIG.SYS file, you can specify whether you want to record system events in the log file. In addition, using LOG parameters, you can specify:

- the buffer size the System Logging Facility uses to store events.
- the minimum amount of free space that must be left on the fixed disk as records are added to the System Log file.
- that new entries be added to the beginning of the file when the size of that file is reached.
- The /F parameter, which allows you to specify the name of the log file created by the Logging Facility.

SYSLOG - Start or Stop Adding Log File Entries
SYSLOG postpones or continues logging of events in the System Log file. Entering this command without a parameter causes OS/2 to display the OS/2 Log Facility Main Menu. From this menu you can display or print the log file and start or stop adding entries.

System Trace Facility
The system trace facility is used for analysis to capture a sequence of system events, function calls, or data. The record is usually produced for debugging purposes. After the trace data is captured, the system trace formatter is used to retrieve it from the system trace buffer and format the data to your display, printer, or file.

On request, the OS/2 system tracing facility records certain important events in the system and system extensions in a circular buffer.

If a system problem can be duplicated without a system crash, the TRACE OFF function allows tracing to be stopped after the problem has been recreated. This allows the state of the trace buffer to be preserved from the time the TRACE OFF command is processed.

The tracing mechanism is performance critical; therefore, no statistical processing of recorded data is performed by the tracing routines.

Records in the buffer are identified by major and minor codes (you can only specify TRACE by major code in the CONFIG.SYS file). Variable length data records are also supported and new trace events can be added as required. Some of the data which may be recorded in the circular buffer will include system events such as interrupts and task switches.

If you need to use the System Trace Facility, an IBM service representative will provide the buffer size. When the trace completes, you can use the trace formatter (TRACEFMT) to organize the data into a report. This helps you isolate causes of problems in the OS/2 system by formatting the information placed in the trace buffer by the TRACE facility.

OS/2 enhancements to the TRACE utility allow you to:

- Trace a given process or set of processes, so that you can focus on the events about the specified process without intermixing events from other processes in the system. This reduces the possibility of trace buffer overflow by minimizing the number of events which are captured. Analyzing the formatted trace data is quicker and easier since only events of the specified process are recorded and displayed.

- Trace by minor code, which allows you to specify exactly what events are to be traced. Since each major event code contains many minor codes, you often must trace events that are of no interest.

- Suspend and resume system tracing, which allows you to suspend system tracing without removing trace points.

- Clear the system trace buffer, which allows you to clear the trace buffer after viewing the contents of the trace buffer.

CREATEDD and the Stand-Alone Dump Facility

The Create Dump Diskette (CREATEDD) facility creates a diskette to be used with the Stand-Alone Dump Facility which contains the contents of storage at a specified point in time.

The Stand-Alone Dump Facility runs independently of OS/2 and provides an image of physical memory. The Stand-Alone Dump Facility puts the contents of storage onto a set of diskettes called *dump* diskettes. With the help of an IBM service representative, you can use this facility for problem determination by following a system error that did not destroy the storage resident dump code and data.

A stand-alone dump is only taken at the direction of OS/2 support personnel. This dump will normally be taken in the case of a system hang or wait state. The dump is a physical record of RAM storage at the time of the system hang condition. The diskette containing the dump is sent to an OS/2 support location for analysis.

PATCH - Apply Software Repairs

This utility should be used by those who understand the need for a patch, how to make a patch, and the effect the patch has on the operation of a program.

PATCH has prompts that guide you through inserting changes to the OS/2 system software. If you select the option to automatically apply patches shipped by IBM to make fixes to IBM-supplied code, verification is performed before the patch is applied. Verification can not be done on non-IBM-supplied patches.

If you select the option to manually apply a patch, you are asked to supply an offset where the patch is to be made. PATCH displays the contents of the location specified by the offset and allows you to enter the patch. Both the offset and the patch contents are expected to be entered in hexadecimal notation.

Recovering User and System INI Files

System settings, such as application defaults, display options, and file options, are contained in the OS2.INI startup file located in the C:\OS2 directory of your fixed disk. There is also a system file called OS2SYS.INI, which contains information about installed fonts and printer drivers. In the event that you receive a message stating that the OS2.INI file is corrupted, the OS2.INI file installed on your system must be replaced by another valid copy of the OS2.INI file.

To accomplish this, use the MAKEINI exec from your command prompt.

Added to your system during the installation process, the MAKEINI.EXE file creates a new OS2.INI file containing default information. You should recreate both the user and system INI files using MAKEINI (located in the C:\OS2 directory of your fixed disk).

Creating New INI Files

To create new user and system INI files, follow these steps:

1. Insert the *Installation* diskette in drive A; then, press and hold the Ctrl, Alt, and Del keys together to restart the system.

2. Press the Escape (Esc) key when OS/2 displays the IBM logo.

3. To change to drive C, enter:

   ```
   C:
   ```

4. To change to the OS2 subdirectory, enter:

   ```
   CD \OS2
   ```

5. To erase the current OS2.INI file, enter:

   ```
   ERASE OS2.INI
   ```

6. To erase the current OS2SYS.INI file, enter:

   ```
   ERASE OS2SYS.INI
   ```

7. To recreate a new user INI file, enter:

   ```
   MAKEINI OS2.INI INI.RC
   ```

8. To recreate a new system INI file, enter:

   ```
   MAKEINI OS2SYS.INI INISYS.RC
   ```

9. Remove the *Installation* diskette from drive A.

10. Restart the system by pressing the Ctrl, Alt, and Del keys together.

Note: You can rename the OS2.INI and OS2SYS.INI files to names of your choice as long as the same file names are reflected in the PROTSHELL statement in the CONFIG.SYS file.

Recovering the CONFIG.SYS File

The CONFIG.SYS file contains command statements that configure your system for both DOS and OS/2 modes. If CONFIG.SYS gets corrupted, you cannot restart the system or edit the file. In the event that the CONFIG.SYS file gets corrupted, follow these steps:

1. Insert the *Installation* diskette in drive A; then, press and hold the Ctrl, Alt, and Del keys together to restart the system.

2. Press the Escape (Esc) key when OS/2 displays the IBM logo.

3. To change to drive C, enter:

   C:

4. To rename the corrupted CONFIG.SYS file, enter:

   REN CONFIG.SYS CONFIG.BAD

5. To copy the CONFIG.SYS backup file to the root directory, enter:

   COPY C:\OS2\INSTALL\CONFIG.SYS C:\CONFIG.SYS

6. Remove the *Installation* diskette from drive A.

7. Restart the system by pressing the Ctrl, Alt, and Del keys together.

Note: If you updated the CONFIG.SYS file after OS/2 installation, use an editor to correct the damaged file (now named CONFIG.BAD) or, keep the current CONFIG.SYS file.

Chapter 6. DOS Mode Compatibility

Often, OS/2 is installed on personal computers that were previously running under IBM DOS. To make the transition easier for you, OS/2 provides a DOS compatibility mode which allows many existing DOS programs to run without modification.

This section addresses some of the considerations that apply either when programs need to be run in a mixed environment or when DOS programs need to be run on an OS/2 system. In addition, it lists DOS mode limitations and contains advice on changes that may be required on your system.

Compatibility with IBM DOS 4.0

When using the OS/2 program, you can choose DOS mode, which is an environment patterned after IBM Disk Operating System Version 4.0. However, some features of IBM DOS 4.0 commands and utilities are not supported or have slightly changed in the OS/2 program's DOS mode. The following listing is an overview of these differences.

ANSI.SYS
The /L parameter is not supported.

APPEND
The /X parameter is not supported.

BACKUP
This command is supported in OS/2 mode only. Along with new error levels, support has been added for long file names, meta characters, and extended attributes.

BUFFERS
The /E parameter is not supported.

CHKDSK
Displays file system type information. Along with new error levels, support has been added for long file names, meta characters, and extended attributes. Memory statistics are reported in DOS mode only. CHKDSK of an HPFS volume is supported in OS/2 mode only.

COMP

End-of-file checking has been removed. You can continue comparing if the file sizes differ. Along with new error levels, support has been added for long file names and meta characters.

COUNTRY

The statement specifying the file path of the code page is not supported.

CTTY

This command is not supported.

DISKCOMP

The /1 and /8 parameters are ignored. DISKCOMP determines the number of sides and sectors by the source diskette. DISKCOMP also skips the volume serial number during comparison.

DISKCOPY

The /1 and /8 parameters are ignored. DISKCOPY determines the number of sides by the source diskette. Also, the target diskette is an identical copy except for the volume serial number.

DISPLAY.SYS

This device driver is not supported because the function is inherent.

DOSSHELL

This command is not supported. Presentation Manager is used for the protect mode shell.

EXE2BIN

This command is not supported.

FASTOPEN

This command is not supported because the function is inherent.

FILES

This command is not supported.

FORMAT

The following are changed functions:

- The /S, /8, /B, and /1 parameters are not supported.

- Along with new error levels, a new /FS parameter has been added.

GRAPHICS

This command is not supported because the function is handled by the OS/2 spooler.

IFSFUNC

This command is not supported because the function is inherent.

INSTALL

This command is not supported.

KEYB

This command is supported in OS/2 mode only.

LABEL

The option to delete the label is not supported.

LASTDRIVE

This command is not supported.

MEM

This command is not supported.

MODE

The following are deleted functions:

- DISPLAY option: Shifting of screen, test pattern not supported.

- ASYNC option: Infinite Retry (,P) is replaced by the Extended Printer Retry option, and the Ctrl+Break function is not supported.

- Redirection of parallel to serial support is not supported in MODE because this function is now supported by the SPOOL command.

- Code page options are not supported because this function is enhanced by CODEPAGE and DEVINFO statements in the CONFIG.SYS file and the CHCP command.

- RATE option is not supported.

- DELAY option is not supported.

NLSFUNC

This command is not supported because the function is inherent.

PATCH

This command has long name support.

PRINT

The /B, /U, /M, /S, /Q, and /P parameters are not supported.

RECOVER

Displays file system type information. Along with new error levels, support has been added for long file names, meta characters, and extended attributes. RECOVER of an HPFS volume is supported in OS/2 mode only.

RESTORE

This command is supported in OS/2 mode only.   Along with new error levels, support has been added for long file names, meta characters, and extended attributes.

SELECT

This command is not supported because the function has been replaced by the installation program.

SHARE

This command is not supported because the function is inherent.

STACKS

This command is not supported.

SYS

This command is not supported as an external command.

TREE

The /A parameter is not supported.   Also, long name support has been added.

VDISK.SYS

The /E parameter is not supported.

XCOPY

Along with new error levels, support has been added for long file names, meta characters, and extended attributes.

XMAEM.SYS

This device driver is not supported.

XMA2EMS.SYS

This device driver is not supported.

Installing Copy-Protected DOS Programs

Some copy-protected DOS programs depend on the operating system or a version of the operating system for correct installation. Refer to your program documentation for specific instructions on which operating system is supported before installing or removing the program. Also, always install or remove these programs from the DOS command prompt.

Copy-protected DOS programs may already be installed on your fixed disk. Once installed, many of these programs can run in DOS mode.

Some copy-protected DOS programs rely on timing during their installation and initial startup. During installation and initial startup of these programs, make sure that no OS/2 program is running and that you do not switch out of DOS mode. This reduces the possibility of an error occurring.

Printer Spooling

OS/2 has its own print spooler to control information going to the printer from different sources. This allows OS/2 programs to send output to a printer while your DOS program is printing.

Similarly, a print job started by your DOS program may not print immediately. This could be either because an OS/2 program has a job printing, or because the DOS program print job has not completed. To force a DOS program print job to end, press the Ctrl, Alt, and PrtSc keys together.

When your DOS program prints using the OS/2 print spooler, more time may be required than when running without the print spooler. If the print spooler has several jobs to print, jobs are printed sequentially.

Note: For more information on spooling, refer to the SPOOL command in the *OS/2 Command Reference*.

Serial Device Support

Serial devices, such as printers and plotters, receive output from the asynchronous communication (COM) port in your system. In OS/2 you must select the *serial device* (COM port) support needed by your OS/2 programs when you configure your system. Refer to the DEVICE = C:\OS2\COM0x.SYS statement in the *OS/2 Command Reference* for further information and examples.

Most existing DOS mode programs, however, are not written to use OS/2 COM port support. You need to select serial device support during system installation only if you do one or more of the following:

- Set asynchronous communication modes using the MODE command

- Redirect parallel printer data (LPT1, LPT2, or LPT3) to a serial device (COM1, COM2, or COM3) using the SPOOL command

- Run OS/2 programs that perform input or output to COM1, COM2, or COM3.

Setting Asynchronous Communications Modes

In DOS mode, the MODE command is used to set asynchronous communication modes for a serial printer or plotter, just as it is in IBM DOS. For example, enter the following to set the baud rate for COM1 to 9600, the parity to NONE, the databits to 8, and the stopbits to 1:

*d:\path*MODE COM1:9600,N,8,1

Notes:

1. A DEVICE = C:\OS2\COM0x.SYS statement must be in the CONFIG.SYS file for the OS/2 program to recognize COM1 as a device name.

2. Refer to the *OS/2 Command Reference* for further information on the MODE command.

Redirecting Parallel Printer Output to a Serial Device

In OS/2, redirection of parallel printer (LPT1, LPT2, or LPT3) output to a serial device (COM1, COM2, or COM3) is supported by the SPOOL command, not by the MODE command as in IBM DOS.

The OS/2 system default is to start the spooler and direct LPT1 to LPT1, LPT2 to LPT2 and LPT3 to LPT3. The spooler can be turned off by using the Print Manager and restarting the system. Redirection can be changed by issuing the SPOOL command dynamically.

Sending Output to Serial Devices

If you have a DOS program that uses a serial printer or plotter for output and your CONFIG.SYS file contains a DEVICE = C:\OS2\COM0x.SYS statement, you may need to use the SETCOM40 command in the DOS session.

Some existing IBM DOS programs need the address of the COM port to which the serial printer or plotter is attached to send data to these devices.

SETCOM40 provides these programs with the COM port address. If you are not sure that your program needs this information, use SETCOM40 in case it does. Refer to SETCOM40 in the *OS/2 Command Reference* for further instructions.

Managing Programs Using Serial Devices

Some IBM DOS programs have not been written for a multitasking environment. OS/2 cannot always manage programs using serial devices in DOS mode. These programs go to the serial device and try to program the COM port hardware instead of first seeing if the device is in use by another program.

To use the same serial printer or plotter in the DOS session and OS/2 sessions, you must manage the DOS program's use of these devices. When running a program that uses the COM port device in DOS mode, make sure that no other OS/2 program simultaneously uses the same COM port. Let your DOS program finish printing or plotting before switching to an OS/2 session; otherwise, unpredictable results, such as intermixed output, can occur.

Note: Even though the DOS program appears to be the only program running, there can be OS/2 programs running in background OS/2 sessions. These background OS/2 programs can interfere with the foreground DOS program if both simultaneously attempt to access the COM port.

To manage the printing or plotting performed by your DOS programs:

- Make sure that the DEVICE = C:\OS2\COM0x.SYS statement is added to your CONFIG.SYS file.

- Issue the MODE command to set the COM port's asynchronous communication mode so that it correctly interfaces to your serial printer or plotter.

- If an OS/2 program (such as SPOOL) is using that COM port with a serial printer or plotter, let it finish printing or plotting before using the same COM port device from the DOS session.

When the OS/2 programs have finished printing:

1. At the DOS command prompt, enter:

   SETCOM40 COMx=ON

2. Start the DOS mode program and use the serial printer or plotter in DOS mode; then, exit the program when completed.

3. At the DOS command prompt, enter:

   SETCOM40 COMx=OFF

**Note:** Replace x with 1, 2, or 3 for the COM port being accessed.

Using a Mouse

If you indicated your mouse type during OS/2 system installation, the correct OS/2 mouse device driver is already specified in your CONFIG.SYS file. Note that only OS/2 mouse device drivers are supported.

Using Programs with Enhanced Graphics Features

For some programs using an Enhanced Graphics Adapter or IBM PS/2 8514/A display adapter, the system might not be able to restore the program's graphics screen when you switch to DOS mode and then switch back again. If you are not able to restore the program's graphics screen, do not switch from DOS mode until you have completed the task.

Chapter 7. Creating and Using Batch Files
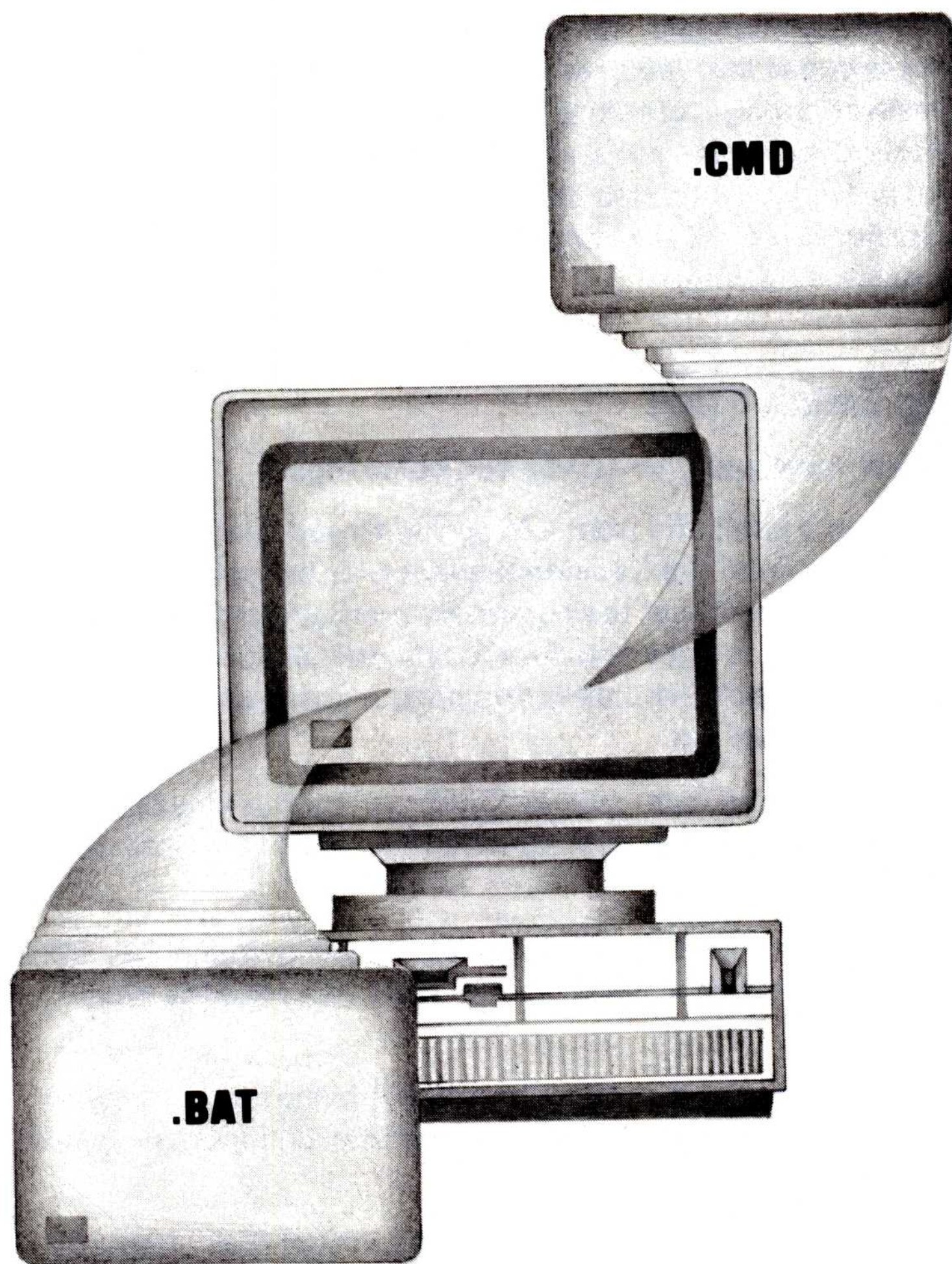
Batch files can be created for both DOS and OS/2 modes.

About Batch Files

A *batch file* is a common processing file containing several com-
mands. When you run a batch file, the OS/2 command processor
processes the commands in the batch file, one at a time. If you have
a repetitive process that requires you to enter a series of commands
at the command prompt, creating a batch file that contains these com-
mands shortens process time and reduces the possibility of typing
errors. All batch files intended for use in DOS mode have a .BAT file
name extension; batch files intended for use in OS/2 mode have a
.CMD file name extension.

Running Batch Files

Some general rules apply when you run batch files:

- You can run a batch file from a Program Group window, from the
  File Manager, from the Desktop Manager, or by typing the file
  name of the batch file at the command prompt and then pressing
  the Enter key. It is not necessary to type the .CMD or .BAT exten-
  sion to start a batch file unless you have a program with the same
  name as the batch file.

- A batch file with an extension of .CMD will not run in a DOS mode
  session, and a batch file with an extension of .BAT will not run in
  an OS/2 mode session.

- If you remove a diskette containing a processing batch file, OS/2
  prompts you to insert it again so that OS/2 can read the next
  command.

- OS/2 remembers the directory the batch file started in. There-
  fore, the commands within the batch file may change the current
  directory during processing.

- Batch files process quickly if placed on a virtual disk. Refer to
  the VDISK.SYS device driver in the *OS/2 Command Reference* for
  information on virtual disks.

Ending Batch File Processing

In DOS mode, batch file processing is similar to IBM DOS. If a *filename*.BAT is running in the DOS session and you press the Ctrl and Break keys together, the following prompt is displayed:

```
Terminate batch job (Y/N)?
```

If you enter Y in response to this prompt, the batch file ends and control is returned to the operating system. If you enter N, the batch file resumes processing.

If you press the Ctrl and Break keys together while a *filename*.CMD is running, it ends, and control is returned to the operating system.

Treating batch files that are dependent on other batch files as a single command avoids unpredictable results in the multitasking environment of the OS/2 program. If a batch file that sets the environment variables with the SETLOCAL command is interrupted, all environment values that were set are no longer in effect. If the batch file is allowed to resume processing, the results may be different from what was intended.

Special Batch Files

During system installation, an AUTOEXEC.BAT batch file is created and is placed in the root directory of the startup drive. You can change it, the way you would any other batch file, by using an editor such as the OS/2 System Editor. You can also create a batch file called STARTUP.CMD, if you want to use the OS/2 autostart facility. (Information on the autostart facility can be found on page 111.)

AUTOEXEC.BAT File

When you select the DOS mode command prompt from the Presentation Manager, the DOS mode command processor searches for a special batch file named AUTOEXEC.BAT in the root directory of the startup drive. When it finds the AUTOEXEC.BAT file, it uses the commands in the batch file to define the way it starts DOS mode and its environment. (AUTOEXEC.BAT does not run in OS/2 mode.)

The following is an example of what you might find in the AUTOEXEC.BAT file created during system installation:

```
PATH C:\OS2;C:\OS2\SYSTEM;C:\;
SET COMSPEC=C:\OS2\COMMAND.COM
APPEND=C:\OS2;C:\OS2\SYSTEM;C:\;
CALL HELP ON
```

The PATH command provides a search path to OS/2 external commands. You can add your own paths to application programs.

The SET COMSPEC= tells the operating system where it can find the COMMAND.COM DOS mode processor.

The APPEND command specifies paths to data files.

The CALL command runs a batch file that turns on the HELP bar as part of your OS/2 command prompt.

The AUTOEXEC.BAT file can also include the command PROMPT PG to set and customize the command prompt.

Note: If you include a PROMPT command after the CALL HELP ON command, it overrides the HELP prompt and the HELP bar is deactivated.

If an AUTOEXEC.BAT file exists, you are not prompted for the date and time when you start DOS mode from the Presentation Manager. If an AUTOEXEC.BAT file does not exist, you are prompted for the date and time every time you start DOS mode. Because the system uses the clock in your computer, it is not necessary to set the time and date each time you start the system or start DOS mode.

STARTUP.CMD File

You can take advantage of the OS/2 autostart facility by creating a special batch file called STARTUP.CMD. Then when you start OS/2, the first OS/2 session is started automatically and displayed on your screen. If there is no STARTUP.CMD file, the Group-Main window of the Presentation Manager is displayed.

To start programs in additional OS/2 sessions, you can use the internal command START in the STARTUP.CMD file. The following are examples of STARTUP.CMD batch files:

Example 1: Assume that you have the following commands in a STARTUP.CMD file:

```
PATH C:\;C:\OS2;C:\MISCPGM;
START CALC
START WP
START
CALENDAR
```

This is what happens when you start OS/2:

1. STARTUP.CMD starts a command processor in a windowed session that is displayed on your screen. This is where the statements in STARTUP.CMD are processed.

2. The PATH command sets the path for the first session.

3. The first START command starts a second session that runs a spreadsheet program.

4. The next START command starts a third session that runs a word processing program.

5. The last START command starts a command processor in a fourth session.

6. The last command starts the CALENDAR program in the first session.

Note: Environment variables defined in STARTUP.CMD are only active in the first session. Any other sessions started with START commands are not affected. Thus, the path specified in this example only affects programs that run in the session where the CALENDAR program is running.

Example 2: Assume that you have these commands in a STARTUP.CMD file:

```
START CALC
START WP
```

When you start the system, the sessions for the spreadsheet and word processing applications are started. The first session started by STARTUP.CMD displays the command prompt on your screen.

Example 3: When this STARTUP.CMD file runs:

```
START ENV
START CALC
START WP
EXIT
```

1. START ENV starts an environment batch file to set the environment variables.

2. The sessions for the spreadsheet and word processing applications are started.

3. The EXIT command causes the command processor started in the first session by STARTUP.CMD to end. The Group-Main window is then displayed.

A program specified with a START command can be an internal or external command, an application program, or a batch file. If the EXIT command is the last statement of a batch file, when the batch file ends, the command processor is exited and the session ends.

Guidelines for Batch Files Run By START Commands

Environment variables for a session are set by having a START command (START ENV) in a STARTUP.CMD batch file. The following is an example of a file called ENV.CMD, and shows what an environment-setting batch file might contain:

```
PATH C:\;C:\OS2;C:\MISCPGM;
DPATH C:\;C:\OS2;C:\LETTERS;C:\REPORTS;
PROMPT [EDIT]
EDIT
```

The last command in this example starts an editing application called EDIT. When you end the application, the command prompt is displayed as:

```
[EDIT]
```

When you use the START command in STARTUP.CMD to run a batch file, your batch file should not make assumptions about the environment, current drive, or current directory. For example, if the purpose of your batch file is to start a program on a specified drive within a specified directory, the complete name of the directory should be specified with the CD (change directory) command.

Assume that your batch file starts a program called FILEMAN located in the \XYZ directory of drive D. Your batch file should look like this:

```
D:
CD \XYZ          Correct Batch File
FILEMAN
```

Do not *assume* the default for the current directory is the root directory, as in this example:

```
D:
CD XYZ           Incorrect Batch File
FILEMAN
```

For more information, refer to the START command in the *OS/2 Command Reference*.

Writing Batch Files

Listed here are some commands specifically designed for general batch file operations. Any command that works on the command line can be used in an AUTOEXEC.BAT file or any other batch file. Refer to the *OS/2 Command Reference* for more information on these commands.

Warning: Because OS/2 can operate in a multitasking environment, it is possible to change, delete, or replace a batch file in one operating session while it is running in another session. While this can be considered a useful tool to some computer operators, it can cause problems and unexpected results for others if these files are altered during their operation cycle.

Command	Description
APPEND	Tells the system where to locate data files outside of the current directory when this command is added to the AUTOEXEC.BAT file.
CALL	Nests a batch file within a batch file.
ECHO	Allows or prevents the display of OS/2 commands while a batch file is running.
ENDLOCAL	Restores the drive, directory, and variables that were in effect before a SETLOCAL command was issued.
EXTPROC	Defines an external batch file processor.
FOR	Allows repetitive processing of commands within a batch file.
GOTO	Transfers batch processing to a specified label.
IF	Allows conditional processing of commands within a batch file.
PATH	Tells the system where to set the search path.
PAUSE	Suspends processing of the batch file.

Command	Description
REM	Displays remarks from within a batch file.
SET	Sets an environment variable equal to a string for later use.
SETLOCAL	Sets the drive, directory and variables that are peculiar to the current batch file.
SHIFT	Allows more than 10 replaceable parameters in a batch file.

Guidelines when Writing Batch Files

Follow these guidelines when writing batch files:

- One purpose of batch files is to simplify jobs; it is a good practice to keep the names simple too.

- A batch file name to be used in DOS mode follows the format of 1 to 8 characters followed by an extension of a maximum of 3 characters.

- OS/2 files under the High Performance File System (HPFS) can have file names of as many as, but not more than, 254 characters.

- Qualified OS/2 file names containing a drive letter and path can have as many as 259 characters.

- Blanks are not allowed in the file names.

- Keep file names simple but meaningful so that they can be identified.

- Do not name batch files with the same names as internal commands, such as COPY, ERASE, and so on.

- To prevent confusion, avoid having a program such as .COM or .EXE and a batch file with the same file name. If you do, then it is necessary to enter the file name and extension when starting the batch file. The command interpreter processes files such as these in the following order:

  - .COM
  - .EXE
  - .BAT or .CMD

    Note: For more information on naming files, see "Naming Files and Directories" in *Getting Started*.

- *Chaining* allows you to start one batch file from another when the first finishes processing. You can chain batch files by making the last command in a batch file the name of another batch file.

- You can *nest* batch files by using the CALL command. This calls a second batch file from within another batch file without ending the first batch file. Nesting allows all of the commands in the second batch file to process before returning to the first batch file.

- Any command you can enter from the command prompt can be used in a batch file.

- The following symbols and device names cannot be used as file or directory names:

    " / \ [ ] : | < > + = ; , &

    KBD$ PRN   NUL   COM1   COM2   COM3   CLOCK$

    LPT1 LPT2 LPT3 CON   SCREEN$ POINTER$ MOUSE$

The two extensions for batch files are as follows:

Mode	Extension
DOS	*filename*.BAT
OS/2	*filename*.CMD

Warning: The different batch file extensions for DOS and OS/2 do not prevent you from putting a command that works only in one mode into a batch file whose extension is for the other mode. If you want to use a command in a batch file and you are not sure if it works in both modes, refer to the command description in the *OS/2 Command Reference* before you use it.

Eliminating the Display of Batch File Commands

The ECHO command allows you to eliminate the display, or *echo*, of batch file commands that are processing in either mode. You can also eliminate, or *suppress*, echoing by typing an at symbol (@) as the first character of a batch file command. For example, if you type:

```
@CMDNAME
```

the echoing of the line the command is on is suppressed. If you enter:

```
@ECHO OFF
```

the echoing of ECHO OFF is suppressed, which in turn suppresses the lines that follow until an ECHO ON is met.

In OS/2 mode, another method can be used to suppress echoing of batch file commands: typing /Q in the syntax of the batch file command. For example:

```
BATFILE.CMD /Q
```

This command in a batch file is equivalent to having ECHO OFF as the first command of the batch file, except that no ECHO OFF message is displayed.

An ECHO ON command in the batch file causes echoing of batch file commands to resume. The /Q, or quiet parameter, can be placed anywhere in the parameter list following the batch file name.

By including a FOR command and including the at (@) symbol in the DO clause, you can suppress the echoing of the iteration of the loop.

```
FOR %I IN (*.*) DO @ECHO %I
```

This example produces a simple list of files as output.

Using Replaceable Parameters

A *replaceable parameter* (placeholder) is a parameter whose value is supplied when a batch file is started either from within another batch file or started at the command prompt. By using replaceable parameters in a batch file, you can pass information to a batch file when it is started. These parameters, named %0 − %9, are replaced by other values, such as file names or directories. The %0 parameter is always replaced by the file specification (drive, path, and file name) of the batch file being started.

For example, if a batch file named SAMPLE.BAT contains these commands:

```
COPY %1 %2
TYPE %2 | MORE
```

OS/2 sequentially replaces %0 with the file being run, and continues replacing parameters %1 and %2, through %9, with the parameters supplied when the file is started.

To run the SAMPLE.BAT file and pass values to the replaceable parameters, type the name of the batch file followed by the parameters you want sequentially substituted for %1, %2. For example:

```
SAMPLE A:DATA.OLD  C:INFO.NEW
```

The batch file name SAMPLE is substituted for %0; A:DATA.OLD is substituted for %1; and C:INFO.NEW is substituted for %2. Because the TYPE command is also in the batch file, the C:INFO.NEW file is then displayed on your screen. The result is the same as if you entered each of the commands with their parameters, as follows:

```
COPY A:DATA.OLD C:INFO.NEW
TYPE C:INFO.NEW | MORE
```

Notes:

1. If you want to specify more than 10 replaceable parameters, you may do so by using the SHIFT command.

2. If you want to use a percent sign (%) as part of a file name within a batch file, you must specify it twice. For example, to specify a file named ABC%.EXT in the batch file, type ABC%%.EXT .

Using Replaceable Parameters with Names

By using the SET command in a batch file you can use *strings* instead of numbers to define replaceable parameters. Strings are names or lines of text. The SET command sets one string in the environment equal to another string for later use in programs.

When you use a string instead of a number, you must begin and end the string with a percent sign (%). However, do not use % when you define its value with SET.

For example, if a batch file has multiple occurrences of a file name, and you want to use this batch file at different times and specify a different file name each time, you can specify the string at each place in the batch file where you want the file name. For example, if you use a string called WEEK:

```
%WEEK%
```

set the value for WEEK with a SET command that precedes the occurrences of %WEEK% in the batch file. For example:

```
SET WEEK=MONDAY.ABC
```

Then run your batch file. The batch processor substitutes each occurrence of this parameter with the value you defined with the SET command. When you want to specify a different file name, you need only change the SET command. For example:

```
SET WEEK=TUESDAY.XYZ
```

You also can specify environmental system variables as replaceable parameters in a batch file. For example, suppose you have this replaceable parameter in your batch file:

```
%PATH%
```

When the batch processor meets this parameter, the processor replaces it with the current value in the environment for PATH.

Chapter 8. Using the System Editor

The information in this chapter describes how to use the System Editor for creating and editing files. Written primarily for new computer users, this chapter leads you step-by-step through the tasks you perform when editing files.

The OS/2 1.2 System Editor replaces the OS/2 1.1 System Editor. Designed as a simple text editor, the System Editor runs in a window, is menu-driven, and capable of editing ASCII text files. Consider installing a more powerful editor if you will be doing extensive text editing or word processing tasks. Using the System Editor, you can:

- Use both the mouse and keyboard when editing files
- Receive help
- Change fonts and colors
- Use the Clipboard to transfer text between applications.

Help is available and can be viewed by pressing the F1 key or by selecting **Help** from the action bar.

This chapter describes the following System Editor basics:

- Starting an editing session
- Getting help
- Displaying drives and directories
- Saving and closing files
- Ending an editing session.

The chapter then goes on and describes:

- Adding and selecting text
- Manipulating and merging text
- Deleting and restoring text
- Finding and changing text.

The term "select" as used throughout this chapter means to choose and highlight an item. The term "start" as used throughout this chapter means to enter your selection and make it active.

Key assignments for the System Editor are described in Appendix A.

Before using the System Editor, review *Getting Started* to become familiar with how to use:

- Desktop Manager
- Presentation Manager
- File Manager
- The mouse and keyboard
- Windows
- Paths, directories and file names.

Starting an Editing Session

There are several ways to start the System Editor:

- Selecting it from Desktop Manager's Group - Main list
- Entering an E at the OS/2 Full Screen or OS/2 Window
- Selecting its program name (E.EXE) from File Manager
- Selecting an associated file from File Manager.

To start the System Editor from Desktop Manager:

- Select and start **OS/2 System Editor** from Group - Main.
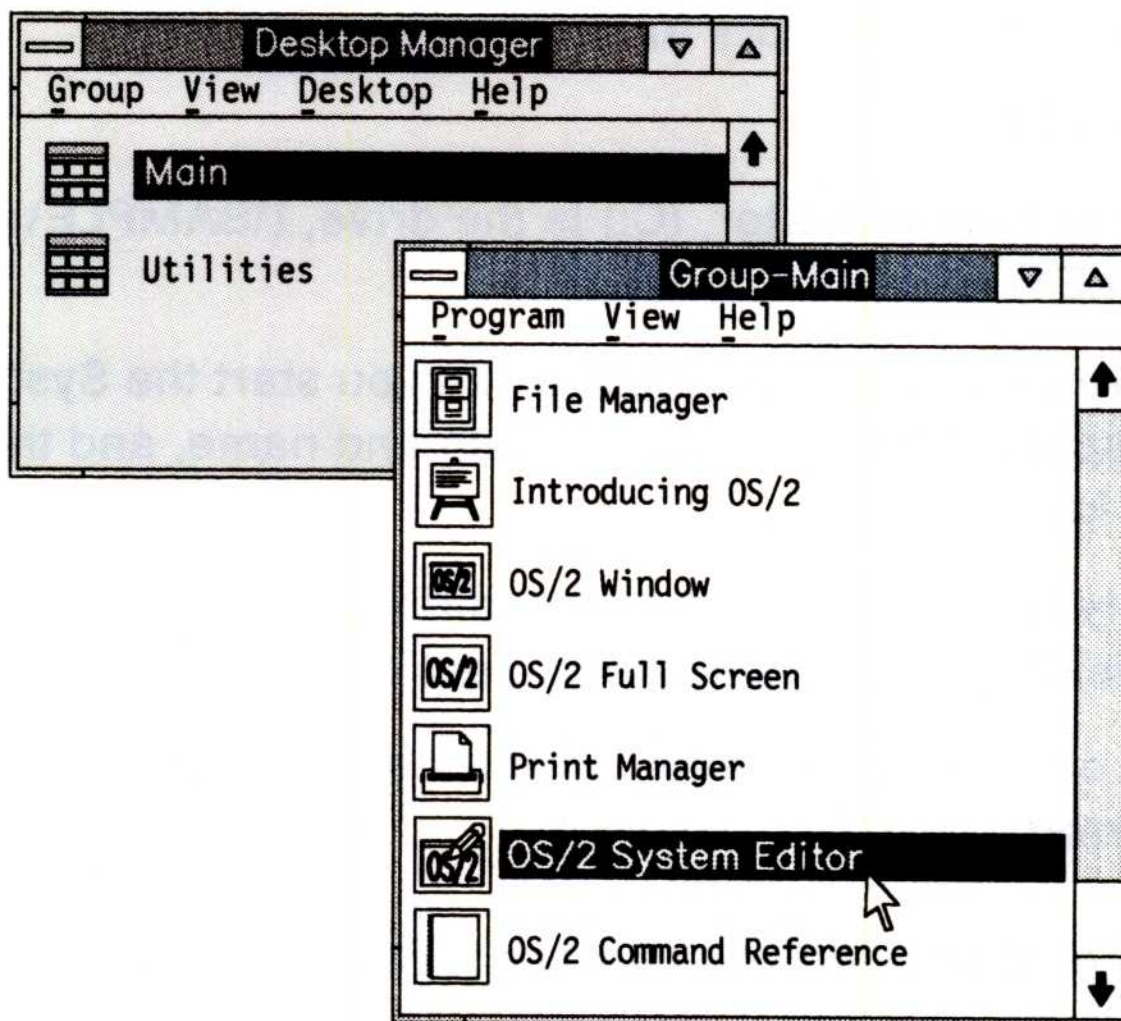
The System Editor starts.

Note: To start additional file windows, return to Group - Main and repeat the above steps.

To start the System Editor from the OS/2 Full Screen or OS/2 Window:

1. From the Group - Main window, select and start either **OS/2 Full Screen** or **OS/2 Window**.

   An OS/2 command prompt ([C:\]) is displayed.

2. Type the letter E; then press the Enter key and the System Editor starts.

   Note: When starting the System Editor from a command prompt, you can specify the file you want to edit or name the file you are about to create.

   To start and edit an existing file, for example one named **FILE1** that is on drive **C** in the directory **Sample**, you would enter:

   ```
   E C:\SAMPLE\FILE1
   ```

   (E) represents the System Editor, (C:) is the drive, (\SAMPLE\) is the path and (FILE1) is the file name.

   To name a file you are about to create when you start the System Editor, leave a space after the E, type a path and name, and then press the Enter key.

   Note: If you only type E and a file name, the file you create will automatically be placed in the current directory.

   For information on specifying paths, directories and naming files, see *Getting Started*.

3. The System Editor starts and you can begin editing or creating your file.

To start the System Editor from the File Manager:

1. From the Directory Tree, select and start drive **C**.

2. Select and start the **OS2** directory.

   The list of OS2 files and subdirectories is displayed.

3. Select and start **E.EXE**.

   The System Editor starts.

4. Begin typing your file.

For information on how to start the System Editor from an associated file, see *Getting Started*.

When the System Editor starts, a blank file window named "OS/2 - UNTITLED" or "E.EXE - UNTITLED" is displayed on your screen. Until a file is given a name, the title bar displays the word "UNTITLED." The text area is blank except for a flashing cursor displayed directly below the action bar. If you typed in a file name, the name you typed is displayed instead of UNTITLED. If the file already exists, its content is displayed in the window.
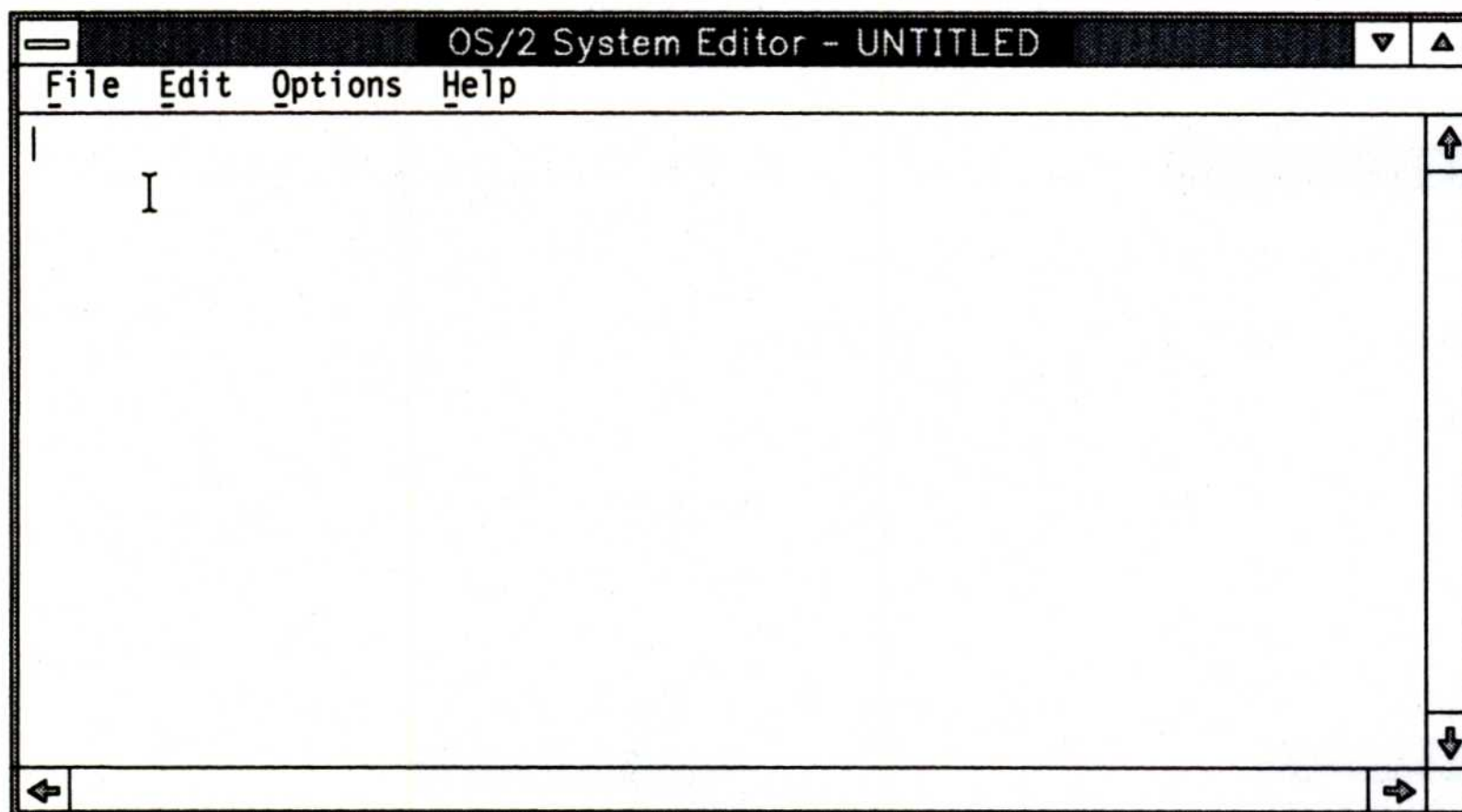
The System Editor window has a title bar, action bar, vertical and horizontal scroll bars, the System Menu icon, and the minimize and maximize icons. These window parts are described in *Getting Started*.

The mouse pointer changes from (↖) to the editing pointer (I) when it is placed in the text area of a file window. The editing pointer is removed when you start typing and displayed when you move the mouse.

Mode	Cursor	Description
Insert	|	Adds characters in front of text.
Overtype	▮	Selects and writes over characters to the right of the cursor.

You can switch between cursor modes at any time by pressing the Insert (Ins) key.

Shown below are the System Editor action bar pull-downs.

```
┌File─┐
│ New                    │
├────────────────────────┤
│ Open...                │
│                        │
│ Save                   │
│ Save as...             │
│ Autosave...            │
│                        │
│ Exit              F3   │
└────────────────────────┘
```

```
┌Edit─┐
│ Undo        Alt+Backspace │
├───────────────────────────┤
│ Cut            Shift+Del   │
│ Copy           Ctl+Ins     │
│ Paste          Shift+Ins   │
│                            │
│ Clear          Del         │
│                            │
│ Find...        Ctrl+F      │
│ Select all                 │
└────────────────────────────┘
```

```
┌Options─┐
│ Set font...        │
│ Set colors...      │
├────────────────────┤
│ √Word wrap         │
└────────────────────┘
```

```
┌Help─┐
│ Help for help...   │
├────────────────────┤
│ Extended help...   │
│ Keys help...       │
│ Help index...      │
├────────────────────┤
│ About...           │
└────────────────────┘
```

Mnemonics for pull-down choices are underscored. When assigned, key combinations are indicated next to a choice. A choice followed by an ellipsis (...) means that a pop-up window follows when that choice is selected. A choice that can be turned on or off is preceded by a check mark (√) when selected (turned on). Some pop-up windows can be dragged.

The System Editor is represented by the ( ) icon when it is mini-mized.

Help Information

Help is provided for information about tasks, help for help, extended help, key assignments, the help index, and field help. Press the F1 key or use the Help pull-down to access help.

The following is the Help pull-down:

```
Help
 Help for help...
 Extended help...
 Keys help...
 Help index...
 About...
```
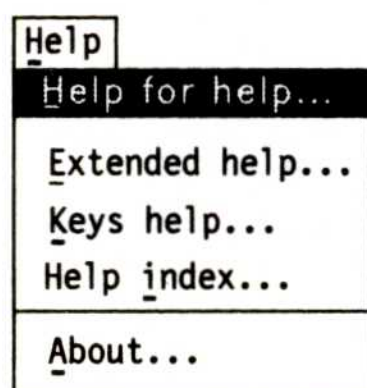
When you request help, a panel containing help information is displayed on your screen. After reviewing the help information, you can remove the help panel by doing one of the following:

- Press the Escape (Esc) key, or
- Start **Close** from the Help window's System Menu.

Help for help displays information on the Help function. To view this information:

- Select and start **Help for help** from the Help pull-down.

Extended help provides information about the active window. To view Extended help either:

- Select and start **Extended help** from the Help pull-down, or
- Press the F9 key. The F9 key is only active when a help window is displayed.

Keys help provides a list of available function keys as well as the actions they perform. To view Keys help:

- Select and start **Keys help** from the Help pull-down, or
- Simultaneously press the Shift and F10 keys.

Help Index displays a list of topics on which you can get help informa-
tion. You can scroll through the list of topics and **select** and start a
topic directly from the list, or you can do the following:

1. Select and start **Help Index** from the Help pull-down.
2. Select and start **Search** from the Help panel's File pull-down.
3. Select and start the **Index** check box.
4. Type one or more search words in the **Search string** entry field.
5. Select and start the **Search** pushbutton.

Field (contextual) help provides information about the area in an
application or pop-up that you have placed the cursor on. To access
Field help information:

1. Place the cursor on a field (area) of interest, then
2. Do one of the following:
   - Press the F1 key, or
   - Select and start the **Help** pushbutton.

Creating Files

You can give files and directories any names that conform to either File Allocation Table (FAT) or High Performance File System (HPFS) rules. For information on specifying paths and directories and naming files, see *Getting Started*.

Creating Files from a Command Prompt

Unless you specify an existing file, when the System Editor starts, a blank file window named "OS/2 - UNTITLED" or "E.EXE - UNTITLED" is displayed on your screen. If you named the file before pressing the Enter key, the name you gave the file is displayed instead of UNTITLED. You create your new file in the blank System Editor window.

A new file can be created when you start the System Editor from one of the OS/2 command prompts. See page 124 for information on opening existing files and naming and creating new files from a command prompt.

Creating Files from the Action Bar

The File pull-down is used when you create files from the action bar. Selecting **Save as** from the File pull-down allows you to name and save the file simultaneously. **Save as** also allows you to rename files. Selecting **New** from the File pull-down opens a new UNTITLED file window. The following is the File pull-down:

```
File
 New

 Open...

 Save
 Save as...
 Autosave...

 Exit          F3
```

To create a file from the action bar:

1. Select and start **Save as** from the File pull-down.

The Save as pop-up is displayed on your screen.

```
┌──────────────────────────────────────────────────────────────┐
│ ▭            ▓▓▓▓▓▓▓▓▓▓▓▓Save as▓▓▓▓▓▓▓▓▓▓▓▓                   │
│                                                                │
│  File name:█                                                  │
│  ┌──────────────────────────────────────────────────────┐    │
│  │                                                        │    │
│  Directory is: C:\                                         │
│                                                            │
│  Default type: Plain text                                  │
│                      Directories:                          │
│              ┌───────────────────────────┬───┐           │
│              │                            │ ⇧ │           │
│              │  OS2                       ├───┤           │
│              │  SPOOL                     │   │           │
│              │  [A:]                      │   │           │
│              │  [C:]                      │   │           │
│              │                            │   │           │
│              │                            ├───┤           │
│              │                            │ ⇩ │           │
│              └───────────────────────────┴───┘           │
│  ┌────────┐    ┌────────┐    ┌────────┐    ┌────────┐     │
│  │   OK   │    │  Type  │    │ Cancel │    │  Help  │     │
│  └────────┘    └────────┘    └────────┘    └────────┘     │
└──────────────────────────────────────────────────────────────┘
```
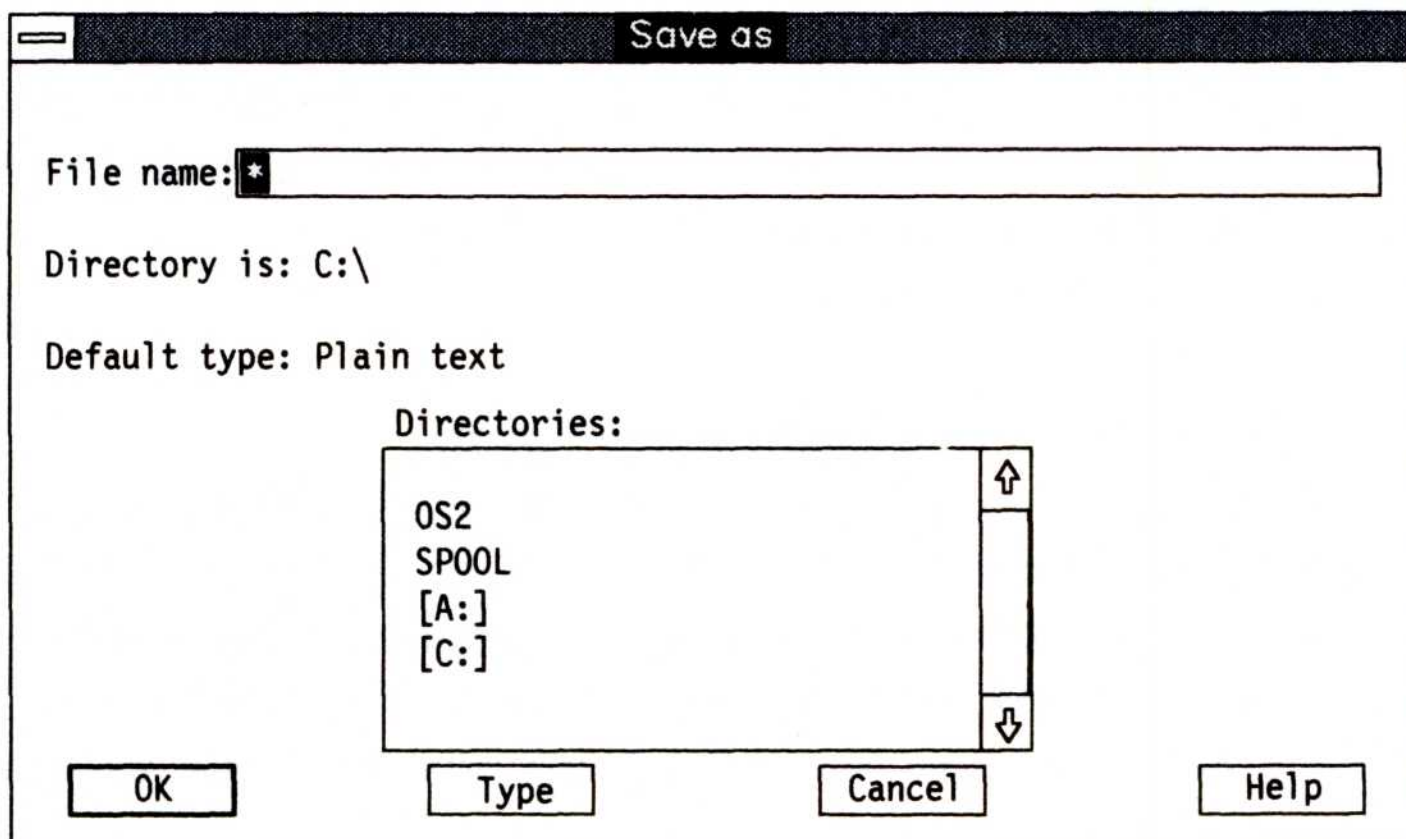
Refer to page 137 for information on using **Save as**.

2. Type a path and file name in the **File name** entry field.

3. Select and start the **OK** pushbutton, or press the Enter key.

   The Save as pop-up is removed.

4. Begin typing your file.

If you are editing a file and want to create a new file:

1. Select and start **Save** from the File pull-down to save the file you are editing.
2. Select and start **New** from the File pull-down.

   Note: If you try to create a new file before saving your current file, a warning message is displayed. If your file had not been previously named, the Save as pop-up is displayed.

   A blank System Editor window titled OS/2 SYSTEM EDITOR - UNTITLED or E.EXE - UNTITLED is displayed.

3. Do one of the following:
   • Select and start **Save as** from the File pull-down and name the file you are about to create.
   • Type in the file; then select and start **Save as** from the File pull-down and name the file you created.

Setting Options

Once an editing session has been started, various options can be changed and set.

The following is the Options pull-down:

```
┌Options─┐
│ Set font...          │
│ Set colors...        │
├──────────────────────┤
│√Word wrap            │
└──────────────────────┘
```
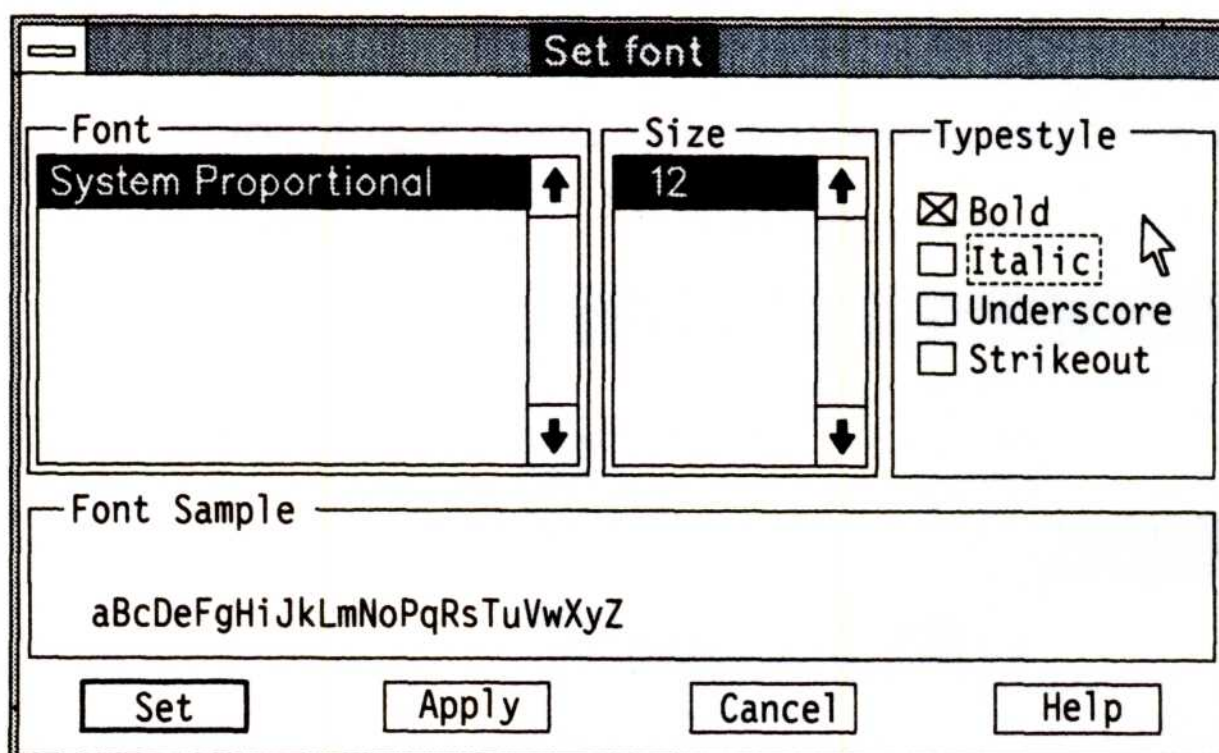
The options you set remain in effect until they are reset. They are saved and used the next time you start editing. You can reset the options at any time from the Options pull-down.

Set font allows you to select a font style, size and typestyle. When editing, if you want to see how your file will actually look when printed, choose a font style that is compatible with your printer. For example, choose a non-proportional font such as Courier if your printer font is non-proportional. If only one font is presently installed, see *Getting Started* for information on font installation.

To select fonts:

1. Select and start **Set font** from the Options pull-down.

   The Set font pop-up is displayed.

```
┌───────────────────── Set font ─────────────────────┐
│ ┌─Font──────────────┐ ┌─Size──┐ ┌─Typestyle ─────┐ │
│ │System Proportional│↑│ │12    │↑│ │⊠ Bold          │ │
│ │                   │ │ │      │ │ │☐ ⌐Italic¬      │ │
│ │                   │ │ │      │ │ │☐ Underscore    │ │
│ │                   │ │ │      │ │ │☐ Strikeout     │ │
│ │                   │↓│ │      │↓│ │                │ │
│ └───────────────────┘ └───────┘ └────────────────┘ │
│ ┌─Font Sample ─────────────────────────────────────┐│
│ │                                                  ││
│ │  aBcDeFgHiJkLmNoPqRsTuVwXyZ                       ││
│ └──────────────────────────────────────────────────┘│
│   ┌─ Set ─┐   ┌─ Apply ─┐   ┌─ Cancel ─┐   ┌─ Help ─┐ │
│   └───────┘   └─────────┘   └──────────┘   └────────┘ │
└─────────────────────────────────────────────────────┘
```

   The Tab key moves the cursor between selection fields. The ↑ and ↓ keys moves the cursor within the selection field.

2. Select a font name from the Font selection field.

As selections are made, an example of what you select is displayed in the Font Sample box. Select **Apply** to select and then continue with selections. Select **Cancel** to remove the Set font pop-up, return to the text area of the file, and leave the fonts unchanged.

3. Select a font size from the Size selection field.

4. Select a font style from the Typestyle check box.

   One or all of the check box styles can be selected. The selection style cannot be varied within the file.

5. Select **Set** and then start your selection.

   The Set font pop-up is removed, and the text within the file takes on the font, font size and type style you selected. You can change these selections as many times as you like during the editing session.

Note: Refer to "Word wrap" on page 133 for more information on displaying text in windows.

Set colors allows you to change the foreground and background colors of the file window. *Foreground* refers to the characters typed within the window, while *background* refers to the window's text area.

To change foreground and background colors:

1. Select and start **Set colors** from the Options pull-down.

   The Set colors pop-up is displayed.
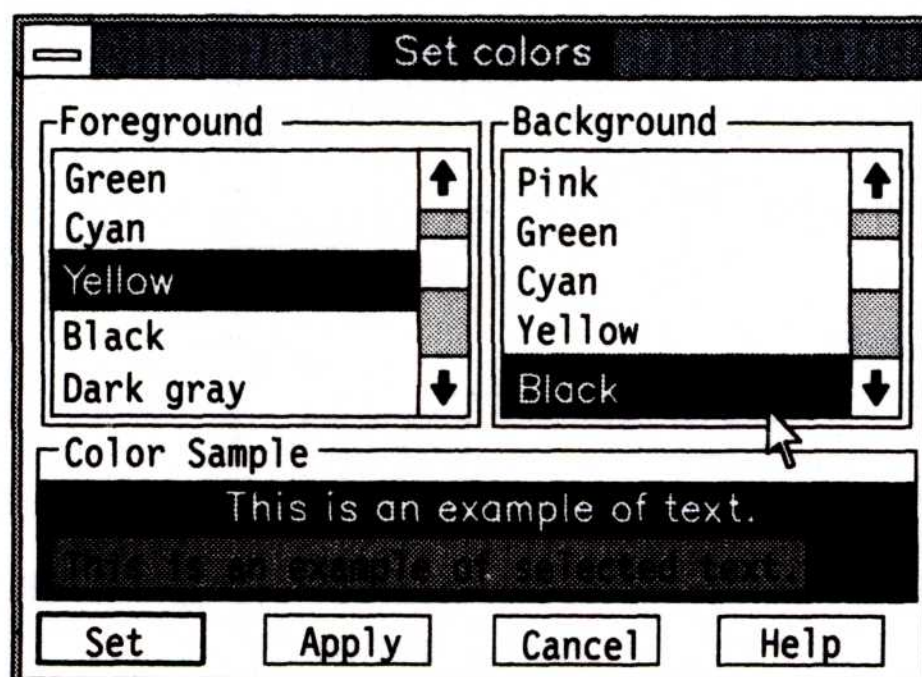
The colors you select are displayed in the Color Sample box. You will be able to see how selections appear together before setting them and returning to the text area of the file. Select **Apply** to display selections in the text area. Select **Cancel** to remove the Set colors pop-up, return to the text area of the file, and leave the colors unchanged.

2. Select a color from the Foreground selection field.
3. Select a color from the Background selection field.
4. Select and start the **Set** pushbutton.

   The Set colors pop-up is removed and the text area of the file is displayed in the colors you selected.

Word wrap affects how text is displayed in a window, not how it is stored or printed. A check mark ($\sqrt{}$) appearing in front of **Word wrap** indicates that it is turned on. Long lines are divided into short lines so that they fit inside the border of the window. This eliminates the need for horizontal scrolling. Word wrap is usually not appropriate for writing lines of code.

To turn Word wrap on:

* Select and start **Word wrap** from the Options pull-down.

   A check mark ($\sqrt{}$) is placed in front of Word wrap and the pull-down is removed.

To see how text will be printed, turn Word wrap off:

* Select and start **Word wrap** from the Options pull-down when it is shown with a check mark.

   The check mark and the pull-down are removed.

   When Word wrap is off, text is displayed according to where end-of-line (EOL) characters have been inserted. See page 136 for information on EOL characters.

   Note: Refer to page 131 for additional information on seeing how text will be printed.

Selecting Drives and Directories and Opening Files

You can display the contents of drives and directories and you can open files by selecting **Open** from the File pull-down.

1. Select and start **Open** from the File pull-down.

   The Open pop-up is displayed.

   ```
   ┌──────────────────────────────────────────────────────────────┐
   │ ▭                          Open                                │
   │                                                                │
   │  File name: [*]                                                │
   │                                                                │
   │  Directory is:  C:\                                            │
   │                                                                │
   │  Files:                         Directories:                   │
   │  ┌─────────────────────┐ ┌─┐    ┌─────────────────────┐ ┌─┐    │
   │  │ ABIOS.SYS           │ │⇧│    │ OS2                 │ │⇧│    │
   │  │ AUTOEXEC.BAT        │ │ │    │ SPOOL               │ │ │    │
   │  │ BASEDDO2.SYS        │ │ │    │ [A:]                │ │ │    │
   │  │ CMDREF.CP           │ │ │    │ [C:]                │ │ │    │
   │  │ COMMAND.COM         │ │ │    │                     │ │ │    │
   │  │ CONFIG.SYS          │ │ │    │                     │ │ │    │
   │  │ E.EXE               │ │⇩│    │                     │ │⇩│    │
   │  └─────────────────────┘ └─┘    └─────────────────────┘ └─┘    │
   │  ┌──────────┐        ┌──────────┐        ┌──────────┐          │
   │  │    OK    │        │  Cancel  │        │  Help    │          │
   │  └──────────┘        └──────────┘        └──────────┘          │
   └──────────────────────────────────────────────────────────────┘
   ```
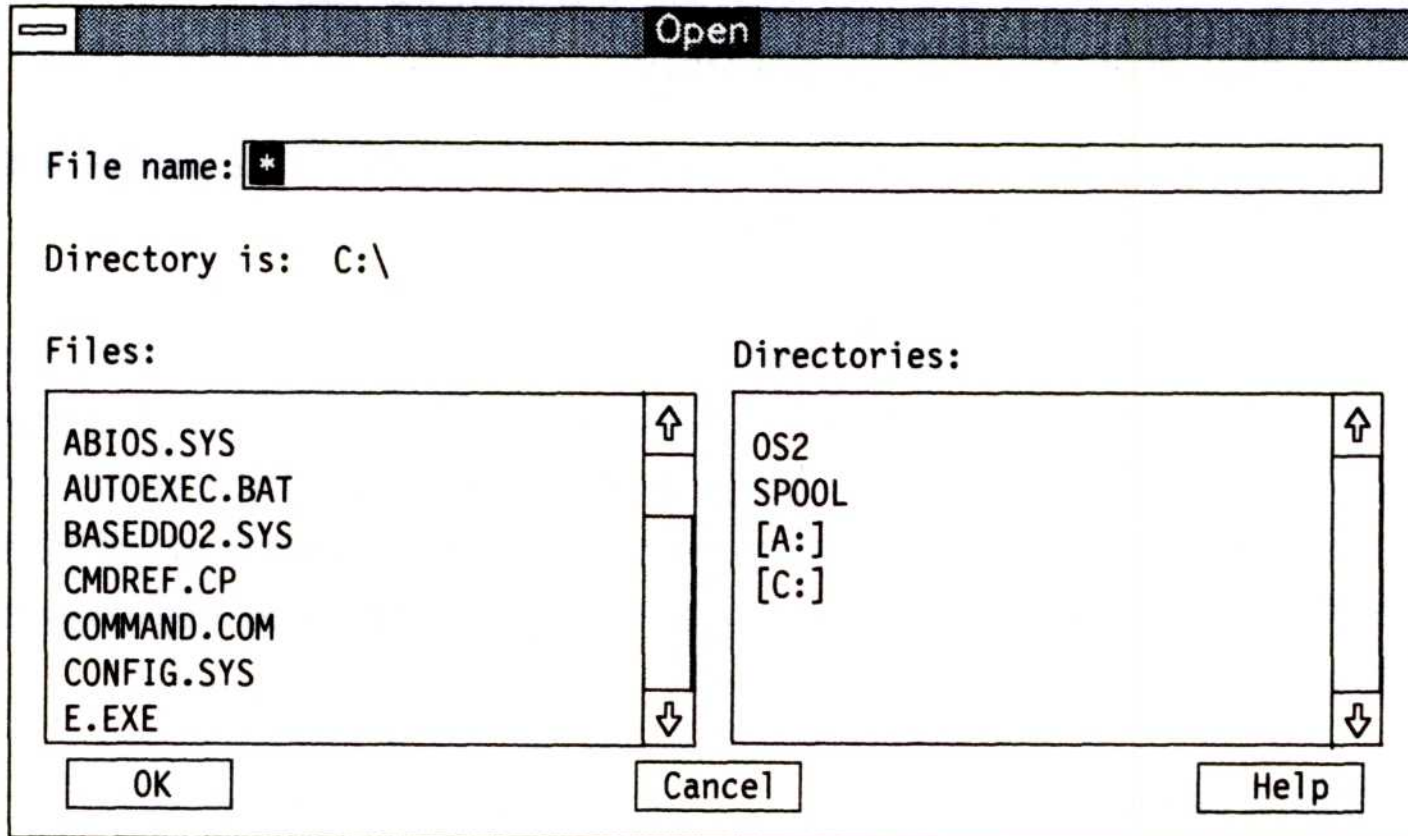
   The **File name** entry field is at the top of this window.

   Located directly below the **File name** entry field is the **Directory is** line. **Directory is** displays your final search path selections. When a path is entered in the **File name** entry field, it becomes part or all of the path and appears on the **Directory is** line. You cannot make direct entries on the **Directory is** line.

   The Files selection field shows the files in the selected directory.

   The Directories selection field displays your drives and directories. When selections are made from the Directories selection field, the **Directory is** line reflects the selection and the associated files are displayed in the Files selection field.

   The Tab key moves the cursor between selection fields. The ↑ and ↓ keys moves the cursor within the fields.

2. Open a file.

   You can open a file from the Open pop-up in these ways:
   - In the **File name** entry field, enter the path and name of the file you want to open.

Note: If you are already working in the correct subdirectory when the Open pop-up is displayed, just type the name of the file you want to open in the **File name** entry field, then press the Enter key.

- Select and start a directory from the Directories selection field, and then
- Select and start a file from the Files selection field.

  Note: If you select a file from the Files selection field, it replaces the name in the **File name** entry field.

The Open pop-up is removed, and the file you selected is displayed.

Entering Text

You can type 255 characters on a line. Spaces between words count as characters. The cursor moves one space to the right every time you type a character or press the Spacebar.

To enter text, begin typing at the cursor location, or use the Spacebar, Arrow keys (← ↑ ↓ →), or the Enter key to move the cursor to a different location; then begin typing.

To start a new line, press the Enter key. A new line is displayed below the one the cursor was on, and the cursor is positioned at the beginning of the new line.

Tabs

Tabs affect the way your file is saved to disk. When pressed, the Tab key moves the cursor to the next tab position which is always a multiple of eight spaces away from the start of a line. This in turn causes text to the right of the cursor to line up with the next tab position. An example of this is:

        One(tab)Three(tab)Twelve

Your information appears as:

        One   Three   Twelve

The text is aligned and will stay aligned even when you make changes. The System Editor adjusts the white space for you. If you

go over the eight space boundary, the tab position will change. Tab settings can be deleted by using the Backspace key.

To use Tabs:

1. Type in some information.
2. Press the Tab key.
3. Type in some information.
4. Press the Tab key.

   Repeat the above steps until all text and tab settings have been entered.

EOL Characters

End-of-line (EOL) characters are invisible and are entered by pressing the Enter key. EOL characters become part of your text and can be deleted in the same manner as any other character. Any character following an EOL character will start on a new line, regardless if Word wrap is on or off.

Saving and Closing Files

The File pull-down provides three ways to save the file you are editing:

- **Save**
- **Save as**
- **Autosave**

The following is the File pull-down:

```
File
 New

 Open...

 Save

 Save as...
 Autosave...

 Exit          F3
```

Save allows you to save and then continue editing your current file whenever you choose. A copy of the file is written to disk with its current name whenever you select **Save**.

To save and then continue editing a file:

- Select and start **Save** from the File pull-down.

  The pull-down is removed and the file is saved. You can continue adding and editing text, or you can exit the file. To ensure that all your changes are saved, always select **Save** before exiting a file.

Save as allows you to name, rename and save the current version of a file.

To name and save a file:

1. Select and start **Save as** from the File pull-down.

   The Save as pop-up is displayed.

```
┌─────────────────────────────────────────────────────────────┐
│ ▬                          Save as                            │
├─────────────────────────────────────────────────────────────┤
│                                                               │
│                                                               │
│   File name: █                                                │
│                                                               │
│   Directory is: C:\                                           │
│                                                               │
│   Default type: Plain text                                    │
│                     Directories:                              │
│                    ┌──────────────────────────────┐ ┌──┐      │
│                    │                               │ │ ⇧ │      │
│                    │ OS2                           │ └──┘      │
│                    │ SPOOL                         │           │
│                    │ [A:]                          │           │
│                    │ [C:]                          │           │
│                    │                               │ ┌──┐      │
│                    │                               │ │ ⇩ │      │
│                    └──────────────────────────────┘ └──┘      │
│   ┌────────────┐    ┌────────────┐   ┌────────────┐  ┌──────┐ │
│   │     OK     │    │    Type    │   │   Cancel   │  │ Help │ │
│   └────────────┘    └────────────┘   └────────────┘  └──────┘ │
└─────────────────────────────────────────────────────────────┘
```
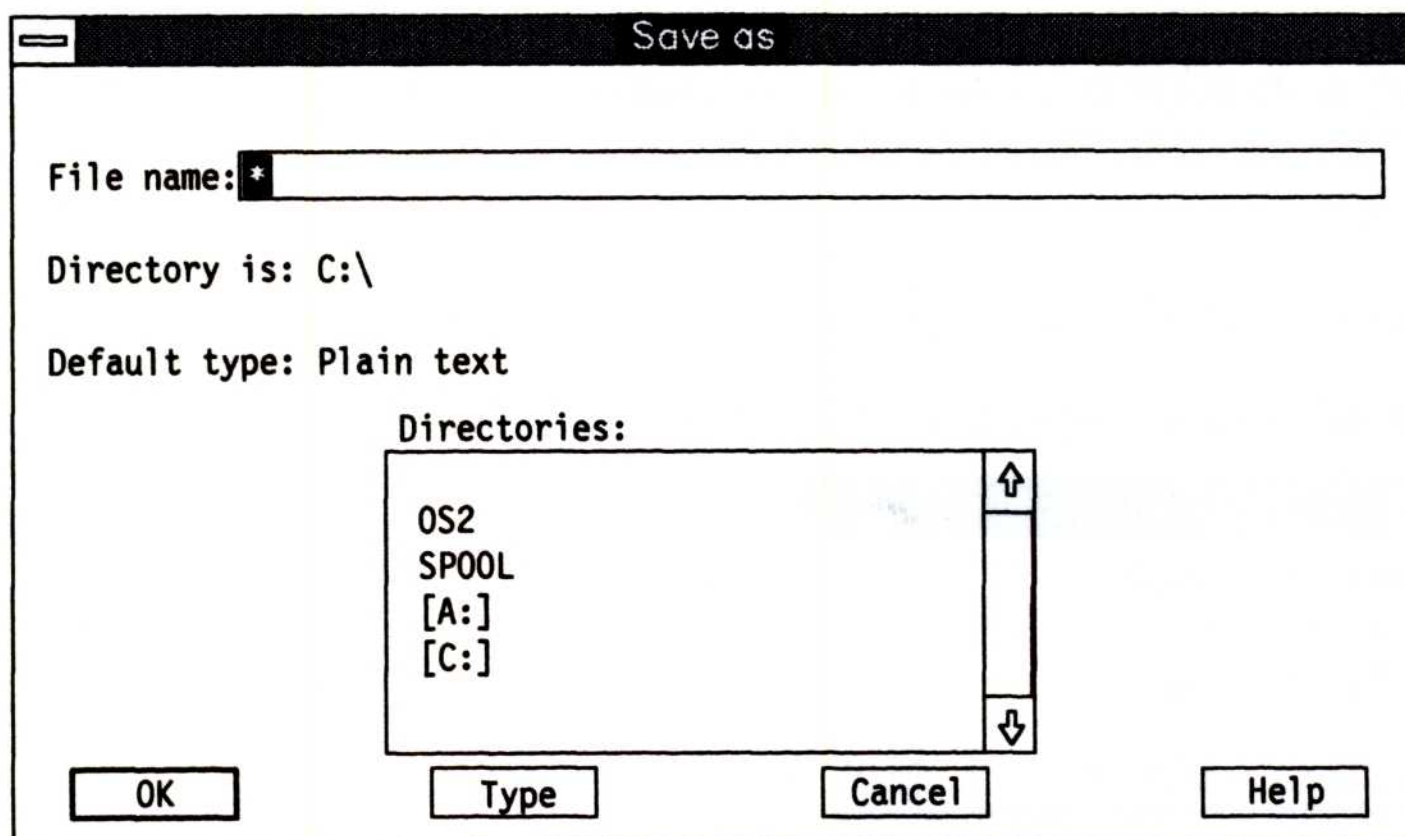
   Drives and directories are displayed in the Directories selection field for reference. The **Directory is** line displays your path entries.

   The Tab key moves the cursor between selection fields. The ↑ and ↓ keys moves the cursor within the selection field.

   To save the file in your current directory:
   - Type the name you want to give the file in the **File name** entry field; then select and start the **OK** pushbutton or press the Enter key.

   To save the file in a different directory:
   - Select the directory in the Directory selection field.

The directory you select is immediately displayed in the **File name** entry field.

2. Type the name you wish to give the file in the **File name** entry field; then press the Enter key.

   Note: A warning message is displayed if the file name entered already exists. A warning message is also displayed if you have not initially saved a new file. The system recognizes the existence of a file only after it is saved.

The Save as pop-up is removed and the file is saved with the path and name you entered. Use that path and name the next time you request the file.

Autosave allows you to automatically save an entire file on disk after changing or adding a specified number of changes in your file. You must set Autosave to on in order to use it.

To set Autosave on:

1. Select and start **Autosave** from the Files pull-down.

   The Autosave pop-up is displayed.
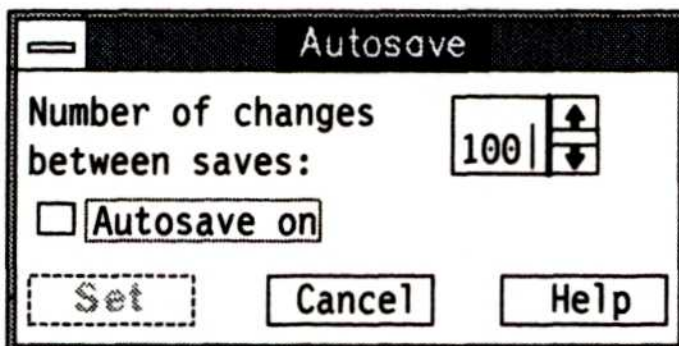
2. Select the **Autosave on** check box, or press the Tab key to reach it and then press the Spacebar to select it.

   Autosave is on when an X appears in the check box.

3. Enter a number, ranging from 1 to 9999, in the entry field.

   Use the Autosave scroll bar to enter a number, or type a number in the entry box. The number entered will represent the number of lines to be edited between Autosaves. Select **Cancel** to exit Autosave without changing your current setting.

4. Select and start the **Set** pushbutton, or Tab to it and press the Spacebar.

   The Autosave pop-up is removed.

   Note: Before closing a file, select **Save** from the File pull-down.

Deleting Files

Files are deleted from File Manager. For information on deleting files, see *Getting Started*.

Printing Files

Files are printed from File Manager. For information on printing files, see *Getting Started*.

Ending an Editing Session

Ending an editing session closes the System Editor program.

To end an editing session:

1. Select and start **Save** from the File pull-down.

   Note: The File Changed pop-up is displayed if you try to close a file before changes to a file have been saved.

```
┌─────────────────────────────────────────────────────┐
│ ━  ▓▓▓▓▓▓▓▓▓▓ Warning: File Changed ▓▓▓▓▓▓▓▓▓▓         │
├─────────────────────────────────────────────────────┤
│                                                       │
│  ┌───┐                                                │
│  │ ✳ │ Untitled has changed. Do you want to save it?  │
│  └───┘                                                │
│                                                       │
│  ┌───────┐ ┌──────────┐ ┌─────────┐ ┌────────┐ ┌──────┐ │
│  │ Save  │ │ Save as...│ │ Discard │ │ Cancel │ │ Help │ │
│  └───────┘ └──────────┘ └─────────┘ └────────┘ └──────┘ │
└─────────────────────────────────────────────────────┘
```
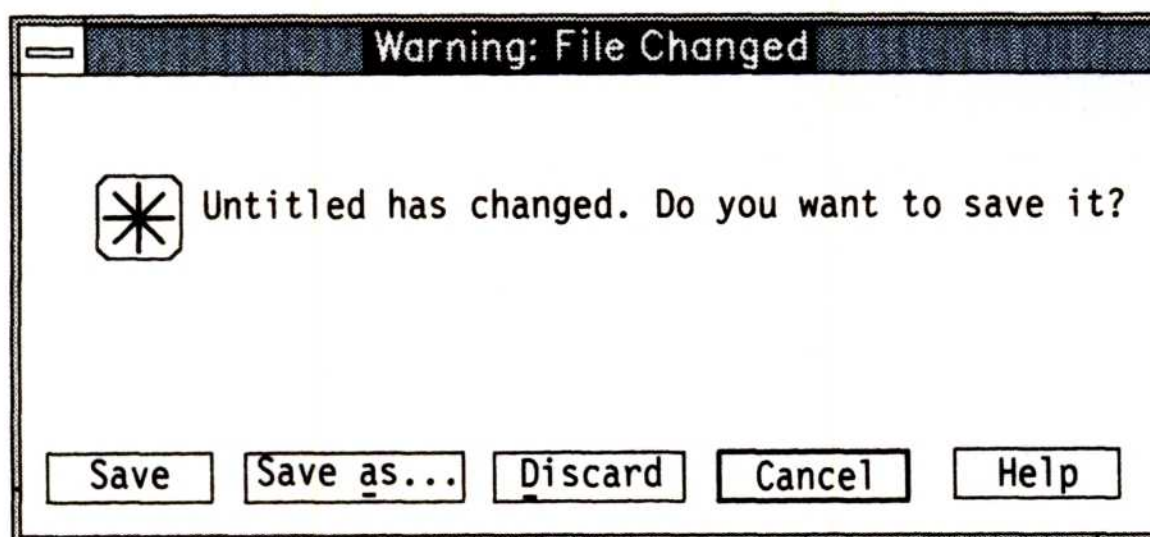
   To name or rename a file before closing it, select and start **Save as**. To erase your unsaved changes and then close the file, select and start **Discard**. To return to the file's text area, select and start **Cancel**.

2. Select and start **Close** from the System Menu pull-down, or **Exit** from the File pull-down.

   The file closes, the System Editor ends, and you are returned to wherever you started the System Editor.

Editing Files

This section provides information on:

- Adding and inserting text
- Splitting and joining lines
- Selecting and deselecting text
- Using Cut, Copy and Paste to merge text
- Deleting and restoring text
- Finding and changing text.

Adding and Inserting Text

To add or insert words or characters:

1. Position the cursor at the point where you want to add or insert text.
2. Type in the new information.

Splitting and Joining Lines

Splitting and joining lines allows you to change how your text is formatted.

To split a line:

1. Position the cursor where you want the text broken to start a new line.
2. Press the Enter key.

   When the Enter key is pressed, an EOL character is inserted at the cursor position and the text splits. The text from the cursor to the end of the line is moved to the next line.

To join lines:

1. Press the End key to move the cursor to the end of the line.
2. Press the Del key to delete the EOL character.

   The text from the line below joins your current line at the right of the cursor. When a line is too long to be joined to the one above, a message is displayed on your screen telling you so.

Selecting Text

To *select* text means to highlight the character, line, or block of lines that you want to *cut* (remove), *copy* (duplicate), or *paste* (insert) elsewhere within the current file or in another file. See page 151 for information on text selection keys.

To select text with the mouse:

1. Position the mouse pointer just before the first character to be selected.

2. Hold down mouse button 1; then *drag* (move) the pointer to just after the last character to be selected.

   You can also drag from just after the last character to just before the first character.

   When the mouse comes to a window border during a drag, the System Editor scrolls. This makes it possible to select more text than displayed in the window.

3. Release mouse button 1.

   The area is selected. The following is an example of selected text:

```
┌─────────────────────────────────────────────────────────┐
│ ▬            OS/2 System Editor - UNTITLED          ▼ ▲ │
│ File  Edit  Options  Help                                │
│┌────────────────────────────────────────────────────┬─┐│
││This will show you an example of selected text. You can select│⬆││
││a character, a word or a block of text. Once you have selected text│ ││
││it can be moved to another place in your file or to a different file.I│ ││
││                                                      │ ││
││                                                      │ ││
││                                                      │ ││
││                                                      │ ││
││                                                      │ ││
││                                                      │ ││
││                                                      │⬇││
│├──┬───────────────────────────────────────────────┬──┤ │
││⬅ │                                                │ ➡│ │
│└──┴───────────────────────────────────────────────┴──┘ │
└─────────────────────────────────────────────────────────┘
```
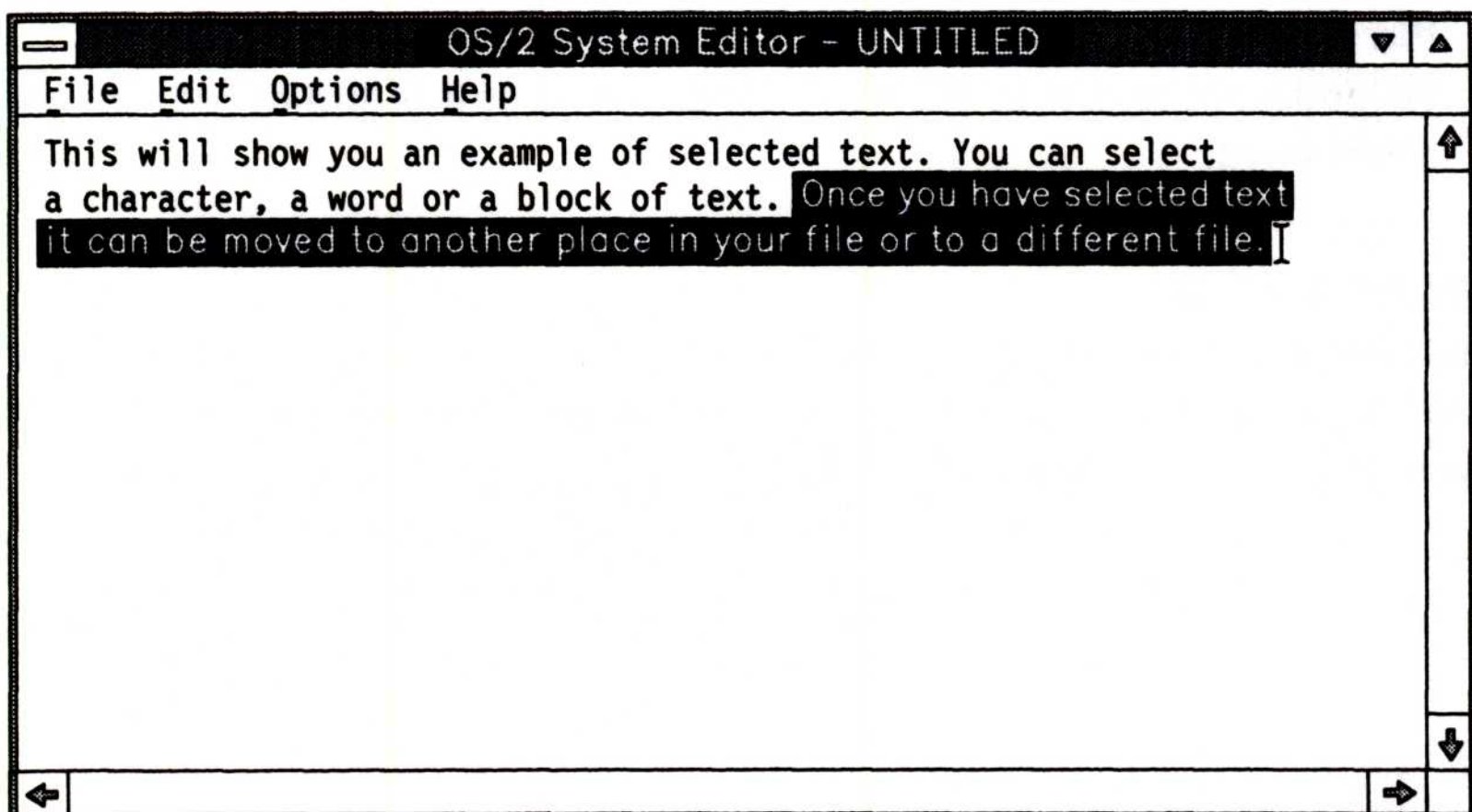
A selected area has an *anchor point* and a *cursor point*.

The anchor point is set when you hold down mouse button 1 and begin a drag. It is also set when you hold down the Shift key and use

the cursor movement keys to move the cursor. The anchor point is set at the cursor position, and the cursor moves to the cursor point position.

The cursor point is set when you release mouse button 1 or the Shift key. You can drag or move the pointer to where you want the cursor point set and then release mouse button 1; or using the cursor movement keys, position the cursor and then release the Shift key.

To select one word, double-click on any part of the word. The selection includes all blank spaces following the word. If you delete the word, the blank spaces are also deleted. The blanks are not deleted if you start typing.

The Edit pull-down has a **Select all** choice which sets the anchor point at the beginning of the text and the cursor point at the end. This is useful when using Cut, Copy and Paste. Information on Cut, Copy and Paste follows.

Deselecting Text

To *deselect* text means to remove the highlighting and leave the text unchanged.

- To deselect text with a mouse, click once within the text area of the window.
- To deselect text with the keyboard, press one of the cursor movement keys.

Merging Files

The System Editor can manage large files, making it possible for you to combine several files into one. When merging files, the Edit pull-down choices **Cut**, **Copy** and **Paste** are used.

Cut, Copy, and Paste

These three choices are used for moving text and merging files and can be found on the Edit pull-down. When using *cut, copy* and *paste*, the Clipboard can hold a maximum of 65,535 characters. See "Selecting Text" on page 141 if you are unfamiliar with how to select text. The following is the Edit pull-down.

```
Edit
Undo          Alt+Backspace

Cut           Shift+Del
Copy          Ctl+Ins
Paste         Shift+Ins

Clear         Del

Find...       Ctrl+F
Select all
```

Cut

This choice cuts selected text from your file and saves it on the Clipboard. The cut text can then be pasted into the file you are editing or into another file.

To use Cut:

1. Select the text to be cut.
2. To remove the selected text, do one of the following:
   - Select and start **Cut** from the Edit pull-down.
   - Hold down the Shift key, then the Del key; release the keys.

   The selected text is removed from the file and saved.

Copy

This choice copies your selected text to the Clipboard. The selected text can be pasted to a new position in the file you are editing or into another file.

To use Copy:

1. Select the text to be copied.
2. Do one of the following:
   - Select and start **Copy** from the Edit pull-down.
   - Hold down the Control (Ctrl) key and then press the Insert (Ins) key.

The selected text is copied from the file and saved on the Clip-
board. The file remains unchanged. See *Getting Started* for
information about the Clipboard.

Paste
This choice takes the text you have cut or copied to the Clipboard and
inserts it into a file at the cursor's position.

To use Paste:

1. Move the cursor to where you want the Clipboard text inserted.
2. Do one of the following:
    *   Select and start **Paste** from the Edit pull-down.
    *   While holding down the Shift key, press the Ins key.

    The selected text is inserted at the cursor position.

To copy all of the text to the Clipboard:

1. Select and start **Select all** from the Edit pull-down.
2. Select and start **Copy** from the Edit pull-down.
3. Open the file you want to paste the text into if you are not cur-
    rently in it.
4. Move the cursor to where you want the text inserted.
5. Select and start **Paste** from the Edit pull-down.

    The text is inserted at the cursor position.

Deleting Text
The System Editor automatically deletes all selected text when you
start entering new text. Other ways of deleting text are as follows:

*   If you only want to delete text, pressing the Del key or Backspace
    key deletes the entire selected text area.
*   The Del key and Backspace key also deletes text if there is no
    selected area. The Del key deletes the character following the
    cursor. The Backspace key deletes the character in front of the
    cursor.
*   Selecting **Clear** from the Edit pull-down deletes selected text.

Clear

Use this Edit pull-down choice to delete and discard your selected text. **Clear** is only active on the pull-down when text has been selected.

To use Clear:

1. Select the text that is to be deleted.
2. Select and start **Clear** from the Edit pull-down.

    The text you selected is deleted from your file.

Undoing Your Last Change

You can undo the very last change you made by selecting **Undo** from the Edit pull-down. This includes font, colors, and editing changes.

To undo your last change:

• Select and start **Undo** from the Edit pull-down.

    The pull-down is removed and your previous font, colors and text are restored.

Finding and Changing Text

You can find the occurrence of a *string* in your current file. A string can be a single character or a set of characters. Strings can be found and changed either from the beginning of your file to the end or from the cursor position to the end. One or all occurrences of a string can be changed.

To locate a string:

1. Position the cursor wherever you want the search to begin.

2. Select and start **Find** from the Edit pull-down.

    The Find pop-up is displayed.

```
 ┌──────────────────────────────────────────┐
 │ ▭               Find                      │
 ├──────────────────────────────────────────┤
 │ Find:     │ │                          │  │
 │                                           │
 │ Change to:│                            │  │
 │                                           │
 │     ⊠ Case sensitive      ☐ Wrap          │
 │                                           │
 │  ┌─────────┐ ┌─────────────────┐ ┌──────┐ │
 │  │  Find  ▗│ │Change, then Find│ │Change│ │
 │  └─────────┘ └─────────────────┘ └──────┘ │
 │  ┌──────────┐   ┌────────┐   ┌──────────┐ │
 │  │Change All│   │ Cancel │   │   Help   │ │
 │  └──────────┘   └────────┘   └──────────┘ │
 └──────────────────────────────────────────┘
```
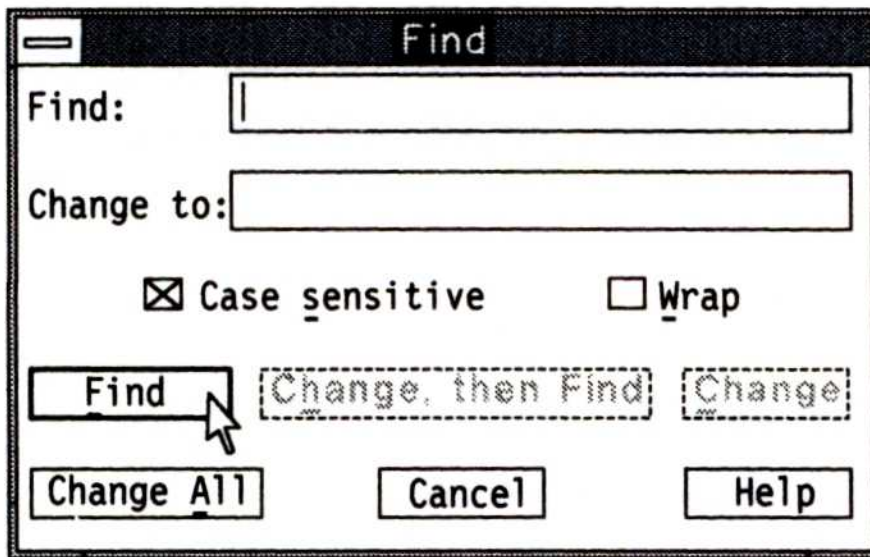
3. In the **Find** entry field, enter the string you want to find.

   The System Editor searches only for what you enter; therefore, the string must be entered exactly as it appears in the file.

4. Select **Wrap** to search the entire file.

   When selected, an X appears in the check boxes. Check box selections remain in effect until you deselect them.

5. Select **Case sensitive** if you wish to have uppercase and lower-case match.

   Select **Case sensitive** when you want to find only mixed case strings.

6. Select and start the **Find** pushbutton.

   When found, the string is highlighted. When the string is not found, a message panel is displayed stating so.

To change the string:

1. In the **Change to** entry field, type in what you want the string changed to.

   If this entry field is left blank, the string will be deleted when one of the **Change** choices is started.

2. Select and start one of the following:

   • To leave a string unchanged and search for the next string, select and start **Find** once again.

- To delete the string, select and start **Change**, then **Find**; **Change**; or **Change All**.
- Return to the text area and use the Backspace or Del key to delete the string.

A message is displayed when the string can no longer be found. To remove the message, select and start **Enter**.

Note: When **Change All** is selected, the System Editor changes all strings found without stopping.

3. To remove the Find pop-up, select and start **Cancel**.

Appendix A. System Editor Key Assignments

The following table shows the assigned keys for the System Editor program. Pressing Shift, Control (Ctrl), or Alternate (Alt) in combination with another key starts the action assigned to those keys. When two keys are joined by a plus (+), while holding down the first key press the second key. When three keys are joined by a plus (+), hold down the first two keys and then press the third key. See the quick reference card for other OS/2 key assignments.

Editing Keys

Key	Purpose
Alt + Backspace	Undoes the last change made.
Backspace	If text is selected, deletes the selected text.
	If no text is selected, deletes the character to the left of the cursor. Any remaining characters to the right of the cursor are shifted left.
	If the cursor is at the beginning of the line, deletes the EOL character on the previous line. The current line is joined with the previous one.
Ctrl + F	Displays the Find window so you can search for and change text.
Ctrl + Ins	Causes the selected text area to be copied and moved to the Clipboard.

Key	Purpose
Delete	If text is selected, deletes the selected text.
	If no text is selected, deletes the character to the right of the cursor. Any remaining characters to the right of the cursor are shifted left.
	If the cursor is at the end of the line, deletes the EOL character. The line is joined with the next line.
Enter	Splits the current line by inserting an EOL character at the cursor position.
Insert	Switches between insert and overtype mode.
	In insert mode, new text is inserted at the cursor position, moving the existing text to the right.
	In overtype mode, the new text replaces the existing text.
Shift + Del	Causes the selected text area to be cut from a file and moved to the clipboard.
Shift + Ins	Pastes the contents of the clipboard into the file at the cursor position.
	If text has been selected, the clipboard text replaces the selected text.
	If text has not be selected, the clipboard text is inserted at the cursor position.
Tab	Inserts a tab setting at the cursor position.

Text Selection Keys

Use these keys to select text and extend existing selected areas of text. The selection keys move the *cursor point* (ending point) in the text while keeping the *anchor point* (starting point) still.

Caution: If an area of text is selected, pressing any character key, the Backspace key, or the Delete (Del) key deletes the selected text.

Key	Purpose
Shift + arrow	Selects text or extends a selection in the direction of the arrow.
Shift + Home	Selects text or extends a selection to the beginning of the current line.
Shift + End	Selects text or extends a selection to the end of the current line.
Shift + Ctrl + →	Selects text or extends a selection to the beginning of the next word.
Shift + Ctrl + ←	Selects text or extends a selection to the beginning of the previous word.
Shift + Ctrl + Home	Selects text or extends a selection to the beginning of the file.
Shift + Ctrl + End	Selects text or extends a selection to the end of the file.

Cursor Movement Keys

Key	Purpose
Up Arrow (↑)	Moves the cursor up one line.
Down Arrow (↓)	Moves the cursor down one line.
Left Arrow (←)	Moves the cursor left one character.
Right Arrow (→)	Moves the cursor right one character.
Ctrl+→	Moves the cursor to the beginning of the next word.

Key	Purpose
Ctrl + ←	Moves the cursor to the beginning of the previous word.
Home	Moves the cursor to the beginning of the current line.
End	Moves the cursor to the end of the current line.
Ctrl+Home	Moves the cursor to the beginning of the first line in the file.
Ctrl + End	Moves the cursor to the end of the last line in the file.
PgUp	Scrolls the text up one window minus one line.
PgDn	Scrolls the text down one window minus one line.

Appendix B. DOS Mode Messages

This appendix contains DOS mode messages that do not have online help. The messages are listed in **bold** type, and the cause and action follow the message. If the first word after the cause is COMMANDS, the message is generated by several different programs or commands. If the first word is SYS (system message), the message is generated by an internal DOS mode file.

Access denied

Cause: COMMANDS. Processing the requested command violates the access mode of the file, subdirectory, or device involved. For example, an attempt was made to write to a file that is read-only, read a file that is write-only, or open a subdirectory as a file.

Action: Use a different file name, or if the file is read-only and you need to use that file, change its attribute with the ATTRIB command.

Bad command or file name

Cause: SYS. The command you entered is not a valid DOS mode command.

Action:

- Make sure that you are not running an OS/2-mode-only command.
- Check the spelling of the command and re-enter it.
- If the command name is spelled correctly, make sure that the default drive contains the external command or batch file you are trying to process.

Batch file missing

Cause: SYS. DOS could not locate the batch file it was processing. The file may have been erased or renamed by one of the steps within it. The current drive may have been changed within the batch file, and the batch processor can no longer find the .BAT file using your PATH. Batch processing stops and the DOS mode prompt appears.

Action:

- If the file name was changed, correct the command that changed the name.
- If the file was erased, use your backup copy. If you used EDLIN to create the file or make changes, rename the .BAK file to .BAT.
- Correct the command that deleted the file. Include the drive letters in the PATH.

BREAK is on/off

Cause: BREAK. This message shows the status of BREAK, either on or off.

Action: Enter the command you want. For example, if the screen shows "Break is off" and "Break is on" is desired, enter:

BREAK ON

Cannot do binary reads from a device

Cause: COPY. You used the /B parameter with a device name while trying to copy from the device. The copy cannot be performed in binary mode because COPY must be able to detect the end-of-file from the device.

Action: Re-enter COPY and omit the /B parameter, or re-enter COPY and use the /A parameter after the device name.

Cannot load COMMAND, this session halted

Cause: SYS. The system attempted to reload the command processor, but the area in which the system keeps track of available storage was destroyed; or the command processor was not found in the path specified by the COMSPEC environment variable. OS/2 sessions may still be active.

Action: If DOS mode is required, end all OS/2 sessions and make sure that COMMAND.COM is present on the diskette or drive you are using; then restart the system.

Cannot start COMMAND, exiting

Cause: SYS. While the system was attempting to load another copy of the command processor, it did not find enough available storage to contain the new copy of COMMAND.COM. OS/2 sessions may still be active.

Action:

- End all OS/2 sessions that may be running.
- Increase the RMSIZE value in your CONFIG.SYS file. The changed RMSIZE is active only after restarting the system.
- Restart the system.

Code page xxx not prepared for all devices

Cause: CHCP. CHCP was unable to select the code page for one of the following reasons:

- A device was not prepared for the requested code page.
- A device I/O error occurred.
- The device is currently printing.
- The device does not support code page switching.

Action: Make sure that there is a DEVINFO statement in the CONFIG.SYS file for each device (for example, printer, keyboard, and screen).

Code page xxx not prepared for system

Cause: CHCP. CHCP was unable to select the specified code page.

Action: Verify that the CODEPAGE statement in CONFIG.SYS includes the code page you are attempting to prepare.

Content of destination lost before copy

Cause: COPY. An illegal concatenation of files was detected during the COPY command. The destination file name was included among the source files being concatenated. The condition was not detectable until the destination file was copied over.

Action:

- Restore damaged files from your backup copy.
- Information message, no correction possible. Review the syntax of the COPY command to prevent this from happening again.

CTTY not supported in OS/2

Cause: SYS. Information message. The DOS command CTTY was attempted.

Action: No action required.

Current drive is no longer valid

Cause: COMMANDS. While attempting to get the current drive for the DOS mode prompt, COMMAND found that the drive is no longer valid.

Action: Change your current drive to a valid drive.

Duplicate file name or file not found

Cause: RENAME. You tried to rename a file to a file name that already exists on the disk, or the file to be renamed could not be found on the specified (or default) drive. RENAME is warning you that you are using the same name for two files, or it cannot find the file you are trying to rename.

Action: Make sure that you typed the file name correctly. Re-enter the RENAME command.

ECHO is on/off

Cause: ECHO. This message shows the status of ECHO, either on or off.

Action: Enter the command you want. For example, if the screen shows "Echo is off" and "Echo is on" is desired, enter the command:

```
ECHO ON
```

Error in EXE file

Cause: SYS. An error was detected in the relocation information placed in the file by the LINK program. This may be because of a modification to the file.

Action:

- If you are using a purchased program, rerun the program using your backup copy.
- If you are using a program you wrote yourself, go through the LINK procedure again.

Error writing to device

Cause: COMMANDS. The system met an I/O error when writing output to a device. The device is unable to handle the number of bytes requested.

Action: Change the amount of data in the file and retry the command.

EXEC failure

Cause: COMMANDS. The system met an error while reading a command or other program from disk.

Action: Try the command again. If the error reappears, attempt to load from a backup copy.

File allocation table bad, drive *d*
Abort, Retry, Ignore?

Cause: SYS. The file allocation has been damaged.

Action:

- Retry.
- Restart the system.
- Reformat drive *d* and restore any lost files from your backup copy.

File cannot be copied onto itself

Cause: COPY or XCOPY. You tried to COPY a file and place the copy (with the same name as the original) in the same directory and on the same disk as the original file.

Action: Change the name given to the copy, put it in a different directory, or put it on another disk.

File creation error

Cause: SYS and COMMANDS. An unsuccessful attempt was made to add a new file name to the directory or to replace a file that was already there.

Action: If the file was already there, check to see if the file is "read-only" and cannot be replaced; otherwise, run CHKDSK to determine if the directory is full or if some other condition caused the error.

File not found

> Cause: SYS and COMMANDS. A file named in a command or command parameter does not exist in the directory of the specified (or default) drive.

> Action: Retry the command using the correct file name.

FOR cannot be nested

> Cause: Batch file. More than one FOR command was found on one command line in the batch file.

> Action: Use only one FOR command per command line; then retry the command.

**Insert disk with batch file
and press any key when ready**

> Cause: SYS. The diskette that contained the batch file being processed was removed. The batch processor is trying to find the next command in the file.

> Action: Insert the diskette in the appropriate drive and press any key. Processing continues.

**Insert disk with \COMMAND.COM in drive *d*
and press any key when ready**

> Cause: SYS. The system is attempting to reload the command processor, but COMMAND.COM is not in the drive indicated by the COMSPEC environment variable.

> Action: Insert the diskette that has COMMAND.COM in the indicated drive and press any key.

**Insert diskette for drive *d*
and press any key when ready**

> Cause: SYS. In a system with logical drives, a drive that is not the current drive is being referenced, so DOS is asking for the diskette corresponding to the referenced drive.

> Action: If the diskette for *d* is different from the one currently in the drive, insert the appropriate diskette and press any key.

Insufficient disk space

Cause: SYS and COMMANDS. The disk does not contain enough free space to contain the file being written.

Action: If you suspect this condition is invalid, run CHKDSK to determine the status of the disk. Otherwise, use another disk and retry the command.

Insufficient memory

Cause: COMMANDS. The amount of available storage is too small to allow these commands to function.

Action: Increase the value for the RMSIZE statement and/or decrease the value for the BUFFERS statement in the CONFIG.SYS file. Restart the system and try the command again. If the message still appears, your system does not have enough storage to process the command.

Intermediate file error during pipe

Cause: SYS. The system is unable to create one or both of its intermediate files because the default drive's root directory was full, the system is unable to locate the piping files, or the disk does not have enough space to hold the data being piped.

Action: Erase some files from the default drive's root directory, and reissue the command that failed. If you get the same message, one of the programs in the command line has erased one or both of the piping files. Correct the program and reissue the command.

Invalid COMMAND.COM in drive d

Cause: SYS. When the system tried to reload the command processor, the copy of COMMAND.COM on the disk was found to be an incorrect version.

Action: Restore the correct COMMAND.COM to the indicated drive.

Invalid date

Cause: SYS. You entered an invalid date or delimiter. The only valid delimiters in a date entry are hyphens (-), slashes (/), and periods (.), or the date separator defined for the country specified by the COUNTRY statement in CONFIG.SYS.

Action: Re-enter a valid date.

Invalid directory

Cause: SYS and COMMANDS. One of the directories in the specified path does not exist.

Action: Retry the command using a valid directory, or create the specified directory.

Invalid disk change

Cause: SYS. The diskette in the drive was changed while files were still open on the diskette.

Action: Reinsert the correct diskette.

Invalid drive in search path

Cause: SYS. An invalid drive specifier was found in one of the paths specified in the PATH command.

Action:

1. Enter PATH. This displays the paths previously defined.

2. Find the invalid specifier.

3. Re-enter the PATH command with the valid drive specifier and the desired paths.

Invalid drive specification

Cause: COMMANDS. An invalid or nonexistent drive specification was entered in the command or in one of its parameters, or the source and target drives are the same.

Action: Re-enter the command using a valid drive specifier.

Invalid file name or file not found

Cause: RENAME or TYPE. You tried to rename a file that was either invalid or not found in the specified directory. TYPE does not allow global file name characters.

Action: Enter the correct file name.

Invalid number of parameters

Cause: COMMANDS. You have specified too few or too many parameters for the command you issued.

Action: Check the syntax of the command and re-enter the command.

Invalid parameter

Cause: SYS and COMMANDS. One or more parameters entered for these commands are not valid or have been placed in the wrong order.

Action: If the program expects a drive specifier, enter a colon (:) following the drive letter. In other cases, make sure that the character following the slash (/) is valid for the program being run.

Invalid path, not directory or directory not empty

Cause: RMDIR.

- The specified directory was not removed because one of the names you specified in the path was not a valid directory name.
- The directory you specified still contains entries for files or other subdirectories (except for the . and .. entries). It is possible that there are hidden files in the directory. The CHKDSK command detects them.
- You cannot remove a current directory.

Action: Try one of the following:

- Correct the invalid directory name in the path.
- Delete any files or remove any subdirectories in the directory.
- Change to a different subdirectory and try again.
- Run CHKDSK *.* in the directory. If there are hidden files, erase them according to the instructions of the program that created them.

Invalid path or file name

Cause: ATTRIB or COPY. You specified a directory or file name that does not exist.

Action: Use the correct name. Retry the command after checking for the following:

- Correct spelling of names
- Valid directory names
- Existence of file in the subdirectory specified.

Invalid time

Cause: TIME. An invalid time or delimiter was entered.

Action: Re-enter the correct time. The only valid delimiters are:

- Colon (:) between the hours and minutes
- Colon (:) between the minutes and seconds
- Period (.) between the seconds and hundredths of a second.

Label not found

Cause: Batch file. A GOTO command named a label that does not exist in the batch file. This caused the system to read to the end of the batch file, ending batch processing.

Action: If you do not want the GOTO to end the batch file, edit the batch file and put the label in the desired location.

Lock violation

Cause: XCOPY. A source file has part or all of it locked against reading.

Action: Wait a short time and try again.

Memory allocation error
Cannot load COMMAND, this session halted

Cause: SYS. The system attempted to reload the command processor, but the area in which the system keeps track of available storage was destroyed, or the command processor was not found in the path specified by the COMSPEC environment variable. OS/2 sessions may still be active.

Action: If DOS mode is required, end all OS/2 sessions and make sure that COMMAND.COM is present on the diskette or drive you are using; then restart the system.

Must specify ON or OFF

Cause: BREAK or VERIFY. You entered something other than ON or OFF.

Action: Try again, specifying ON or OFF.

No free file handles
Cannot start COMMAND, exiting

Cause: SYS. An attempt to load a second copy of the command processor failed because there are too many file handles opened throughout the system.

Action: To reduce the number of open files, end programs that are using OS/2 sessions.

No free file handles
The AUTOEXEC.BAT file could not be opened
Close some files in OS/2 mode and press any key to continue

Cause: SYS. An attempt to load the command processor failed because there are too many file handles opened throughout the system.

Action: To reduce the number of open files, end programs that are using OS/2 sessions.

No path

Cause: PATH. Information message. An alternate path for DOS to search for commands and batch files is not specified.

Action: No action required unless you want to define a set of paths. If so, enter PATH and the set of paths you want; then, press the Enter key.

Out of environment space

Cause: SYS. Information message. DOS did not accept the SET command you just issued because it could not expand the area in which the environment information is kept. This normally occurs when you try to add to the environment after loading a program which makes itself resident (PRINT or MODE, for example).

Action: Edit CONFIG.SYS to include or increase the /E: parameter in the SHELL statement; then restart the system.

Path not found

Cause: SYS and COMMANDS. A file or path named in a command or command parameter does not exist in the directory of the specified (or default) drive.

Action: Retry the command using the correct path and file name.

Program too big to fit in memory

Cause: SYS. The file containing the external command cannot be loaded because it is larger than the available free storage.

Action: Increase the number in the RMSIZE statement in your CONFIG.SYS file to a larger value and/or decrease the value for the BUFFERS statement in the CONFIG.SYS file. Restart your system and reissue the command. If the message reappears, your system does not have enough storage to process the command.

Syntax error

Cause: SYS. The command format you typed is incorrect.

Action: Use the correct format for this command.

Terminate batch job (Y/N)?

Cause: SYS. This message appears when you press the Ctrl and Break keys together while DOS is processing a batch file.

Action: Press the Y key to stop processing the batch file. Pressing the N key ends only the command that was processing when the Ctrl and Break keys were pressed; processing resumes with the next command in the batch file.

Top level process aborted, cannot continue

Cause: SYS. The system attempted to reload the command processor, but the area in which the system keeps track of available storage was destroyed; or the command processor was not found in the path specified by the COMSPEC environment variable. OS/2 sessions may still be active.

Action: If DOS mode is required, end all OS/2 sessions and make sure that COMMAND.COM is present on the diskette or drive you are using; then restart the system.

Unable to create directory

Cause: SYS and COMMANDS.

- The directory you want to create already exists.
- One of the directory path names you specified could not be found.
- You attempted to add a directory to the root directory and it is full.
- A file by that name already exists in that directory.

- The directory name you specified contains invalid characters or is a reserved device name.

Action: Do the following:

- Determine if a directory by that name already exists in the parent directory (or current directory).
- Recheck all your directory names to make sure that they are valid.
- Use CHKDSK to see if your current directory is full.

VERIFY is on/off

Cause: VERIFY. This message indicates the status of VERIFY, either on or off.

Action: Enter the command you want. For example, if the screen shows "Verify is off" and "Verify is on" is desired, enter the command:

VERIFY ON

Appendix C. Table of OS/2 Commands

This section contains an alphabetical table listing of OS/2 commands by name, mode, purpose, and type. If no mode is specified, the command operates in both OS/2 and DOS modes.

Note: Refer to the *OS/2 Command Reference* for the syntax of the command, associated parameters, and examples.

Name	Mode	Purpose of Command	Type
ANSI	OS/2	Permits ANSI control sequences that allow extended display and keyboard support.	Prompt
APPEND	DOS	Establishes a path for your system to search for non-program files outside the current directory that the PATH command does not search for.	Prompt
ASSIGN	DOS	Assigns a drive letter to a different drive.	Prompt
ATTRIB		Turns on or off the read-only attribute and the archive bit of a file, for selected files in a directory or for all files in a directory level.	Prompt
AUTOFAIL	OS/2	This statement allows or prevents system-wide hard-error and exception pop-up.	Config
BACKUP	OS/2	Backs up one or more files from one disk to another.	Prompt
BOOT		Provides the ability to change between DOS and OS/2 from the same fixed-disk drive.	Prompt

Name	Mode	Purpose of Command	Type
BREAK	DOS	Allows you to instruct DOS to check if you have pressed the Ctrl and Break keys together when a program requests OS/2 to perform any functions.	Config Prompt
BUFFERS		Establishes the number of disk buffers used by your system for maximum performance.	Config
CACHE		Specifies the parameters that the High Performance File System uses to write information to a disk. This command is specified as part of a RUN statement in the CONFIG.SYS file.	Config
CALL		Allows a file to be called from within another batch file without ending the first batch file.	Batch Config
CHCP		Allows you to switch back and forth between two code page character sets that are defined in your CONFIG.SYS file.	Prompt
CHDIR or CD		Changes the directory you are presently in or displays the directory name.	Prompt
CHKDSK		Analyzes the directories, files, and file allocation table (FAT) on the specified or default drive, and produces a disk status report.	Prompt
CLS		Clears the window or entire display screen of any information.	Prompt
CMD	OS/2	Starts a secondary command processor.	Prompt

Name	Mode	Purpose of Command	Type
CODEPAGE		Selects the system code pages (defined character sets) to be prepared by OS/2 for code page switching.	Config
COMMAND	DOS	Starts another DOS command processor.	Prompt
COMP		Compares the contents of two files.	Prompt
COPY		Copies or combines one or more files.	Prompt
COUNTRY		Identifies the country for which country-dependent information is selected.	Config
CREATEDD	OS/2	Creates a dump diskette.	Prompt
DATE		Displays or changes the date known to the system and resets the date on your system's clock.	Prompt
DDINSTAL	OS/2	Provides an automated way to install new device drivers.	Prompt
DETACH	OS/2	Simultaneously starts and detaches an OS/2 program from its command processor.	Prompt
DEVICE		Specifies the path and filename of a device driver to be installed in the CONFIG.SYS file.	Config
DEVINFO		Prepares a device for system code page switching.	Config
DIR		Displays files contained in a directory.	Prompt

Name	Mode	Purpose of Command	Type
DISKCACHE		Specifies the number of blocks of storage to allocate for control information and for use by the disk cache.	Config
DISKCOMP		Compares the contents of the diskette in the source drive to the contents of the diskette in the target drive.	Prompt
DISKCOPY		Copies the contents of a diskette to another diskette of the same capacity.	Prompt
DPATH	OS/2	Searches directories for data files located outside the current directory. (For use in CONFIG.SYS, refer to the SET command.)	Prompt Config
EASPLIT	OS/2	Provides the ability to split extended attributes from a file and re-join extended attributes to a file.	Prompt
ECHO		Allows or prevents the screen display of OS/2 commands run from a batch file. ECHO does not interfere with messages produced while commands are running.	Batch
ENDLOCAL	OS/2	Restores the drive, directory, and environmental variables that were in effect before you ran the SETLOCAL command.	Batch
ERASE or DEL		Erases files that you no longer want (or had previously copied to another subdirectory).	Prompt

Name	Mode	Purpose of Command	Type
EXIT		Ends the current command processor (CMD or COMMAND) and returns to the previous one, if one exists.	Prompt
EXTPROC	OS/2	Defines an external batch file processor, giving you the option to use your own batch processor.	Batch
FCBS	DOS	Determines file control blocks (FCBs) management information.	Config
FDISKPM	OS/2	Allows you to create or delete a partition or logical drive or to make a partition startable.	Prompt
FIND		Searches for a specific string of text in a file or files.	Prompt
FOR		Allows repetitive processing of OS/2 commands.	Batch
FORMAT		Creates the directory and file allocation tables on a disk. Formats a disk in the specified drive to accept OS/2 files.	Prompt
GOTO		Transfers control to the line following the one containing the appropriate label.	Batch
GRAFTABL	DOS	Allows additional characters from a language code page to be displayed when using display adapters in graphics mode.	Prompt
HELP		Allows a line of help as part of the command prompt, a help screen, and information related to a warning or error message.	Prompt

Name	Mode	Purpose of Command	Type
IF		Allows conditional processing of OS/2 commands.	Batch
IFS		Specifies the parameters used in managing disks and diskettes formatted for file systems other than the File Allocation Table (FAT).	Config
IOPL	OS/2	Allows I/O privilege to be granted to requesting processes.	Config
JOIN	DOS	Logically connects a drive to a directory on another drive. (You can only join a drive at the root directory.)	Prompt
KEYB	OS/2	Selects a keyboard layout to replace the current keyboard layout for all OS/2 and DOS full-screen sessions and all OS/2 windows sessions.	Prompt
KEYS	OS/2	Turns ON or OFF the retrieve mode of command line editing.	Prompt
LABEL		Creates or changes the volume identification label on a disk.	Prompt
LIBPATH	OS/2	Identifies the locations of dynamic link libraries for OS/2 programs.	Config
LOG		Allows you to log system events and place a record of that event in the System Log file.	Config
MAXWAIT	OS/2	Establishes the time limit for any program so there is no lack of access to the processor resource.	Config

Name	Mode	Purpose of Command	Type
MEMMAN	OS/2	Selects storage allocation options.	Config
MKDIR or MD		Creates a new subdirectory used to assist you in organizing your files and programs.	Prompt
MODE		Establishes operation modes for devices.	Prompt
MORE		Reads data from the standard input device and sends data to the standard output device (usually the display) one full screen at a time.	Prompt
MOVE	OS/2	Moves one or more files from one directory to another directory on the same drive. (If you prefer, you can give the files different names.)	Prompt
PATCH		Allows you to apply IBM-supplied patches to make repairs to software.	Prompt
PATH		Establishes a path for your system to search for program and data files in subdirectories other than your current directory. (For use in CONFIG.SYS, refer to the SET command.)	Prompt Config
PAUSE		Suspends running of the batch file and displays a "Press any key when ready ..." message.	Batch
PAUSEONERROR		Allows or prevents pausing when error messages are issued during the processing of the CONFIG.SYS file.	Config

Name	Mode	Purpose of Command	Type
PICICHG	OS/2	Converts a picture file into a format that can be exchanged between non-Presentation Manager applications.	Prompt
PICPRINT	OS/2	Prints one or more picture files on a specified device.	Prompt
PICSHOW	OS/2	Displays a picture file.	Prompt
PRINT		Prints or ends the printing of files.	Prompt
PRIORITY	OS/2	Selects priority calculation in scheduling regular class threads.	Config
PROMPT		Changes the command prompt. (For use in CONFIG.SYS, refer to the SET command.)	Prompt Config
PROTECTONLY	OS/2	Selects one or two operating modes.	Config
PROTSHELL	OS/2	Loads the user interface program and OS/2 command processor.	Config
PSTAT	OS/2	Allows you to display process status information at run time.	Prompt
RECOVER		Recovers files from a disk containing defective sectors.	Prompt
REM		Adds comments or line spacing in a batch file or a CONFIG.SYS file.	Config Batch
RENAME or REN		Changes a file name.	Prompt

Name	Mode	Purpose of Command	Type
REPLACE		Selectively replaces files on the target drive with files of the same name from the source drive. Also, selectively adds files from the source drive to the target drive.	Prompt
RESTORE	OS/2	Restores one or more BACKUP files from one disk to another.	Prompt
RMDIR or RD		Removes empty directories.	Prompt
RMSIZE	DOS	Specifies the highest storage address allowed for the DOS operating environment.	Config
RUN		Loads and starts a system program during system startup.	Config
SET		Sets one string in the environment equal to another string for later use in programs.	Config Prompt Batch
SETCOM40	DOS	Sets the COM port address so that a DOS program can access the COM port interface directly to support serial device, such as a plotter, printer, or mouse, when the COM0x device driver has been installed.	Prompt
SETLOCAL	OS/2	Lets you define the drive, directory, and environmental variables that are local to the current batch file.	Batch

Name	Mode	Purpose of Command	Type
SHELL	DOS	Loads and starts the DOS command processor, COMMAND.COM, or allows you to replace the DOS command processor with another command processor.	Config
SHIFT		Allows the use of more than ten replaceable parameters in batch file processing.	Batch
SORT		Reads data from standard input, sorts the data, and writes it to standard output.	Prompt
SPOOL	OS/2	Intercepts data on its way to the printer from simultaneously running applications, stores the output temporarily, and then releases the output to your printer so that it does not merge.	Prompt
START	OS/2	Starts an OS/2 mode program in another session.	Prompt
SUBST	DOS	Substitutes a drive letter for another drive and path.	Prompt
SWAPPATH	OS/2	Specifies the location and size of the swap file.	Config
SYSLOG	OS/2	Suspends or resumes logging of events in the System Log file. You can also display or print the log file as well as start or stop adding entries.	Prompt
THREADS	OS/2	Determines the maximum number of independent actions, known as threads, that can exist in the system at one time.	Config

Name	Mode	Purpose of Command	Type
TIME		Displays or changes the time known to the system, and resets the time on your system's clock.	Prompt
TIMESLICE	OS/2	Sets the amount of processor time allocated to processes and programs for both modes.	Config
TRACE	OS/2	Selects or sets the tracing of system events; also is used for debugging because it controls the tracing action taken (how much gets displayed to the user) during the execution of a program.	Config Prompt
TRACEBUF	OS/2	Establishes the size of the trace buffer.	Config
TRACEFMT	OS/2	Formats the contents of the system trace buffer, including time stamps, and outputs it to a display or printer for further analysis.	Prompt
TREE		Displays all the directory paths found on the specified drive, and optionally lists the files in the root directory and in each subdirectory.	Prompt
TYPE		Shows the text of a file.	Prompt
UNPACK		Decompresses files that have been compressed on the shipped diskettes. Files that are not compressed are copied.	Prompt
VER		Displays the OS/2 version number.	Prompt

Name	Mode	Purpose of Command	Type
VERIFY		Confirms that data written to a disk has been correctly written.	Prompt
VOL		Displays the disk volume label, if it exists.	Prompt
XCOPY		Selectively copies multiple files, including lower-level subdirectories.	Prompt

Index

S